Pattern Recognition 45 (2012) 3962–3984

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

# On minimum distribution discrepancy support vector machine for domain adaptation

Jianwen Tao [a], Fu-lai Chung [b], Shitong Wang [a,b,*]

[a] School of Digital Media, Jiangnan University, Wuxi, Jiangsu, China
[b] Department of Computing, Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

*Article history:*
Received 22 September 2011
Received in revised form
5 April 2012
Accepted 13 April 2012
Available online 11 May 2012

*Keywords:*
Domain adaptation learning
Support vector machine
Pattern classification
Maximum mean discrepancy
Maximum scatter discrepancy

ABSTRACT

Domain adaptation learning (DAL) is a novel and effective technique to address pattern classification problems where the prior information for training is unavailable or insufficient. Its effectiveness depends on the discrepancy between the two distributions that respectively generate the training data for the source domain and the testing data for the target domain. However, DAL may not work so well when only the distribution mean discrepancy between source and target domains is considered and minimized. In this paper, we first construct a generalized projected maximum distribution discrepancy (GPMDD) metric for DAL on reproducing kernel Hilbert space (RKHS) based domain distributions by simultaneously considering both the projected maximum distribution mean and the projected maximum distribution scatter discrepancy between the source and the target domain. In the sequel, based on both the structure risk and the GPMDD minimization principle, we propose a novel domain adaptation kernelized support vector machine (DAKSVM) with respect to the classical SVM, and its two extensions called LS-DAKSVM and $\mu$-DAKSVM with respect to the least-square SVM and the $\nu$-SVM, respectively. Moreover, our theoretical analysis justified that the proposed GPMDD metric could effectively measure the consistency not only between the RKHS embedding domain distributions but also between the scatter information of source and target domains. Hence, the proposed methods are distinctive in that the more consistency between the scatter information of source and target domains can be achieved by tuning the kernel bandwidth, the better the convergence of GPMDD metric minimization is and thus improving the scalability and generalization capability of the proposed methods for DAL. Experimental results on artificial and real-world problems indicate that the performance of the proposed methods is superior to or at least comparable with existing benchmarking methods.

## 1. Introduction

Data mining (especially web mining) and machine learning techniques, such as classification, clustering and regression, have already achieved significant success in many real world applications ranging from information retrieval, spam detection, to online advertisement and Web search. However, these technologies work well only under a common hypothesis that the training and testing data are drawn from the same distribution and/or feature space. When the distribution changed, the models learned from the prior information need to be reconstructed from scratch using the newly got training data. Constructing mining and learning algorithms for data that may not be identically and independently distributed (i.i.d.) is one of the newly emergent research topics in data mining and machine learning [1,3]. For example, the key challenge of text classification is that accurately-labeled task-specific 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. To solve non-i.i.d. learning problems, domain adaptation in transfer learning has been proposed to classify target domain data by using some other source domain data, even when the data may have different distributions. 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 by utilizing different but considerably related data in source domain to perform new learning tasks in target domain by connecting samples to their

[*] Corresponding author at: School of Digital Media, Jiangnan University, Wuxi, Jiangsu, China. Tel.: +86 510 85912151.
*E-mail address:* wxwangst@yahoo.com.cn (S. Wang).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.04.014

true labels, thus simplifying the expensive data collection process for exploring the knowledge discovery process [5,6].

Transfer learning refers to the problem of retaining and leveraging the knowledge available for one or more tasks, domains, or distributions to efficiently develop a reasonable hypothesis for a new task, domain, or distribution [22]. Instead of involving generalization across problem instances, transfer learning emphasizes the transfer of knowledge across tasks, domains, and distributions that are related but not the same. The default assumption of traditional supervised learning methods is that training and testing data are drawn from the same distribution. When the two distributions do not match, two distinct transfer learning sub-problems can be defined depending on whether the training and testing data refer to the same domain or not [6]: (1) learning under sample selection bias and (2) learning under domain adaptation. In this paper, we mainly focus on domain adaptation learning problems.

For a pattern classification problem, given a domain $D$ with a distribution $P(\boldsymbol{x},y)$, $\boldsymbol{x}\in\mathbf{X}$, $y\in\mathbf{Y}$, which is the true underlying distribution of the problem, where $\mathbf{X}$ and $\mathbf{Y}$ denote all possible instances and the corresponding class labels respectively. For DAL, unlabeled test patterns $\mathbf{X}_t = \boldsymbol{x}_{ti}{}_{i=1}^{m}$, $\mathbf{X}_t \subset \mathbf{X}$ are drawn from a target domain $D_t$ which is different from the source domain $D_s$ of training samples $\mathbf{X}_s=\{(\boldsymbol{x}_{si},y_{si})\}_i$, $\boldsymbol{x}_{si}\in\mathbf{X}$, $y_{si}\in\mathbf{Y}$. This may happen when the available labeled data are out of date, whereas the testing data are obtained from fast evolving information sources, or when series of data acquired at different times should be classified, while only training samples collected at a particular instance 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 $P_t(\boldsymbol{x},y)=P_t(y|\boldsymbol{x})\cdot P_t(\boldsymbol{x})$ 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 techniques. If $P_t(y|\boldsymbol{x})$ does not deviate a lot from $P_s(y|\boldsymbol{x})$, DAL may become necessary. In the framework of domain adaptation, most of the learning methods are inspired by the idea that these two considered domains, although different, are highly correlated [1,5,6].

As it may be well known, a major computational problem in domain adaptation is how to reduce the difference between the distributions of the source and the target domains. Hence, the core of DAL is to efficiently measure the distribution discrepancy between the two domains by finding a function from a given class of functions in a reproducing kernel Hilbert space. There exist several works describing how to measure the distance between distributions [1,49]. Intuitively, discovering a good feature representation across domains is crucial [3,6]. A good feature representation should be able to reduce the distribution discrepancy between two domains as much as possible, while at the same time preserving the underlying geometric structures (or scatter information) of both source and target domain data as much as possible. For instance, Blitzer et al. [4] showed that the hyperplane classifier that could best separate the data could provide a good method for measuring the distribution distance for different data representations [1]. Similarly, Gretton et al. [10] showed that for a given class of functions, the measure could be simplified by computing the discrepancy between two means of the distributions in a reproducing kernel Hilbert space, thus resulting in a maximum mean discrepancy (MMD) measure. The particular form of this measurement makes it easier to be incorporated into the corresponding optimization problems. Inspired by the ideas of both transductive support vector machine (TSVM) [15] and MMD, Quanz and Huan [1] proposed a so-called large margin kernel projected TSVM paradigm (LMPROJ) for domain adaptation problems based on the projected distance measure in a reproducing kernel Hilbert space (RKHS). The basic idea of LMPROJ is to minimize the distribution mean distance between the source and the target domain data by finding a feature translation in a RKHS based on empirical risk minimization principle, thus implementing transfer learning with cross-domains. Following LMPROJ and based on multiple kernel learning framework, Duan et al. also proposed a domain transfer SVM (DTSVM) and its extended version DTMKL for DAL problems such as cross-domain video concept detection and text classification [49,55].

As we may know well, mean (or expectation) and variance (or scatter) are two main features characterizing a distribution of samples in terms of first-order and second-order statistics, respectively. However, most existing DAL methods focus only on the first-order statistical matching which attempts to make the empirical means of the training and testing instances from the source and target domain to be closer in a Reproducing Kernel Hilbert Space (RKHS) [46]. Intuitively, it is not enough to measure the distribution distance discrepancy between two domains to some extent by only considering the mean of the distribution of samples [3,46,49]. Hence, MMD-based DAL methods [1,38,49], which only focus on the first-order statistics of the data distributions instead of simultaneously considering both the first-order and second-order statistics of the data distributions, still have considerable limitations on the generalization capacity for specific domain adaptation learning problems. Furthermore, since LMPROJ or DTSVM (or DTMKL) only focus on the mean consistency of domain distributions in the RKHS, they sometimes project the data onto some noisy directions of separation which is completely irrelevant to the target learning task [3], and thus resulting in poor performance.

In this paper, we claim that it is indispensable to consider both the mean and variance (or scatter) of data distribution in order to effectively measure the distribution discrepancy between the source and target domains. This motivates us to definitely utilize both MMD and scatter information of both domains to sufficiently evaluate their distribution discrepancy. To address the drawbacks of the MMD-based methods, we propose a novel domain adaptation kernelized support vector machine (DAKSVM) using a generalized projected maximum distribution discrepancy (GPMDD) metric on RKHS embedding domain distributions by simultaneously considering both the distribution mean and scatter discrepancies between the source and target domains. DAKSVM addresses the non-i.i.d. DAL problem by learning a low complexity decision function that well separates the source domain data and is regularized by both the complexity risk of the function and the projected distribution discrepancy between domains measured by simultaneously considering both means and variances of the two domains. The idea is to find a RKHS for which the means and variances of the training and testing data distributions are brought to be consistent, so that the labeled training data can be used to learn a model for the testing data. Particularly, we aim to obtain a linear kernel classifier based on the representer theorem [32], in a RKHS such that it achieves a trade-off between the maximal margin between classes and the minimal discrepancy between the training and test distributions.

Our idea can be illustrated in Fig. 1 where only two artificial data sets in 2D space are shown for simplicity and the training data and testing data have distinct distributions. In Fig. 1(a), $\mu_s$, and $\mu_t$ denote the means of source domain and target domain, respectively and $\sigma_s$ and $\sigma_t$ denote the scatters of the two domains, respectively. As shown in Fig. 1(a), in order to reduce the distribution distance between the two domains as much as possible, the MMD-based method LMPROJ aims to minimize the square mean discrepancy $\|\mu_s-\mu_t\|^2$ between these two domains. However, to be different from LMPROJ, DAKSVM attempts to minimize both the maximal scatter discrepancy $|\sigma_s-\sigma_t|$ and the square mean discrepancy $\|\mu_s-\mu_t\|^2$ between the two domains in

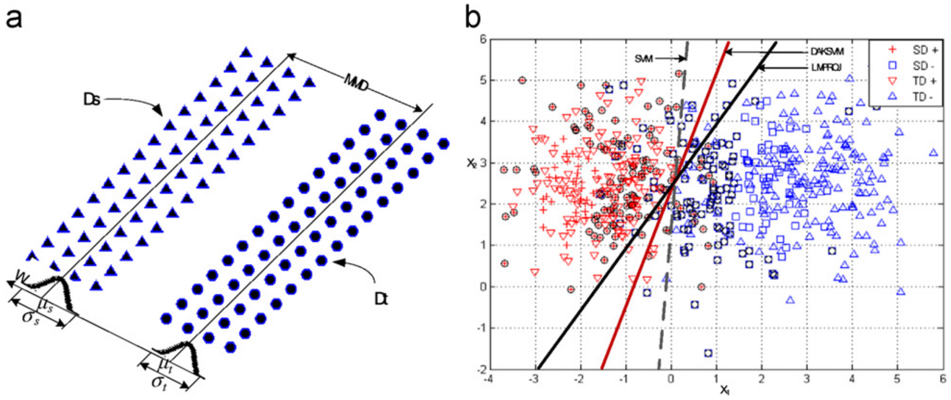

**Fig. 1.** Comparison among SVM, LMPROJ and DAKSVM. (a) Schematic diagrams of LMPROJ and DAKSVM. (b) Decision boundaries for classical SVM, LMPROJ and DAKSV on an artificial data set in 2-D DAL problem.

order to keep the distribution consistency of the two domains as much as possible. Fig. 1(b) shows a simple artificial 2D DAL problem, where the number of classes in source domain (SD) and target domain (TD) is 2 and the number of data in SD and TD is 300. $\mathrm{SD}_+$ and $\mathrm{SD}_-$ denote positive class and negative class in source domain, respectively, while $\mathrm{TD}_+$ and $\mathrm{TD}_-$ denote positive class and negative class in target domain, respectively. The mean and variance of SD are [−0.1345 2.9497] and [13.5742 14.0050], respectively, while the mean and variance of TD are [0.1419 2.9497] and [4.8217 0.9409] respectively. Intuitively, for both domains, $\mathbf{x}_1$ is the discriminative direction that separates the positive and negative class samples, while $\mathbf{x}_2$ is some noisy direction with minimal mean discrepancy between both domains. By focusing only on minimizing the mean discrepancy between both domains, LMPROJ would select the noisy direction $x_2$ as its optimal separation direction, thus leading to its discriminative direction of the pattern separation biased towards $x_2$, which is however completely irrelevant to the supervised learning task. Otherwise, though SVM builds a decision boundary fitting the training data well, the decision boundary is clearly not the optimal one as estimated on the test data set. However, our proposed method obviously outperforms LMPROJ and SVM due to simultaneously consider the discrepancy minimization of both means and variances of both domains.

The rationale for developing a domain adaptation technique in the framework of SVMs [17] is due to the effectiveness of this classification methodology that attempts to separate samples belonging to different classes by defining maximum margin hyperplanes [6]. Hence, the potential advantages of adopting the paradigm of SVMs can be seen from the followings:

(1) Empirical effectiveness with respect to other traditional classifiers, which results in relatively high classification accuracy and very good generalization capabilities.
(2) Convexity of the objective function used in learning the classifier, which results in a unique solution (i.e., the system cannot fall into suboptimal solutions associated with local minima).
(3) Capability of addressing classification problems in which no explicit parametric models on the distributions of data classes are assumed (distribution-free classifier).
(4) Possibility of representing the optimization problem in a dual format, where only nonzero Lagrange multipliers are necessary for defining the separation hyperplane (sparsity of the solution). And there are rigorous mathematical foundations for the dual problem such as the representer theorem, global optimization with polynomial running time using convex optimization, and geometric interpretations through generalized singular value decomposition.
(5) Possibility of defining nonlinear decision boundaries by implicitly mapping the available observations into a higher dimensional space (i.e., kernel trick).

In the literature, semi-supervised and transductive techniques based on SVMs have been proposed for solving problems under sample selection bias characterized by a large amount of unlabeled data but a reduced number of labeled data [6,20]. In particular, they try to recover information from the distribution of unlabeled data in the input space in order to improve the final classification performances. Nevertheless, these techniques are designed for handling problems where labeled and unlabeled data come from the same domain with the same distribution; thus, they are ineffective on domain adaptation problems with non-i.i.d. data, especially when the source/target-domain distributions are significantly different. In order to overcome the drawbacks of the previously proposed methods, the proposed methods here explore domain adaptation problems partially using the principles of both transductive SVMs (TSVMs) [15] and MMD. Hence, compared with the state-of-the-art DAL methods, our main contributions include the followings:

(1) The proposed methods inherit and extend the potential advantages of classical transductive SVMs (TSVMs) and MMD-based methods described as above, and further extend them to DAL.
(2) As a novel large margin domain adaption classifier, the proposed methods can reduce the distribution gap between different domains in a RKHS as much as possible, as they effectively integrate the large margin learner with the proposed generalized projected maximum distribution distance (GPMDD) metric, where both the distribution mean discrepancy and the distribution scatter discrepancy on RKHS embedding domain distributions are *simultaneously* considered. We justified the importance of the distribution scatter consistency between domains in DAL. The experimental results on artificial and real-world problems indicate that the proposed methods outperform or have comparable performance with existing benchmark methods.
(3) GPMDD metric in essence can help us design a natural multiple kernel learning framework in DAL. In contrast to multiple kernel learning methods (e.g. DTSVM [49] or DTMKL [55]) for DAL, our methods have clearer statistical interpretation from multi-kernel learning perspective and easier implementation in the sense of requiring less parameters to be

determined and no iterative two-stage minimization with the convex combination constraint. By introducing a tunable parameter $\gamma$ to control the Gaussian kernel bandwidth, we can adaptively reduce the scatter discrepancy between domains such that the scalability and generalization capability and even convergence rate of the proposed methods for DAL can be improved.

(4) In addition, we propose two extensions to the standard formulation of DAKSVM based on both $\nu$-SVM and least-square SVM (LS-SVM), respectively.

## 2. Discrepancy metrics on RKHS embedding domain distributions

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) [7]. A generalization of this idea is to embed probabilistic distributions into RKHS, giving us a linear method for dealing with higher order statistics [8,25]. Let a complete inner product space $H$ of functions $F$, and for $g \in F$, $g:\mathbf{X}\rightarrow\mathbf{R}$, where $\mathbf{X}$ is a nonempty compact set. If the linear dot function mapping $g \rightarrow g(\boldsymbol{x})$ exists for all $\boldsymbol{x} \in \mathbf{X}$, we call $H$ as a reproducing kernel Hilbert space. Under the aforementioned conditions, $g(\boldsymbol{x})$ can be denoted as an inner product:$g(\boldsymbol{x})=\langle g,\varphi(\boldsymbol{x})\rangle_H$, where $\varphi:\mathbf{X}\rightarrow H$ denotes the feature space projection from $\boldsymbol{x}$ to $H$. The inner product of the images of any points $\boldsymbol{x}$ and $\boldsymbol{x}'$ in the feature space is called kernel $k(\boldsymbol{x},\boldsymbol{x}')=\langle\varphi(\boldsymbol{x}),\varphi(\boldsymbol{x}')\rangle_H$. It is pointed out in [8] that the RKHS with Gaussian kernel is universal.

**Definition 1.** Integral probability metric on RKHS embedding distributions [2]

Given the set $\Theta$ 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, denote $Pk:=\int_M k(.,\boldsymbol{x})dP(x)$ for any $P\in\Theta$. If $k$ is measurable and bounded, then we may define the embedding of $P$ in $H$ as $Pk\in H$. Then, the RKHS embedding distribution distance between two such mappings associated with $P$, $Q\in\Theta$ is defined as:

$$\gamma_k(P,Q)=\|Pk-Qk\|_H. \qquad (1)$$

We may say $k$ is a characteristic kernel (CK) if the mapping $P\mapsto Pk$ is injective [2,8], in which $\gamma_k(P,Q)=0$ if and only if $P=Q$ [10]. Hence, $\gamma_k$ is viewed as the distance metric on $\Theta$. The RKHS embedding distributions cannot be distinguished when $k$ is not a CK, thus leading to the failure of RKHS embedding distribution measure. Hence, it is a key factor for the success of RKHS embedding distribution measure that whether $k$ is a CK or not. Fortunately, many popular kernel functions, such as polynomial kernel function, Gaussian kernel function and Laplace kernel function, are all CK and universal ones [8]. Particularly, it is worth noting that Gaussian kernel mapping can provide us an effective RKHS embedding skill for the consistency estimation of the probability distribution distance between different domains [2,8]. Hence, in the sequel, we adopt the Gaussian kernel function $k_\sigma(\boldsymbol{x},\boldsymbol{z})=\exp(-(1/2\sigma^2)\|\boldsymbol{x}-\boldsymbol{z}\|^2)$, where $\boldsymbol{x},\boldsymbol{z}\in\mathbf{X}$, and $\sigma$ 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 $\gamma_k$.

For domain adaptation learning problems, let $D_s$ and $D_t$ denote the source and target domain respectively and $\mathbf{X}_s\in D_s$ and $\mathbf{X}_t\in D_t$ denote the samples from $D_s$ and $D_t$ respectively with probability measures $P_s$ and $P_t$. Let $P_{x_s,x_t}$ denotes the joint probability measure of $\mathbf{X}_s\times\mathbf{X}_t$. Assume all measures are Borel ones and $\mathbf{X}_s$ and $\mathbf{X}_t$ are two compact sets. Besides, let a RKHS $H$ of a class of functions $F$ with kernel $k$, then for $g\in F, g:\mathbf{X}\rightarrow\mathbf{R}$, where $\mathbf{X}$ is a nonempty compact set, there exists the reproducing property as follows:

$$\langle g(\cdot),k(\mathbf{x},\cdot)\rangle=g(\mathbf{x}),\ \langle k(\mathbf{x},\cdot),k(\mathbf{x}',\cdot)\rangle=k(\mathbf{x},\mathbf{x}'),$$

where $\langle,\rangle$ denotes the inner product operator. Thus, by Definition 1, the unbiased empirical estimator of MMD on RKHS embedding domain distributions is defined as [44]:

$$\mathrm{MMD}(F,\mathbf{X}_s,\mathbf{X}_t)=\left\|\frac{1}{n}\sum_{i=1}^{n}\varphi(\mathbf{x}_i)-\frac{1}{m}\sum_{j=1}^{m}\varphi(\mathbf{z}_j)\right\|^2, \qquad (2)$$

where $\boldsymbol{x}_i\in\mathbf{X}_s$ and $\boldsymbol{z}_j\in\mathbf{X}_t$.

Specifically, by Definition 1, we can have the following definitions on RKHS embedding distribution distance metric.

**Definition 2.** Projected maximum mean distance metric on RKHS embedding domain distributions

Let a linear function $f$:$f(x)=\langle\boldsymbol{w},\varphi(x)\rangle$, where $\boldsymbol{w}$ is a projection vector. Then, the projected maximum mean distance metric on RKHS embedding domain distributions is defined as

$$\gamma_{KM}(f,\mathbf{X}_s,\mathbf{X}_t)=\boldsymbol{w}^T\mathrm{MMD}\boldsymbol{w}=\boldsymbol{w}^T\left\|\frac{1}{n}\sum_{i=1}^{n}\varphi(\mathbf{x}_i)-\frac{1}{m}\sum_{j=1}^{m}\varphi(\mathbf{z}_j)\right\|^2\mathbf{w}, \qquad (3)$$

where $\boldsymbol{x}_i\in\mathbf{X}_s$ and $\boldsymbol{z}_j\in\mathbf{X}_t$.

**Definition 3.** Projected maximum scatter distance metric on RKHS embedding domain distributions

Let a linear function $f$:$f(x)=\langle\mathbf{w},\phi(x)\rangle$, where $\boldsymbol{w}$ is a projection vector. Then, along the same line of Definition 2, the projected maximum scatter distance metric on RKHS embedding domain distributions is defined as

$$\gamma_{KS}(f,\mathbf{X}_s,\mathbf{X}_t)=\boldsymbol{w}^T\left|\frac{1}{n}\sum_{i=1}^{n}\varphi(\mathbf{x}_i)[\varphi(\mathbf{x}_i)]^T-\frac{1}{m}\sum_{j=1}^{m}\varphi(\mathbf{z}_j)[\varphi(\mathbf{z}_j)]^T\right|\mathbf{w}, \qquad (4)$$

where $\boldsymbol{x}\in\mathbf{X}_s$ and $\boldsymbol{z}\in\mathbf{X}_t$.

**Definition 4.** Generalized projected maximum distribution distance (GPMDD) metric on RKHS embedding domain distributions

By Definitions 2 and 3, a generalized projected maximum distribution distance metric on RKHS embedding domain distributions with probabilistic distribution $p$, $q\in P$ can be defined as

$$\gamma_{KMS}(f,\mathbf{X}_s,\mathbf{X}_t)=(1-\lambda)\gamma_{KM}(f,\mathbf{X}_s,\mathbf{X}_t)+\lambda\gamma_{KS}(f,\mathbf{X}_s,\mathbf{X}_t), \qquad (5)$$

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

It can be guaranteed by the following theorem that the GPMDD between both domains can be measured sufficiently.

**Theorem 1.** *[2] Let F be a unit ball defined in a universal RKHS H with kernel $k(\cdot,\cdot)$, which are all defined in a compact metric space. Let $\boldsymbol{X}_s$ and $\boldsymbol{X}_t$ be two compact sets generated from Borel probability metrics p and q, respectively, in the metric space with p and q. Then, $\gamma_{KMS}(F,\boldsymbol{X}_s,\boldsymbol{X}_t)=0$ if and only if $p=q$.*

## 3. Domain adaptation kernelized support vector machines (DAKSVMs)

### 3.1. Concepts and problem formulation

In this section, we introduce several definitions to clarify our terminologies in order to propose our algorithm and analysis on the domain adaptation learning problems in the subsequent sections.

**Definition 5.** Domain

A domain $D$ is composed of both feature space $\chi$ and marginal probabilistic distribution $P(\mathbf{X})$, i.e., $D=\{\chi,P(\mathbf{X})\}$, where $\mathbf{X}=\boldsymbol{x}_{i}{}_{i=1}^{N}\in\chi$.

**Definition 6.** Task

Given a specific domain $D=\{\chi,P(\mathbf{X})\}$, a task is composed of both tag space $\mathbf{Y}$ and target prediction function $f(\cdot)$, i.e., $T=\{\mathbf{Y},f(\cdot)\}$, where $f(\cdot)$ learned from the training dataset $\{\boldsymbol{x}_i,y_i\}$, where $\boldsymbol{x}_i\in\mathbf{X}$ and $y_i\in\mathbf{Y}$. The function $f(\cdot)$ can be used to make prediction for the tag $f(\boldsymbol{x})$ corresponding to $\boldsymbol{x}$. From a probabilistic point of view, $f(\boldsymbol{x})=P(y|\boldsymbol{x})$.

**Definition 7.** Domain adaptation learning

Given a source domain $D_s$ with its learning task $T_s$ and a target domain $D_t$ with its learning task $T_t$, we refer to DAL as the following problem. Given a set of labeled training dataset $\mathbf{X}_s=\{(\boldsymbol{x}_i,y_i)\}_i\in D_s\times\{\pm1\}$, where $y_i\in\mathbf{Y}_s\subset\mathbf{Y}$ is the class label corresponding to $\boldsymbol{x}_i$, from source domain $D_s$. Thus, we need to make prediction $f_t(\cdot)$ for some unlabeled test dataset $X_t=\{\boldsymbol{x}_j\}_j\in D_t$ from target domain $D_t$. $D_s$ with its task $T_s$ and $D_t$ with its task $T_t$ are different in the same feature space. When $D_s=D_t$ and $T_s=T_t$, DAL will be degenerated into classical machine learning problems.

Given an input space $\mathbf{X}$ and a label set $\mathbf{Y}$ of classes, a classifier is a function $f(\boldsymbol{x})$:$\mathbf{X}\rightarrow\mathbf{Y}$ which maps data $\boldsymbol{x}\in\mathbf{X}$ to label set $\mathbf{Y}$. In this context, let us consider two data sets $\mathbf{X}_s=\{(\boldsymbol{x}_{s1},y_{s1}),\ldots,(\boldsymbol{x}_{sn},y_{sn})\}$ drawn from $\mathbf{X}\times\mathbf{Y}$ with probabilistic distribution $P_s(\boldsymbol{x}_s,y_s)$ and $\mathbf{X}_t=\{\boldsymbol{x}_{t1},\ldots,\boldsymbol{x}_{tm}\}$ drawn from $\mathbf{X}$ with probabilistic distribution $P_t(\boldsymbol{x}_t,y_t)$ where $y_t$ needs to be predicted, which are composed of $n$ source domain and $m$ target domain patterns respectively, and usually $0\le m\ll n$. $\boldsymbol{x}_s$ and $\boldsymbol{x}_t$ are $d$-dimensional feature vectors with respect to $\mathbf{X}_s$ and $\mathbf{X}_t$ respectively. The classical large margin learning machines (such as SVMs) work well under such hypothesis as $P_s(\boldsymbol{x}_s,y_s)=P_t(\boldsymbol{x}_t,y_t)$. However, DAL can make accurate prediction for the unlabeled target data to some extent by learning a classifier even under such hypothesis $P_s(\boldsymbol{x}_s,y_s)\neq P_t(\boldsymbol{x}_t,y_t)$. The performance of DAL depends on both the complexity of the investigated problems and the correlation between $P_s(\boldsymbol{x}_s,y_s)$ and $P_t(\boldsymbol{x}_t,y_t)$ [6]. In this paper, the proposed method is formulated under the following hypothesis:

(1) There are only one source domain and one target domain sharing the same feature space in DAL problems, which is the most popular hypothesis used by the state-of-the-art methods.
(2) A training data set $\mathbf{X}_s=(\boldsymbol{x}_{si},y_{si})_i$ is available for $D_s$ while a testing data set $\mathbf{X}_t=(\boldsymbol{x}_{tj},y_{tj})_j$ is available for $D_t$ with $y_{tj}$ which is unknown.
(3) $P_s(\boldsymbol{x}_s,y_s)\neq P_t(\boldsymbol{x}_t,y_t)$ and $P_s(y_s|\boldsymbol{x}_s)\neq P_t(y_t|\boldsymbol{x}_t)$.

Inspired by the idea of manifold regularization [12], MMD based methods (e.g. LMPROJ [1], DTSVM [49], etc.) can be formulated as

$$f=\min_{\boldsymbol{w}\in H_K} C\sum_{i=1}^{n}\xi_i+\tfrac{1}{2}\|\boldsymbol{w}\|_K^2+\lambda\gamma_{KM}(f,\mathbf{X}_s,\mathbf{X}_t)$$

$$\text{s.t.}\quad \begin{array}{l} y_i(\boldsymbol{w}^T\varphi(\boldsymbol{x}_i)+b)\ge 1-\xi_i,\\ \xi_i\ge 0,\quad i=1,\ldots,n \end{array}, \tag{6}$$

where $\boldsymbol{w}$ is a normal projection vector, $K$ is a kernel with kernel mapping function $\varphi$, $H_K$ is a set of functions in the kernel space, $\lambda$ is a trade-off parameter, and $\gamma_{KM}(f,\mathbf{X}_s,\mathbf{X}_t)$ is the projected distribution mean distance metric between the source and target domains, where, $\boldsymbol{x}_i\in\mathbf{X}_s$.

However, Eq. (6) reveals a key limitation of MMD-based methods to some extent, i.e., they have not considered sufficiently the potential scatter statistics, which may include the underlying discriminative information in both domains for DAL, such that they may lead to "overfitting" phenomenon in some specific pattern recognition applications. Therefore, in this paper, we propose a robust domain adaptation kernelized SVM (DAKSVM) regularized by GPMDD metric on the RKHS embedding domain distributions, which partially extends the ideas of classical SVMs and MMD. The key goals of our methods are to find a feature transformation 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 criterion for the training data, and thus learning a robust model to effectively make prediction for target domain.

### 3.2. DAKSVM

#### 3.2.1. Optimal formulation of DAKSVM

For simplicity, we first consider binary pattern classification problems, and then we propose a so-called least-square DAKSVM (LS-DAKSVM) based on the classical least-square SVM (LS-SVM) [40] as an extension to the standard DAKSVM method for multi-class pattern classification problems.

For DAL problems, DAKSVM aims to find a linear transform $f(x)=w^T\varphi(\mathbf{x})$ in a universal RKHS with Gaussian kernel mapping, where $\boldsymbol{w}$ is a linear projection vector, so as to minimize the distribution discrepancy between domains as well as to reduce the empirical risk of the classification decision function as much as possible, thus implementing transfer learning across domains. DAKSVM can be formulated as

$$\min f=C\sum_{i=1}^{n}V(\mathbf{x}_i,y_i,f)+\gamma_{KMS}(f,\mathbf{X}_s,\mathbf{X}_t), \tag{7}$$

where $\boldsymbol{x}_i\in\mathbf{X}_s$ is a set of training data and matrix $\varphi(\mathbf{X}_s)=(\varphi(x_{s1}),\varphi(x_{s2}),\ldots,\varphi(x_{sn}))$, $y_i\in\mathbf{Y}_s$ is the class label of $\boldsymbol{x}_i$, $C>0$ is a regularization coefficient, and $V$ measures the fitness of the function in terms of predicting the class labels for the training data and is called the risk function. The hinge loss function is a commonly used risk function in the form of $V=(1-y_if(x_i))_+$ [19] in which $(x)_+=x$ if $x\ge0$ and zero otherwise.

Therefore, the linear function $f$ in (7) represented by a vector $\boldsymbol{w}$ can be represented as

$$\underset{w,b,\xi}{\operatorname{argmin}} f=C\sum_{i=1}^{n}\xi_i+\gamma_{KMS}(f,\mathbf{X}_s,\mathbf{X}_t)$$

s.t.

$$y_i((\boldsymbol{w},\varphi(\mathbf{x}_i))+b)\ge 1-\xi_i,\quad i=1,2,\ldots,n. \tag{8}$$

In order to solve the primal in (8) effectively, we introduce the following revised Representer Theorem for DAL problems as follows.

**Theorem 2.** *Representer Theorem for DAL [32]*

For a DAL problem, let $\psi$:$[0,\infty)\rightarrow\mathbf{R}$ be a strictly monotonic increasing function, $\mathbf{X}=\mathbf{X}_s\cup\mathbf{X}_t$ be a dataset, and $c$:$(\mathbf{X}\times\mathbf{R}^2)^n\rightarrow\mathbf{R}\cup\{\infty\}$ be any loss function. Then, regularized risk function is defined as

$$R(f)=c((\boldsymbol{x}_i,y_i,f(\boldsymbol{x}_i))_{i=1}^{n})+\psi(\|f\|_H^2), \tag{9}$$

where $f\in H$ is represented as

$$f(\boldsymbol{x})=\sum_{i=1}^{m}\beta_i k(\boldsymbol{x}_i,\boldsymbol{x})+\sum_{j=1}^{n}\beta_j k(\boldsymbol{z}_j,\boldsymbol{x}), \tag{10}$$

where $k$ is a kernel, $\boldsymbol{x}_i\in\mathbf{X}_s$, $\boldsymbol{y}_i\in\mathbf{Y}_s$, $\boldsymbol{z}_j\in\mathbf{X}_t$ and $\beta_i$ is a coefficient.

By Theorem 2, we can have the following theorem.

**Theorem 3.** *The primal of DAKSVM can be formulated as*

$$\min_{\boldsymbol{\beta},\xi,b} f=\tfrac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\Omega}\boldsymbol{\beta}+C\sum_{i=1}^{N}\xi_i, \tag{11}$$

$$\text{s.t. } y_i\left(\sum_{j=1}^{n+m}\beta_j k_\sigma(\boldsymbol{x}_i,\boldsymbol{x}_j)+b\right)\geq 1-\xi_i,\quad i=1,\ldots,n, \tag{12}$$

*where $\boldsymbol{x}_i\in\mathbf{X}_s$, $\boldsymbol{x}_j\in\mathbf{X}_s\cup\mathbf{X}_t$, $\boldsymbol{\Omega}$ is a positive semi-definite kernel matrix as defined in Appendix 1, and $\xi_i\geq 0$.*

A proof of Theorem 3 can be found in Appendix 1.

It is shown from Eq. (27), Eqs. (28) and (29) in Appendix 1 that the proposed DAKSVM measures the distribution discrepancy across 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 a certain extent.

**Theorem 4.** *The dual of the primal in Eqs. can be formulated as*

$$\min_{\alpha}\ \tfrac{1}{2}\boldsymbol{\alpha}^T\mathbf{H}^{\phi}\boldsymbol{\alpha}-1^T\boldsymbol{\alpha}, \tag{13}$$

$$\text{s.t. } 0\leq\alpha_i\leq C,\quad i=1,\ldots,n, \tag{14}$$

$$\sum_{i=1}^{n}\alpha_i y_i=0, \tag{15}$$

where $\mathbf{H}^{\phi}=\tilde{\mathbf{Y}}\mathbf{K}_s^T(\boldsymbol{\Omega})^{-1}\mathbf{K}_s\tilde{\mathbf{Y}}$, and $\tilde{\mathbf{Y}}=\mathrm{diag}(y_1,y_2,\ldots,y_n)$, $y_i\in\mathbf{Y}_s$.

A proof of Theorem 4 above can be found in Appendix 2.

In the same way of the classical SVM, the biased variable $b^{\phi}$ in the kernel space can be formulated as

$$b^{\phi}=-\frac{1}{2}\left(\frac{1}{|\mathbf{X}_{s+}|}\sum_{\boldsymbol{x}\in\mathbf{X}_{s+}}\sum_{j=1}^{n+m}\beta_j k_\sigma(\boldsymbol{x}_j,\boldsymbol{x})+\frac{1}{|\mathbf{X}_{s-}|}\sum_{\boldsymbol{x}\in\mathbf{X}_{s-}}\sum_{j=1}^{n+m}\beta_j k_\sigma(\boldsymbol{x}_j,\boldsymbol{x})\right). \tag{16}$$

### 3.2.2. Projected maximum scatter discrepancy between domains

In this section, we will explain why the projected maximum scatter discrepancy between domains is introduced into RKHS embedding domain distribution distance measure. By considering the kernel bandwidth on RKHS embedding domain distribution distance measure, we have the following theorem.

**Theorem 5.** [8] *Given a class of Gaussian kernel functions $K_g=e^{-\|\mathbf{x}-\mathbf{z}\|_2^2/2\sigma^2},\boldsymbol{x},\boldsymbol{z}\in\mathbf{R}^d:\sigma\in[\sigma_0,\infty)$ , where $\sigma_0>0$, for any $k_\sigma,k_\tau\in K_g$ and $0<\tau<\sigma<\infty$, $\gamma_{k\sigma}(P,Q)\geq\gamma_{k\tau}(P,Q)$.*

Theorem 5 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 DAKSVM. Hence, an appropriate Gaussian kernel bandwidth (or the parameter $\sigma$) may perhaps significantly improve the performance of RKHS embedding domain distribution consistency estimation. However, when the intra-domain scatter information is limited, the often-used cross-validation approach can not make sure that appropriate kernel parameters $\sigma$ for DAL can be selected. This fact may significantly degrade the generalization performance of SVM for DAL to some extent. More importantly, if the Parzen window estimator with the above Gaussian kernel width $\sigma$ is adopted to estimate the distribution of a dataset with dimensionality reduction applied, $\sigma$ reflecting the scatter of this dataset will tend to become smaller, which indeed conflicts with our hope that the scatter of this dataset should keep unchangeable as much as possible after dimension reduction. In the following analysis, we will see that MMD can not help us degrade such a tendency while GPMDD can do so.

**Proposition 1.** *By observing the projected maximum distribution scatter discrepancy between RKHS embedding domain distributions, the proposed GPMDD metric can help us measure RKHS embedding distribution discrepancy between domains with two different kernel bandwidth (i.e., $\sigma$ and $\sigma/\sqrt{2}$ ). Furthermore, by using the Gaussian kernel $k_{\sigma/\gamma}(\mathbf{x},\mathbf{x}_i)=\exp(-((\|\mathbf{x}-\mathbf{x}_i\|^2)/(2(\sigma/\gamma)^2)))$ with bandwidth $\sigma/\gamma$ and introducing a tunable parameter $\gamma$ $(\geq 1)$ in the projected maximum distribution scatter discrepancy in GPMDD, one can adaptively achieve the maximal scatter consistency between domains.*

The detailed proof of Proposition 1 can be found in Appendix 3. More importantly, from multiple kernel perspective, we can see from the proof of Proposition 1 that the proposed GPMDD metric in essence plays a multi-kernel role in a specific way of combination of the Gaussian kernels with bandwidths $\sigma$ and $\sigma/\gamma$. As we may know well, multiple kernel techniques can often help us improve the performance of a learner [49,55]. The above fact indeed tells us that the power of the proposed methods essentially benefits from its *natural* multi-kernel learning capability.

By Proposition 1, the matrix $\boldsymbol{\Omega}$ in Theorem 3 can be reformulated as $\overline{\boldsymbol{\Omega}}=(1-\lambda)\boldsymbol{\Omega}_1^{(\sigma)}+\lambda\boldsymbol{\Omega}_2^{(\sigma/\gamma)}$, where $\boldsymbol{\Omega}^{(\bullet)}$ denotes the kernel matrix $\boldsymbol{\Omega}$ with kernel bandwidth $\bullet$. We can see from $\overline{\boldsymbol{\Omega}}$ that the matrix $\boldsymbol{\Omega}$ in DAKSVM can be tuned by parameter $\gamma$, thus further improving the adaptation capability of the proposed method for different DAL problems. Thereby, the dual of DAKSVM in Theorem 3 in a kernel feature space can be reformulated as follows:

$$\min_{\beta,\xi,b} f=\frac{1}{2}\boldsymbol{\beta}^T\overline{\boldsymbol{\Omega}}\boldsymbol{\beta}+C\sum_{i=1}^{N}\xi_i,$$

$$\text{s.t. } y_i\left(\sum_{j=1}^{N}\beta_j k_{\sigma/\gamma}(\mathbf{x}_i,\mathbf{x}_j)+b\right)\geq 1-\xi_i,\quad i=1,\ldots,n,$$

where $\boldsymbol{x}_i\in\mathbf{X}_s$, $\boldsymbol{x}_j\in\mathbf{X}_s\cup\mathbf{X}_t$, and $\xi_i\geq 0$.

According to the above analysis, GPMDD on RKHS embedding domain distributions can sufficiently preserve the discriminative structure consistency between both domains by a tunable kernel bandwidth $\sigma/\gamma$, thus accelerating the convergence of the proposed algorithm and improving the effectiveness of the proposed method for DAL.

### 3.2.3. The algorithm

The proposed DAKSVM algorithm can be summarized as follows.

**Algorithm 1.** DAKSVM

*Input*: data set matrix $\mathbf{X}=(\boldsymbol{x}_i,y_{i\,i=1}^{\,n},\boldsymbol{z}_{j\,j=1}^{\,m})$, $x_i\in\mathbf{X}_s$, $y_i\in\mathbf{Y}_s$, $z_j\in\mathbf{X}_t$.
Set Gaussian kernel bandwidths $\sigma$, $\sigma/\gamma$ respectively in $\gamma_{KM}$ and $\gamma_{KS}$ of GPMDD.
*Output*: decision function $f(x)$.
*Step* 1: determine the parameter$\gamma$ in $\gamma_{KS}$ of GPMDD such that the scatter consistency between source and target domains is maximized.

*Step* 2: compute the matrices $\boldsymbol{\Omega}_1$ and $\boldsymbol{\Omega}_2$ by Eqs. (28) and (29) in Appendix 1, respectively. In terms of $\lambda$ given by users, construct the matrix $\boldsymbol{\Omega}=(1-\lambda)\boldsymbol{\Omega}_1+\lambda\boldsymbol{\Omega}_2$.
*Step* 3**:** for the given $C$, find the optimal vector $\boldsymbol{\beta}$ by applying Theorem 4 to solve the corresponding dual. Then, recover the optimal normal vector $\boldsymbol{w}$ and bias $b^{\phi}$ by $\boldsymbol{\beta}$.
*Step* 4**:** output the decision function $f(\boldsymbol{x})=\boldsymbol{w}^T\phi(x)+b^{\phi}$.

### 3.3. *Variants and extensions*

#### 3.3.1. *Least-square DAKSVM*

One variant of DAKSVM is the least-square DAKSVM which is based on the idea of LS-SVM [24,40], which can be formulated as:

$$\underset{\boldsymbol{w},b,\xi}{\operatorname{argmin}} f = \frac{C}{2}\sum_{i=1}^{n}\xi_i^2+\gamma_{KMS}(p,q)$$

s.t.

$$(\boldsymbol{w},\phi(x_i))+b=y_i-\xi_i, \quad i=1,2,\ldots,n. \tag{17}$$

Along the same line of DAKSVM, the primal of Eq. (17) is defined as

$$\min_{\boldsymbol{\beta},\xi,b} f = \frac{1}{2}\boldsymbol{\beta}^T\overline{\boldsymbol{\Omega}}\boldsymbol{\beta}+\frac{C}{2}\sum_{i=1}^{n}\xi_i^2, \tag{18}$$

$$\text{s.t.} \sum_{j=1}^{n+m}\beta_j k_{\sigma/\gamma}(\boldsymbol{x}_i,\boldsymbol{x}_j)+b=y_i-\xi_i, \quad \xi_i\geq 0, i=1,\ldots,n. \tag{19}$$

**Theorem 6.** *Analytic solution to binary classification*

Given parameter $\lambda\in[0,1]$, for a binary classification problem, the optimal solution of Eqs. (18) and (19) is equivalent to the linear system of equations with respect to variable $\boldsymbol{\alpha}$ as follows:

$$\begin{bmatrix} 0 & \mathbf{1}_n^T \\ \mathbf{1}_n & \tilde{\boldsymbol{\Omega}} \end{bmatrix}\begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix}=\begin{bmatrix} 0 \\ \mathbf{Y}_s \end{bmatrix}. \tag{20}$$

A proof of Theorem 6 above can be found in Appendix 4.

As for multi-class classification problems, the traditional skills are to decompose a multi-class classification problem into several binary classification problems in one against one (OAO) or one against all (OAA) way. However, these skills suffer from the problem of high computational complexity and imbalance data between classes. As such, we introduce the vector labeled outputs into the solution of LS-DAKSVM, which can make the corresponding computational complexity independent of the number of classes and requires no more computations than a single binary classifier [31]. Furthermore, Szedmak and Shawe-Taylor [31] 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 the training sample $\boldsymbol{x}_i$ $(i=1,\ldots,n)$ belongs to the $k$th class, then the class label of $x_i$ are $\boldsymbol{Y}_i=[\underbrace{0,\ldots,1}_{k},\ldots,0]^T\in\mathbf{R}^c$, where the $k$th element is 1 and all the other elements are 0. Hence, for some multi-class classification problems, the optimal solution of LS-DAKSVM can be formulated as

$$\min_{\beta,\xi,b} f = \frac{1}{2}\tilde{\boldsymbol{\beta}}^T\overline{\boldsymbol{\Omega}}\tilde{\boldsymbol{\beta}}+\frac{C}{2}\sum_{i=1}^{n}\xi_i^2, \tag{21}$$

s.t.

$$\tilde{\boldsymbol{\beta}}^T\mathbf{K}_s+b=Y_i-\xi_i, \quad i=1,\ldots,n, \tag{22}$$

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

**Theorem 7.** *Analytic solution to multi-class classification*

Given parameter $\lambda\in[0,1]$, for a multi-class classification problem, the optimal solution of Eqs. (21) and (22) is equivalent to a linear system of the following equation:

$$\begin{bmatrix} b & \boldsymbol{\alpha} \end{bmatrix}\begin{bmatrix} 0 & 1_n^T \\ 1_n & \tilde{\boldsymbol{\Omega}} \end{bmatrix}=\begin{bmatrix} 0_c & \tilde{\mathbf{Y}}_s \end{bmatrix} \tag{23}$$

where,$0_c=[0,\ldots,0]^T$, $\alpha=[\alpha_1,\ldots,\alpha_n]^T$, $\tilde{\mathbf{Y}}_s=[Y_1,Y_2,\ldots,Y_n]^T$,$\tilde{\boldsymbol{\Omega}}$ is the same as in Theorem 6.

**Proof.** The procedures of this proof are the same as that in Theorem 6.

Theorems 6 and 7 actually provide us the LS-DAKSVM versions for both binary and multi-class classification problems, respectively. It is clearly shown from Eqs. (20) and (23) that LS-DAKSVM keeps the same solution framework for both binary and multi-class cases.

#### 3.3.2. *$\mu$-DAKSVM*

The $\nu$-support vector machine ($\nu$-SVM) [35] is a typical extension of SVM for classification in which Schölkopf et al. introduced a new parameter $\nu$ instead of $C$ in SVM to control the number of support vectors and the training errors. More details about $\nu$-SVM can be found. Hence, as the second variant of DAKSVM based on $\nu$-SVM, termed here as $\mu$-DAKSVM can be formulated as:

$$\min_{\boldsymbol{\beta},\xi,b} f = \frac{1}{2}\boldsymbol{\beta}^T\overline{\boldsymbol{\Omega}}\boldsymbol{\beta}-\mu\rho+\frac{1}{N}\sum_{i=1}^{n}\xi_i, \tag{24}$$

$$\text{s.t. } y_i\left(\sum_{j=1}^{N}\beta_j k_{\sigma/\gamma}(\boldsymbol{x}_i,\boldsymbol{x}_j)+b\right)\geq\rho-\xi_i, \quad i=1,\ldots n, \tag{25}$$

where the variables $N=n+m$, $\rho\geq 0$, $\mu>0$ and $\xi_i\geq 0$ have the same meaning as in $\nu$-SVM. Similar to $\nu$-SVM, the dual of the primal in Eqs. (24) and (25) can be formulated as:

$$\min_{\boldsymbol{\alpha}} \tfrac{1}{2}\boldsymbol{\alpha}^T\mathbf{H}^{\phi}\boldsymbol{\alpha}$$

$$\text{s.t. } 0\leq\alpha_i\leq\frac{1}{N}, \quad i=1,\ldots,n,$$

$$\sum_{i=1}^{n}\alpha_i y_i=0, \quad \sum_{i=1}^{n}\alpha_i\geq\mu,$$

where $\mathbf{H}^{\gamma}=\tilde{\mathbf{Y}}\mathbf{K}_s^T(\overline{\boldsymbol{\Omega}})^{-1}\mathbf{K}_s\tilde{\mathbf{Y}}$, and $\tilde{\mathbf{Y}}=\operatorname{diag}(y_1,y_2,\ldots,y_n)$, $y_i\in\mathbf{Y}_s$.

We are now in a position to state the result that can explain the significance of $\mu$ similar to $\nu$ in $\nu$-SVM.

**Proposition 2.** [36] *Suppose we run $\mu$-DAKSVM with kernel function k on some data sets with the result $\rho>0$. Then:*

(1) *$\mu$ Is an upper bound on the fraction of margin errors (and hence also on the fraction of training errors).*
(2) *$\mu$ Is a lower bound on the fraction of support vectors (SVs).*
(3) *Suppose the data $(\boldsymbol{x}_i,y_i)_{i=1}^n$ were generated i.i.d. from a distribution$P(\boldsymbol{x},y)=P(y|\boldsymbol{x})P(\boldsymbol{x})$ , such that neither $P(\boldsymbol{x},y=1)$ nor $P(\boldsymbol{x},y=-1)$ contains any discrete component. Moreover, suppose the kernel used is analytic and non-constant, with probability 1. $\mu$ Will be asymptotically equal to both the fraction of SVs and the fraction of margin errors.*

Compared with the dual of DAKSVM, $\mu$-DAKSVM's dual has two differences. First, there is an additional constraint $\sum_{i=1}^n\alpha_i\geq\mu$. Second, the linear term $\sum_{i=1}^n\alpha_i$ no longer appears in the dual of $\mu$-DAKSVM. By considering the optimal $\boldsymbol{\alpha}$ being a well defined decreasing function of $C$ in DAKSVM, the connection between DAKSVM and $\mu$-DAKSVM can be revealed by Proposition 3.

**Proposition 3.** *[13] By considering the term $(1/Cn)\sum_{i=1}^{n}\alpha_i$ of DAKSVM being a well defined decreasing function of C, we can have* $\lim_{C\to\infty}\frac{\sum_{i=1}^{n}\alpha_i}{Cn}=\mu_{min}\ge 0$ and $\lim_{C\to 0}\frac{\sum_{i=1}^{n}\alpha_i}{Cn}=\mu_{max}\le 1$.

Then,

(1) $\mu_{max}=2\min(n_+,n_-)/n$ *, where* $n_+$ *and* $n_-$ *are the number of positive and negative samples, respectively.*

(2) *For any* $\mu>\mu_{max}$ *, the dual of* $\mu$*-DAKSVM is infeasible. For any* $\mu\in(\mu_{min},\mu_{max}]$*, the optimal solution set of the dual of* $\mu$*-DAKSVM is the same as that of either one or some DAKSVM where these C form an interval. The optimal objective values of* $\mu$*-DAKSVM are strictly positive. For any* $\mu\in[0,\mu_{min}]$ *, the dual of* $\mu$*-DAKSVM is feasible with zero optimal objective values.*

(3) *If the kernel matrix is positive definite, then* $\mu_{min}=0$ .

In terms of Proposition 3, for a given DAL problem and a given kernel, there is an interval $[\mu_{min},\mu_{max}]$, of feasible values for $\mu$, with $0\le\mu_{min}\le\mu_{max}\le 1$. Hence, this property as well as the conclusions in Proposition 3 can be used for parameter selection in $\mu$-DAKSVM later.

### 3.4. Computational complexity

In terms of Algorithm 1, DAKSVM and its variants can be implemented by using the standard SVM solvers (e.g. LibSVM [13]) with the quadratic form induced by the matrix $\mathbf{\Omega}$ mentioned above, and using the optimal solution to obtain the expansion coefficients by Eqs. (34) and (13)–(15), respectively. It is worth noting that our algorithms compute the inverse of a dense Gram matrix $\mathbf{\Omega}$ which leads to $O((n+m)^3)$ training complexity comparable to SVM. This seems to be impractical for large data sets. However, for highly sparse data sets, for example, in text categorization problems, effective conjugate gradient schemes can be used in a large scale implementation [50]. For the nonlinear case, one may obtain approximate solutions (e.g. using greedy, matching pursuit techniques) where the optimization problem is solved over the span of a small set of basis functions instead of using the full representation in $f(x)=\boldsymbol{w}^T\phi(x)$. Besides, CVM [51] may be an alternative choice in addressing scalability issues occurring in SVM learning. We will study these directions in the near future. The testing complexity of DAKSVM depends on the number of support vectors learned from the training stage. In fact, the proposed DAKSVM model and its variants take less than half a minute to finish the whole prediction process for the testing samples from target domain in most of the experiments in the next section.

## 4. Discussions

### 4.1. Comparing with related works

In general, current domain adaptation methods can be classified into two categories [22,6], namely, instance-based methods and feature-based methods [9]. Instance-based methods assume a common relationship between the class labels 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 [9]. Our methods are most similar to feature-based methods for transfer learning. At present, several domain adaptation algorithms rely on defining new features to seize the correspondence between source and target domains [9,23]. In this way, these two domains appear to have similar distributions, thus enabling effective DAL. Moreover, as features are often highly correlated, careful feature selection could lead to significant accuracy gains [37]. In [23], Blitzer et al. describe a heuristic method for DAL, which exploits unlabeled data from both domains to induce the correspondence among features in the two domains. And then the unlabeled target data are used to induce a good feature representation, which can be referred as feature weighting [23]. In [9], rather than heuristically choosing a common feature representation, Ben-David et al. try to directly learn a new representation which minimizes a bound on the target generalization error. The bound is determined by using source domain labeled and unlabeled samples and target domain unlabeled samples as well, and it is represented in terms of a representation function designed to minimize the domain discrepancy, as well as the classification error. The algorithm aims at jointly minimizing a trade-off between source–target dissimilarity and source-domain training error. In [37], Satpal and Sarawagi present a method for addressing domain adaptation problems that selects a subset of features for which the distance (evaluated in terms of a particular distortion metric) between the source and target distributions is minimized, while maximizing the likelihood of the labeled training data. Bruzzone and Marconcini [6] propose the domain adaptation support vector machine (DASVM), which extends transductive SVM (TSVM) [15] to label unlabeled target samples progressively and simultaneously remove some auxiliary labeled samples. Zhong et al. [45] utilize the kernel discriminative analysis (KDA) to make the marginal distributions from two domains closer by reweighing labeled samples in the source domain for training. Besides, several methods have also been developed for transductive transfer learning such as clustering [14] and co-clustering [18], etc., which consider the local structure of the unlabeled data by utilizing some unsupervised learning methods. Xiang et al. [5] propose a domain transfer learner called BIG for bridging information gap between different domains, which bridges domains using worldwide knowledge for transfer learning. Cross-domain SVM (CD-SVM), proposed by Jiang et al. [54], 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 samples.

Although our methods are very similar to feature-based methods, especially those in [3,37], the difference is that we propose a regularization framework, which aims to avoid overfitting and minimize the generalization error. In particular, our work is different from the previous cross-domain learning methods in [1,5,6,14,49] which have ignored the scatter information between domains. Recall that the intra-domain data distribution structure may include the underlying discriminative information [41,42], which plays a crucial role in classification learning. As we may know well, the cross-validation approach can not choose the kernel parameters very well for small or medium-sized datasets, which may significantly degrade the generalization performance of SVMs. Besides, as for distribution distance measure criterion for cross-domain learning, most existing cross-domain learning algorithms [5,6,14,16,18,23,28,43–45] do not explicitly consider any specific criterion to measure the distribution mismatch of samples between different domains. Even though considering the distribution measure criterion explicitly, these methods [1,3,7,8,38,47,49,53] only use such criterion to minimize the distribution mean mismatching between the source and target domains. However, to be different from the aforementioned methods for DAL problems, we propose a novel distance metric criterion GPMDD, inspired by the idea of MMD but essentially different from that of MMD, on RKHS embedding domain distributions by simultaneously minimizing the discrepancy between both distribution mean and scatter. In addition, our proposed algorithms have such good property as stated in Proposition 1.

As pointed out in [56,57,60], the brute-force domain transfer approach without the selection of source domains may degrade the classification performance of DAL, which is still an open problem termed as negative transfer [60]. Hence, several multiple source domain adaptation methods [56–60] have been recently proposed to learn robust classifiers with training data from multiple source domains. In particular, the works in [57] and its extension in [56] 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. 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 or criterion used to minimize the scatter distribution mismatch between the source and target domains in these methods. Even though only focusing on a single source domain in this paper, we propose a general framework for DAL with respect to the newly introduced GPMDD criterion, which can be readily extended into a framework for multiple source domains adaptation learning in terms of the existing techniques.

Again, recall that in some RKHS, MMD originally proposed by Borgwardt [44] is a distribution distance measure for two-sample statistics test. Though the higher order statistics of the data (e.g., higher order moments of probability distribution) in domains can be recovered by transforming the samples in MMD into a higher dimensional or even infinite dimensional space using some higher order nonlinear kernel mapping function [44], this may lead to higher computational complexity or even "dimensionality curse" problem, which is obviously impracticable for high-dimensional data sets. Besides, to encode higher order statistics in MMD, empirically only when the number of samples in domains is large enough, asymptotic MMD criterion can converge steadily or comparatively fast by decreasing kernel bandwidth as the samples grow [44] and this may also lead to higher computational complexity due to its $O(m^2)$computational complexity [2,44]. In other words, asymptotic MMD may be too conservative to detect the difference between domains when the sample sizes are small [44]. Therefore, according to the analysis described above, we argue that the MMD may not always be very efficient for DAL problems to some extent as demonstrated in Fig. 1(b). Nonetheless, by Proposition 1, the proposed GPMDD can measure the consistency between both RKHS embedding domain distributions and inter-domain discriminative structures, thus significantly improving the generalization capability for DAL. Besides, for some specific DAL problems where the number of samples in domains is usually finite or not large, the need for a large number of samples in MMD may limit its adaptation ability on these problems. However, by Proposition 1 and the experimental results in Section 5, the proposed GPMDD still can steadily converge even with relatively not large training datasets, which actually improves the scalability of DAKSVM.

In addition, as for DAL algorithms, several current works [1,43,46,49,55] are most closely related to DAKSVM here. However, there exist two obviously different aspects between our methods and those methods as elaborated below:

(1) As described before, both LMPROJ [1] and DTSVM [49] (or DTMKL [55]) are all based on MMD measure. To be different from these methods, our methods are based on the proposed GPMDD metric, which extends MMD by considering both projected maximum mean discrepancy and projected maximum scatter discrepancy. As analyzed above, there intrinsically exist several drawbacks in the MMD measure. Hence, those MMD-based methods such as LMPROJ and DTSVM (or DTMKL) inevitably inherit the short-comings of MMD. Although DTSVM (or DTMKL) uses multiple kernel techniques to circumvent the shortage of MMD, our methods have clearer statistical interpretation from multi-kernel perspective of the proposed GPMDD metric. It is also worthy to note that although DTSVM, in contrast to our methods which has *natural* multi-kernel learning capability (see Section 3.2.2) and *no convex* combination constraint, can obtain comparable DAL performance on several specific DAL problems as demonstrated in our experimental results, its multiple kernel learning is more subtle in the sense of requiring more parameters to be determined and an iterative two-stage minimization with the convex combination constraint. The same conclusion still holds for Chen et al.'s work in [46].

(2) By considering both first-order (or mean) and second-order (or scatter) statistics, Harchaoui et al. [43] recently proposed a modification of the kernel MMD statistics. In this modification, they scale up the feature space mean distance using the inverse within-sample covariance operator and use the kernel Fisher discriminant analysis to test homogeneity. Very recently, Chen et al. [46] proposed a nonparametric distance metric measure (LSM), which also jointly considers the empirical mean (location) and sample covariance (scatter) differences by using an improved symmetric Stein's loss function to combine the mean and covariance discrepancy into a unified Bregman matrix divergence for DAL. Though our method DAKSVM is also based on both first-order (or mean) and second-order (or scatter) statistics, our key idea is substantially different from these methods in [43,46]. In particular, LSM measures domain distributions location distance as well as scatter distance by Stein loss function while our method DAKSVM does so by the proposed measure GPMDD. DAKSVM is to learn a robust kernel SVM classifier by finding a feature transform in some RKHS such that the distance between the testing and training data distributions measured by GPMDD is minimized while at the same time the class separation distance or classification performance for the training data is maximized. DAKSVM based on the classical SVM learning framework can efficiently inherit all advantages underlying SVM (e.g., kernel skill, convex optimization and sparsity solution). Moreover, by Proposition 1, the proposed methods can adaptively tune the consistency of both RKHS embedding domain distributions distance and projected distribution discriminative information gap between domains by introducing a tunable parameter $\gamma$ for relatively not large training samples.

In summary, in contrast to the state-of-the-art methods mentioned above, DAKSVM is a unified cross-domain kernel machine learning framework, in which the kernel classifier is learned in some RKHS by explicitly minimizing both mean and scatter distribution mismatch between source and target domains where only labeled samples from source domain are required. More importantly, three kernel learning machines (e.g., SVM, $\nu$-SVM, and LS-SVM) can be readily embedded into our DAKSVM framework to highlight its adaptability for various cross-domain learning problems.

### 4.2. *Target generalization bounds for DAL*

Given a DAL problem for binary classification, let $\mathbf{X}$ be the data set in the domain with distribution $P(\boldsymbol{x})$, where $\boldsymbol{x} \in \mathbf{X}$, $f$:$\mathbf{X} \rightarrow [0,1]$ be a label function, and $\tilde{H} : X \rightarrow 0,1$ be a hypothesis function defined on $\mathbf{X}$. Then, the discrepancy (or called hypothesis risk function $\varepsilon(h,f)$) between hypothesis function $h \in \tilde{H}$ and label function $f$ is defined as [33]:

$$\varepsilon(h,f)=E_{x\sim P}[|h(\boldsymbol{x})-f(\boldsymbol{x})|].$$

For simplicity, let use denote $\varepsilon(h,f)$ by $\varepsilon(h)$ with the corresponding empirical risk function $\overline{\varepsilon}(h)$. Then, $\varepsilon_s(h)$ and $\varepsilon_t(h)$ are the risk functions with the empirical risk functions $\overline{\varepsilon}_s(h)$ and $\overline{\varepsilon}_t(h)$ respectively for the source domain and target domain. Thus, the ideal hypothesis risk should minimize the sum of $\varepsilon_s(h)$ and $\varepsilon_t(h)$, i.e.

$$h^*=\underset{h\in\tilde{H}}{\arg\min}[\varepsilon_s(h)+\varepsilon_t(h)]$$

Let $\lambda^*(h)=\varepsilon_s(h^*)+\varepsilon_t(h^*)$ denote the combined risk of the ideal hypothesis which clearly characterizes adaptability between source and target domains. When the ideal hypothesis performs poorly, it is impossible to learn a good target classifier for target domain by minimizing the source error. For a DAL problem, we expect $\lambda^*(h)$ to be small, and thus we can reasonably approximate the target risk using both source risk and the distance between $D_s$ and $D_t$.

We illustrate the result available in the above case with the following risk bound on the target risk in terms of the source risk, i.e. the distance between the distributions $D_s$ and $D_t$. This bound is essentially a restatement of the main theorem of Ben-David et al. [9], with a slight modification here.

**Theorem 8.** *Risk bound on target domain*

Let $\tilde{H}$ be a hypothesis space of VC-dimension $d$ and $\mathbf{U}_s$, and $\mathbf{U}_t$ be unlabeled samples of size $s$ and $t$, respectively, drawn from $D_s$ and $D_t$, respectively. Then, with probability at least $1-\delta$, for every $h\in\tilde{H}$,

$$\varepsilon_t(h)\le\varepsilon_s(h)+\gamma_{KMS}(f,\mathbf{X}_s,\mathbf{X}_t)+4\sqrt{\frac{2d\log(2s)+\log(4/\delta)}{s}}+\lambda^*(h).$$

It is obvious from Theorem 8 that the risk bound on target domain is relative to $\lambda^*$. When the combined error of the ideal hypothesis is large, there is no classifier that can perform well on both source and target domains. Hence, we cannot expect to find a good target hypothesis by training only on the source domain. On the other hand, for small $\lambda^*$, Theorem 8 shows that both source error and distribution discrepancy are important quantities for computing the target error.

Theorem 8 shows how to relate source risk with target risk. We may now proceed to give a learning risk bound for the empirical risk minimization using both source and target domain data. In order to simplify the presentation of the trade-offs that arise in this scenario, we state the bound in terms of VC dimension.

When the number of samples in target domain is small, as in domain adaptation, minimizing the empirical target risk as well as the source risk may not be the best choice. So, we should analyze learners by minimizing a convex combination of the empirical source and target risks as

$$\overline{\varepsilon}_{\tilde{\alpha}}(h)=\tilde{\alpha}\overline{\varepsilon}_t(h)+(1-\tilde{\alpha})\overline{\varepsilon}_s(h),$$

where $\tilde{\alpha}\in[0,1]$ is a trade-off parameter. We denote $\varepsilon_{\tilde{\alpha}}(h)$ as the corresponding weighted combination of true source and target risks, measured with respect to $D_s$ and $D_t$. There exists a relationship between the true target risk and the true combined risk as follows [9]:

$$|\varepsilon_{\tilde{\alpha}}(h)-\varepsilon_t(h)|\le(1-\tilde{\alpha})(\gamma_{KMS}+\lambda^*(h)),$$

which shows that with $\tilde{\alpha}$ approaching 1, we increasingly rely on the target data, and the distance discrepancy between both domains becomes less.

**Theorem 9.** *Empirical risk bound on target domain [9]*

Let $\tilde{H}$ be a hypothesis space of VC-dimension $d$, $\mathbf{U}_s$ and $\mathbf{U}_t$ be unlabeled data sets of size $s$ and $t$ drawn from $D_s$ and $D_t$, respectively, and $\mathbf{S}$ be a random labeled data set of size $s$ generated by drawing $\tilde{\beta}s$ points from $D_t$ and $(1-\tilde{\beta})s$ points from $D_s$, and labeled according to source domain label function $f_s$ and target domain label function $f_t$. If $\overline{h}\in\tilde{H}$ is an empirical minimizer of the combined risk $\overline{\varepsilon}_{\tilde{\alpha}}(h)$ on $\mathbf{S}$ and $h_t^*=\min_{h\in\tilde{H}}\varepsilon_t(h)$ is the minimizer of the target risk, then with probability at least $1-\delta$, for every $h\in\tilde{H}$,

$$\varepsilon_t(\overline{h})\le\varepsilon_t(h_t^*)+2\sqrt{\frac{\tilde{\alpha}^2}{\tilde{\beta}}+\frac{(1-\tilde{\alpha})^2}{1-\tilde{\beta}}}\sqrt{\frac{d\log(2s)-\log\delta}{2s}}$$
$$+2(1-\tilde{\alpha})\left[\gamma_{KMS}(f,\mathbf{X}_s,\mathbf{X}_t)+4\sqrt{\frac{2d\log(2s)+\log(4/\delta)}{\varepsilon}}+\lambda^*(h)\right],$$

when $\tilde{\alpha}=0$ (i.e., we ignore the target data), the risk bound mentioned in Theorem 9 is identical to that of Theorem 8, but with an empirical estimate for the source risk. However when $\tilde{\alpha}=1$ (i.e., we use only target data), the bound is the standard learning bound using only target data. With the optimal $\tilde{\alpha}$, the bound is always at least as tight as either of these two cases. It should be noted that by choosing different $\tilde{\alpha}$, the bound can effectively trade off the small amount of target data against the large amount of less related source data [9].

### 4.3. Singular matrix problem

It is worthwhile to note that the matrix $\mathbf{\Omega}$ in the proposed algorithm may possibly be singular, i.e., the so-called singular matrix problem. If this case happens, the inverse matrix of $\mathbf{\Omega}$ cannot be obtained and the proposed algorithm will become 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 the matrix $\mathbf{\Omega}$ with a small identity matrix with the same dimension as $\mathbf{\Omega}=(1-\lambda)\mathbf{\Omega}_1+\lambda\mathbf{\Omega}_2+\lambda_0\mathbf{I}$, where $\lambda_0\ge0$ is a tunable parameter and $\mathbf{I}$ is a $(n+m)\times(n+m)$ unit matrix.

## 5. Experimental results

### 5.1. Experiment Settings

To evaluate the effectiveness of the proposed DAKSVM and its extensions for DAL problems, we systematically compare them with several state-of-the-art algorithms on different data sets. Three classes of domain adaptation problems are investigated: (1) a series of two-dimensional synthetic problems having different complexities with a two-moon data set and a random Gaussian data set, (2) several real-world cross-domain text classification problems with different domain adaptation data sets such as 20Newsgroups, Reuters, Email Spam Filtering, web query set and Amazon sentiment reviews set, and (3) a real problem in the context of multi-class classification in intra-domain on face recognition with Yale and ORL datasets. For all these data sets, true labels are available for both source and target domain instances. However, prior information related to the target domain $D_t$ is considered only for an objective and quantitative assessment of the performances of the proposed algorithms.

We construct synthetic data sets (Gaussian and two-moon) to study the performance of the proposed methods and choose real-world data sets to demonstrate their classification performance and the involved parameter analysis. We also carry out a multi-class classification experiment to show the performance of the proposed method LS-DAKSVM in multi-class classification problems. In the sequel, we will first describe the whole experimental setup. In the next sections, we will present in detail the experimental results obtained for each task. In all our experiments, we use standard Gaussian kernel function as $k_{\sigma}(\boldsymbol{x},\boldsymbol{z}) = \exp(-(1/2\sigma^2)\|\boldsymbol{x}-\boldsymbol{z}\|^2)$ for several related kernel methods such as SVM, TSVM, KMM, TCA, LMPROJ, and DTSVM. For multiple kernel learning in DTSVM, according to the setting in [49], we use four Gaussian base kernels with bandwidth equal to $1.2^{\delta}\sigma$, where $\delta$ is set as {0, 0.5, 1, 1.5}. For our methods, we use the parameterized Gaussian kernel as $k_{\sigma/\gamma}(\mathbf{x},\mathbf{x}_i) = \exp(-(\|\mathbf{x}-\mathbf{x}_i\|^2/2(\sigma/\gamma)^2))$ in $\gamma_{KS}$ of GPMDD, where the kernel parameter $\sigma$ can be obtained by minimizing MMD with the most conservative test, which follows the setting in [44]. Empirically, we first select $\sigma$ 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 multi-class classification. The tunable parameter $\gamma$ can be set by minimizing GPMDD with the most optimal target test.

Currently, how to choose the algorithm parameters for the kernel methods still keeps an open and hot topic. In general, the algorithm parameters are manually set. In order to evaluate the performance of the algorithm, a strategy, as pointed out in [36], is that a set of the prior parameters is firstly given and then the best cross-validation mean rate among the set is used to estimate the generalized accuracy. In this work, we have also adopted this strategy. The five-fold cross validation is used on the training set for parameter selection. Finally, the mean of experimental results on the testing data is used for performance evaluation. We chose the overall accuracy AC% (i.e., the percentage of correctly labeled samples over the total number of samples) as the reference classification accuracy measure.

Under this context, the SVMs (such as SVM or $\nu$-SVM, TSVM) were implemented by the state-of-the-art software package such as LIBSVM [13] and the other algorithms were implemented using MATLAB.

## 5.2. *Synthetic datasets*

### 5.2.1. *Random Gaussian datasets*

A series of random Gaussian 2D datasets with different sizes are generated to assess the performance of the proposed distribution distance metric by sufficiently considering the discrepancy minimization of both distribution mean and scatter between source and target domains. In the experiments, we construct samples of size $n$ and $m$ for source and target domains, respectively, with corresponding means $\mu_1$ and $\mu_2$, and variance $\Sigma_1$ and $\Sigma_2$ for both domains, where $n,m \in [10^1,10^6]$.

Firstly, to test on the DAL performance of the proposed methods, i.e., DAKSVM and $\mu$-DAKSVM, and compare them with related methods, i.e., SVM and LMPROJ, we synthetically constructed a dataset of size $m=300$ and $n=300$, with each domain generated according to two different Gaussian distributions with $\mu_1=[-0.1345\ \ 2.9497]$, $\Sigma_1=[13.5742\ \ 14.0050]$, $\mu_2=[0.1419\ \ 2.9497]$, and $\Sigma_2=[4.8217\ \ 0.9409]$, i.e. each distribution has 150 samples. Fig. 2 shows that the DAL performance of four methods: SVM, LMPROJ, DAKSVM, and $\mu$-DKSVM. In the sequel, we also investigate the influence of the parameter $\mu$ on the classification accuracy of $\mu$-DKSVM with the same dataset. Table 1 shows the fractions of the errors and SVs, along with the margins of class separation of $\mu$-DAKSVM.

We next investigate the trade-off between computational cost and performance of our methods on changing sample sizes by comparing with several related methods (such as LMPROJ, DTSVM,KMM,TCA and LSM) on two random Gaussian datasets with means $\mu_1=[-0.1345\ \ 2.9497]$, $\mu_2=[5.1419\ \ 14.9497]$ and variances $\Sigma_1=[13.5742\ \ 17.1050]$, $\Sigma_2=[4.8217\ \ 1.2409]$ for both domains. Average experimental results are plotted in Fig. 3. Note that we do not compare them with TCA on larger sample sets because their work cannot cope with thousands of training and test samples [49].

From Figs. 2 and 3 and Table 1, we can have the following observations:

(1) From Fig. 2, we can observe that SVM can separate the binary class samples in source domain smoothly by considering the margin maximization between positive and negative classes in source domain. However, SVM can not preserve the separation consistency with target domain. LMPROJ outperformed SVM by considering both margin maximization and mean discrepancy minimization between the two distributions of source and target domains. However, since it did not consider the scatter discrepancy minimization between the two distributions of source and target domains, the separation direction of LMPROJ biases in favor of the direction of the mean distribution discrepancy minimization, thus resulting in its deteriorated performance. As an extension to LMPROJ, the proposed DAKSVM and its variation $\mu$-DKSVM not only inherit all the advantages of LMPROJ and meanwhile they still sufficiently consider both the mean distribution discrepancy and the scatter distribution discrepancy between source and target domains, therefore exhibiting better performance than LMPROJ and/or SVM.

(2) In Table 1, we can clearly observe that $\mu$ upper bounds the fraction of errors and lower bounds the fraction of SVs and that the bigger $\mu$ becomes, i.e., allowing more errors, the bigger the margin will be, which is consistent with Proposition 1. Besides, it can be verified from Table 1 that the optimal model parameters $\mu$ plays an important role in terms of classification accuracy of $\mu$-DKSVM, thus impacting on the decision function of $\mu$-DKSVM, so it is essential to test these parameter sets. Thereafter, we apply the cross-validation method on the whole training data to select the model parameters and evaluate the corresponding generalized accuracy.

(3) From Fig. 3, without considering the computational cost, DAKSVM shows the second best overall performance with the dataset size, and LSM is better than DAKSVM. In other words, DAL performance can be significantly improved by simultaneously considering both distribution mean discrepancy and distribution scatter discrepancy between domains, simultaneously. It is worth to note that from Fig. 3(a), the proposed DAKSVM can obtain comparably better performance than KMM, DTSVM and LMPROJ for relatively not large datasets. A possible explanation is that by Proposition 1, the proposed method can adaptively tune the consistency between inter-domain discriminative (scatter information) structures of both domains by a tunable parameter $\gamma$. On the other hand, from Fig. 3(a)–(b), although DTSVM and LSM also obtain good learning performance to certain extent by utilizing multiple kernel skill and scatter information, respectively, their running time is higher than other methods. The computational cost of our method is a little higher than other MMD-based methods such as LMPROJ and KMM due to the additional computational cost of the covariance matrix and its inverse of the data matrix. However, because they ignore the scatter information, LMPROJ and KMM obtain the worst DAL performance compared to other methods. In summary, the proposed method is a promising one in the sense of the trade-off of the DAL performance and computational cost.

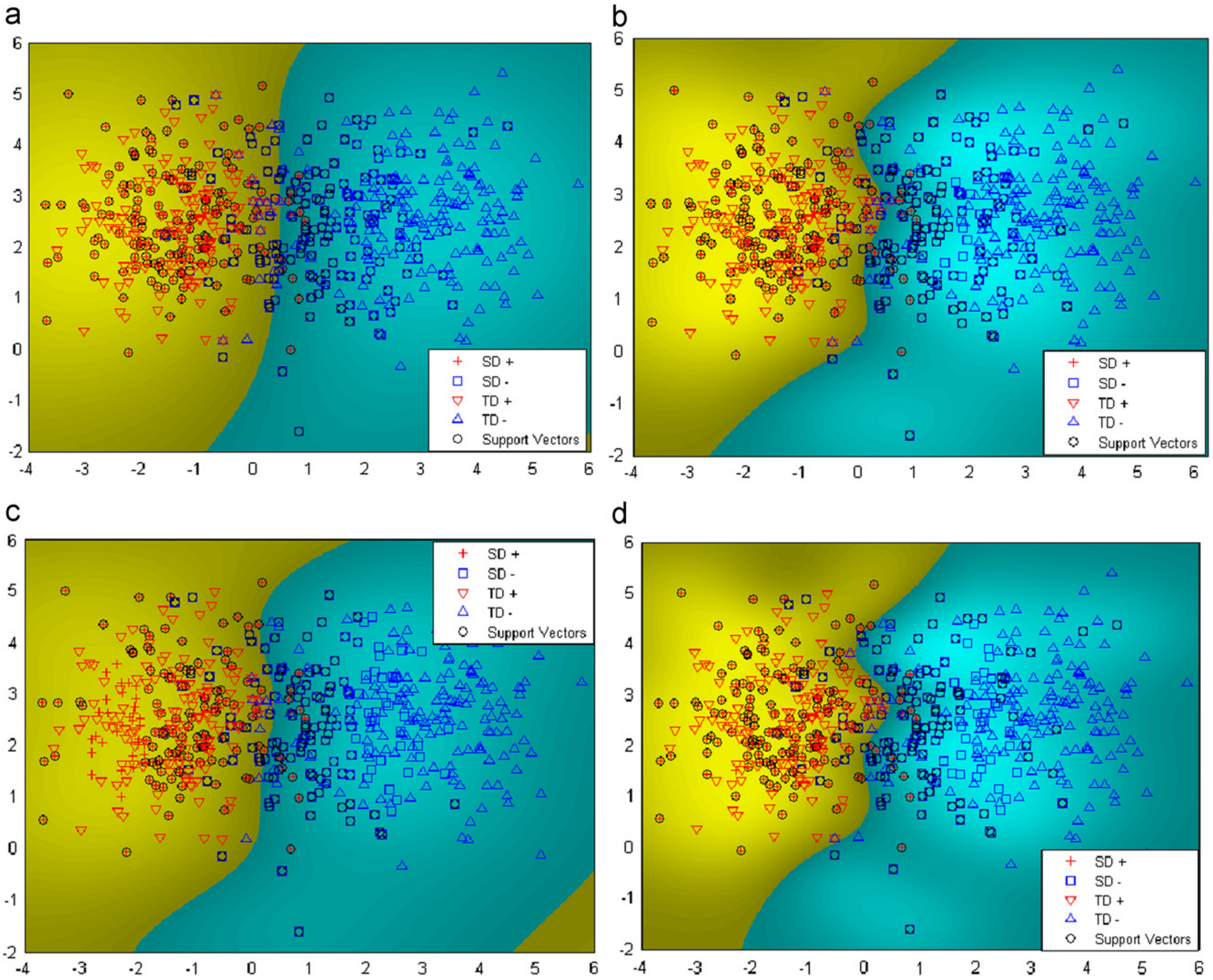

**Fig. 2.** Experimental results of four different classifiers on a random Gaussian data set with fixed sample size. (a) Classification accuracy of SVM: 93.3%, (b) classification accuracy of LMPROJ: 94.6%, (c) classification accuracy of DAKSVM: 97.4%, and (d) classification accuracy of $\mu$-DAKSVM: 97.7%.

**Table 1**
Fractions of errors and SVs, along with the margins of class separation of $\mu$-DAKSVM on random Gaussian dataset with $\mu_1=[-0.1345\ 2.9497]$, $\mu_2=[0.1419\ 2.9497]$, $\Sigma_1=[13.5742\ 14.0050]$, and $\Sigma_2=[4.8217\ 0.9409]$.

| $\mu$ | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|
| Fraction of errors | 0.03 | 0.06 | 0.18 | 0.36 | 0.42 | 0.58 | 0.72 | 0.75 | 0.84 |
| Fraction of SVs | 0.65 | 0.65 | 0.67 | 0.75 | 0.78 | 0.85 | 0.88 | 0.88 | 0.97 |
| Margin $\rho/\lVert w\rVert$ | 0.002 | 0.004 | 0.021 | 0.142 | 0.322 | 0.514 | 0.532 | 0.575 | 0.601 |

### 5.2.2. *Two-moon datasets*

In this section, we construct a series of two-moon datasets trials to justify the basic rational of our method DAKSVM. The two-moon data sets with different complexities are used to evaluate the generalization capability of the proposed DAKSVM on DAL. We compare it with SVM and LMPROJ on this toy data.

We considered as source domain data a synthetic data set composed of 600 samples generated according to a bi-dimensional pattern of two intertwining moons associated with two specific information classes (300 samples each), as shown in Fig. 4(a1). Target data were generated by rotating anticlockwise the original source data set 11 times by 10°, 15°, 20°, 25°, 30°, 35°, 40°, 45°, 50°, 55°, and 60°, respectively. Due to rotation, source and target-domain data exhibit different distributions. Particularly, the greater the rotation angle, the more complex the resultant domain adaptation problem, as confirmed by the values for Jensen–Shannon scatter ($D_{JS}$) [6] shown in Fig. 5(a). The proposed DAKSVM is proved to be particularly effective for solving this kind of problems with high accuracy.

Fig. 4(a2)–(a3) shows the target domain data with the rotation angle 30° and 60°, respectively. Fig. 4(b1)–(b2), (c1)–(c2) and (d1)–(d2) shows the learning accuracy of different methods on the data sets shown in Fig. 4(a2)–(a3). Fig. 5(b) shows the performance comparison among different methods on 11 target data sets mentioned previously. From Figs. 4(b)–(d) and 5(b), we can observe that with appropriate learning parameters, the proposed method can obtain perfect separation between classes even if the rotation angles range from 10° to 50°. Besides, we can also observe several results as follows:

(1) From Fig. 4(b1)–(b2), (c1)–(c2) and (d1)–(d2), we can observe that the accuracies of DAKSVM and LMPROJ are always higher

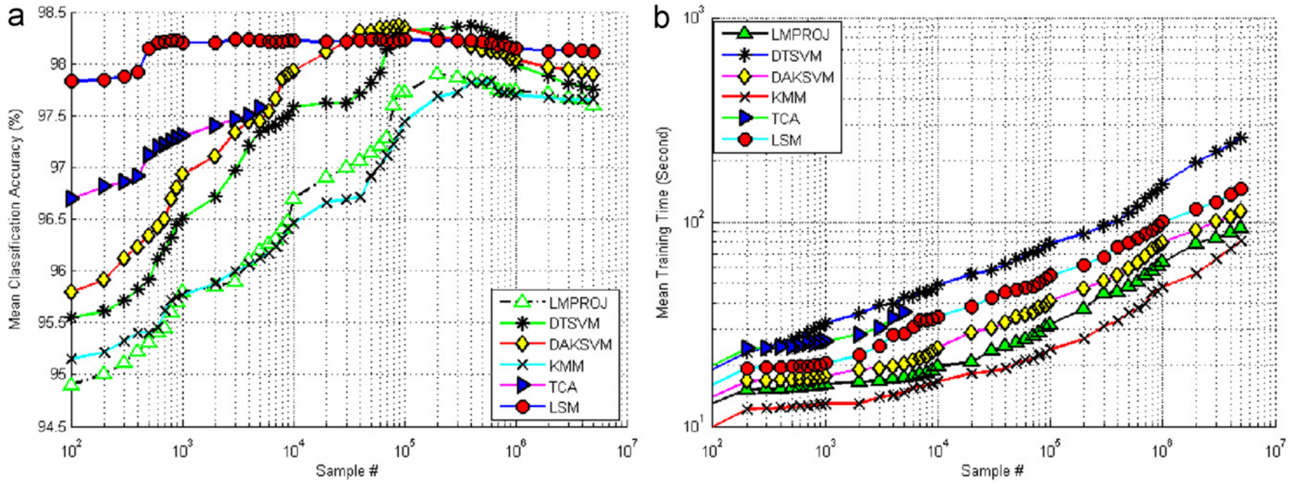

**Fig. 3.** Performance on random Gaussian datasets with changing dataset size (denoted as sample #). (a) DAL classification performance comparison and (b) DAL runtime comparison.

than those by SVM according to a 10-fold Cross-Validation on source-domain data. This result shows that it is unsuitable for SVM on cross-domain learning. With Figs. 4 and 5, in some angles' range (i.e., from 10° to 50°), the proposed method and LMPROJ can preserve the solution consistency well with target domain to some extent, which shows that the proposed method is better than or at least comparable to LMPROJ in this experiment.

(2) Fig. 5(b) shows that for large rotation angles (i.e., from 50° to 60°), the classification accuracy of all methods decrease dramatically, which seems reasonable due to the increase of the complexity of the corresponding domain adaptation problems. However, the rate of decrease of the proposed method is slower than those of other methods due to its capability to preserve the distribution consistency of both means and variances of different domains. When the rotation angle is large enough, all methods will not be able to keep the solution consistency with target domain. If this case happens, the aforementioned hypothesis will not be satisfied.

### 5.3. Binary classification: DAL on text data

In this section, we demonstrate the overall efficiency and effectiveness of the proposed methods DAKSVM and $\mu$-DAKSVM on five different real-world domain adaptation tasks for text datasets such as 20Newsgroups, Reuters, mail spam filtering, web query classification, and Amazon sentiment reviews classification. In addition to SVM, KMM, DTSVM, LMPROJ and the transductive support vector machine (TSVM), we also select the cross domain spectral classifier (CDSC) [16] and locally-weighted ensemble (LWE) classifier [14] for a more comprehensive comparison.

Unlike SVM and TSVM with default parameter values adopted in most cases, we report the best performance of each method over a range of parameter selections to facilitate a fair comparative study.

#### 5.3.1. Characteristics of datasets

A brief description of each dataset and its set-up are given in this section. Tables 2 and 3 summarize the datasets and give the indices to some of which we will refer in our experimental results. For example, dataset 6 is a 20Newsgroup dataset about Rec. vs. Sci. where the number of positive and negative training samples is 1984 and 1977, respectively, and the number of positive and negative class testing samples is 1993 and 1972, respectively.

(1) *20Newsgroups and Reuters*
Reuters and 20Newsgroups are two cross domain text classification datasets commonly used by the state-of-the-art DAL classifiers [1,5,6,14,46,49]. They represent text categorization tasks, with Reuters made up of news articles with 5 top-level categories among which Orgs, Places, and People are the largest, and the 20Newsgroups dataset containing 20 newsgroup categories each of which consist of approximately 1000 documents. For these text categorization data, in each case the goal is to correctly discriminate between articles at the top level, e.g. "sci" articles vs. "talk" articles, using different sets of sub-categories within each top-category for training and testing, e.g. *sci.electronics* and *sci.med* vs. *talk.politics.misc* and *talk.religion.misc* for training and *sci.crypt* and *sci.space* vs. *talk.politics.guns* and *talk.politics.mideast* for testing. For more details about the sub-categories, see [18]. 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. Details about this procedure can also be found in [18]. Table 2 shows more detail information about the experimental datasets drawn from the aforementioned data sets.

(2) *Email spam filtering*
In email spam filtering datasets [52], there are three email subsets (denoted as User1, User2 and User3) 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 different users, the data distributions of the three subsets are different but related. Each subset has 2500 emails, of which one half are non-spam ones (labeled as 1) and the other half are spam one (labeled as $-1$). In this dataset, in terms of [52], we consider three settings: (1) User1 (source domain) and User2 (target domain); (2) User2 (source domain) and User3 (target domain) and (3) User3 (source domain) and User1 (target domain). For each setting, the training dataset contains all labeled samples from the source domain and the samples in the target domain are used as the unlabeled testing ones. We report the experimental results with their means and the standard deviations of all methods. Again, the word-frequency feature is used to represent each document as in [52]. The details about the experimental datasets drawn from email spam filtering datasets can be found in Table 2.

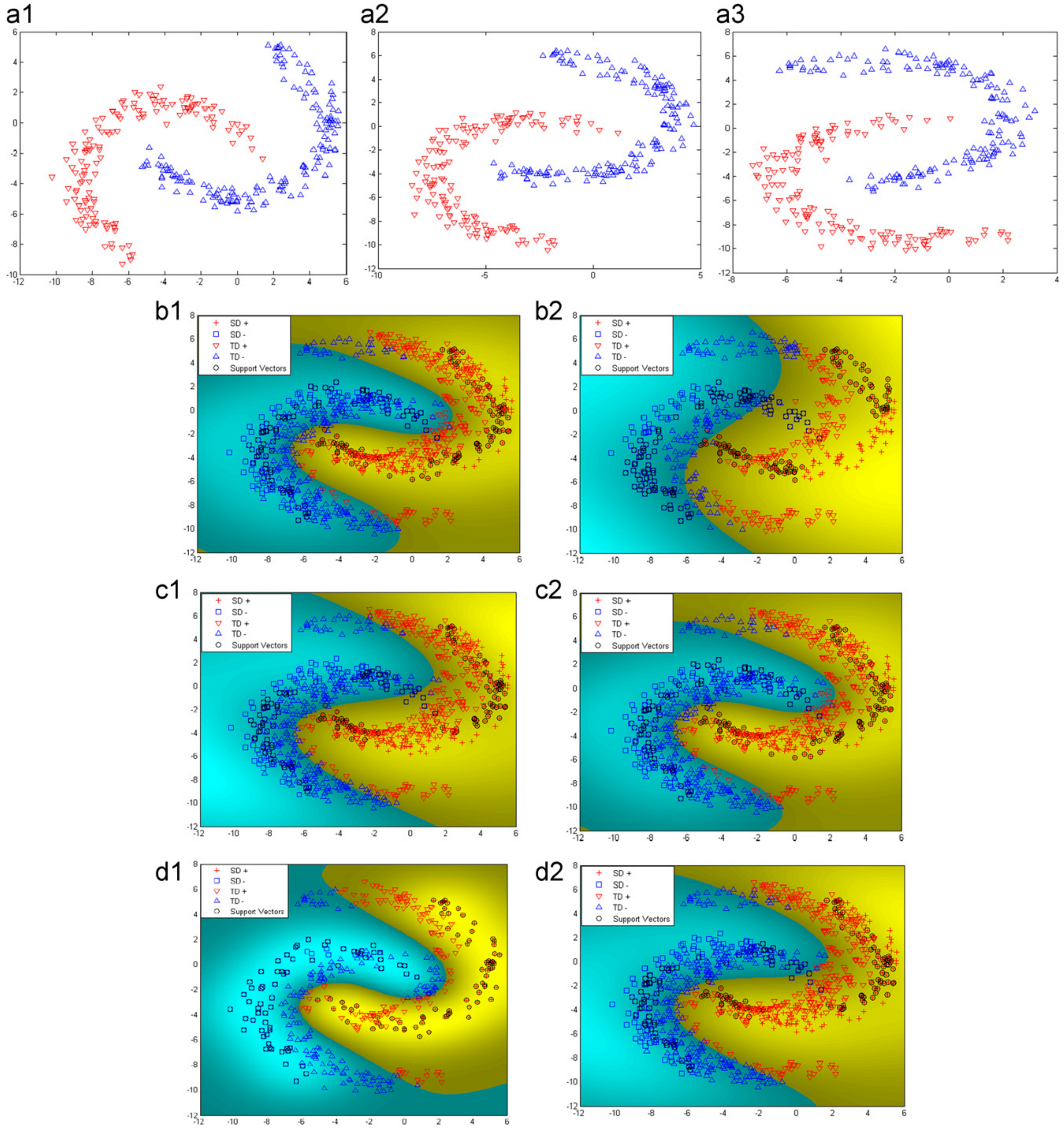

**Fig. 4.** Performance of different classifiers on two two-moon datasets with different complexities. (a1) The original two-moon dataset, (a2) rotated by 30°, (a3) rotated by 60°, (b1) classification accuracy of SVM: 95.4%, (b2) classification accuracy of SVM: 65%, (c1) classification accuracy of LMPROJ: 97.3%, (c2) classification accuracy of LMPROJ: 78.7%, (d1) classification accuracy of DAKSVM: 98.7% and (d2) classification accuracy of DAKSVM: 87.5%.

(3) *Web query*

We also construct a set of tasks on cross-domain query classification for a search engine, e.g. Google. We use a set of search snippets gathered from Google as our training data and some incoming unlabeled queries as the testing data. The detail descriptions of the procedure can be found in [27]. We use the labeled queries from AOL provided by [28] (http://grepgsadetsky.com/aol-data) for evaluation. We consider queries from five classes: *Business*, *Computer*, *Entertainment*, *Health*, *and Sports* which appear in both training and testing datasets. We form 10 binary classification tasks for query classification [5] and the detail information can be seen in Table 3.

(4) *Sentiment reviews*

The data of sentiment domain adaptation [26] consist of Amazon product reviews for four different product types, including books, DVDs, electronics, and kitchen appliances. Each review consists of a rating with scores ranging from 0 to 5, a reviewer name and location, a product name, a review title and date and the review text. Reviews with rating higher than 3 are labeled as positive while reviews with rating lower than 3 are labeled as negative. The rest are discarded since the polarity of these reviews is ambiguous. Details of the data in different domains are summarized in Table 3. The experimental settings are the same as those in [26]. To study the performance of our

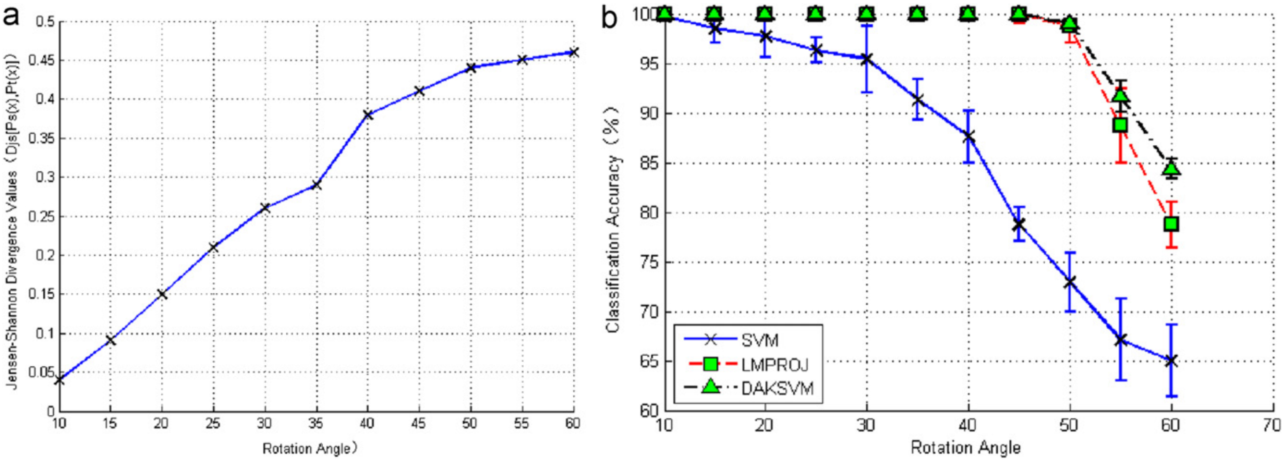

**Fig. 5.** Jensen–Shannon divergence values and classification accuracies on target domain data for different rotation angles. (a) Jensen–Shannon divergence values for different rotation angles ($D_{JS}$) and (b) accuracies exhibited on target domain data for different rotation angles.

**Table 2**
Cross domain text classification tasks.

| Task | | Data sets | Number of training samples | | Number of testing samples | |
|---|---|---|---|---|---|---|
| | | | Positive class | Negative class | Positive class | Negative class |
| 1 | Reuters | Orgs vs. people | Documents from sub-categories | | Documents from different sub-categories | |
| 2 | | Orgs vs. place | | | | |
| 3 | | People vs. place | | | | |
| 4 | 20 Newsgroup | Comp vs. sci | 1958 | 1972 | 2923 | 1977 |
| 5 | | Rec vs. talk | 1993 | 1568 | 1984 | 1658 |
| 6 | | Rec vs. sci | 1984 | 1977 | 1993 | 1972 |
| 7 | | Sci vs. talk | 1971 | 1403 | 1978 | 1850 |
| 8 | | Comp vs. rec | 2916 | 1993 | 1965 | 1984 |
| 9 | | Comp vs. talk | 2914 | 1568 | 1967 | 1685 |
| 10 | Email spam filtering | User1 vs. User2 | User1's emails | | User2's emails | |
| 11 | | User2 vs. User3 | User2's emails | | User3's emails | |
| 12 | | User3 vs. User1 | User3's emails | | User1's emails | |

**Table 3**
Web query text and sentiment reviews classification tasks.

| Task | | Categories | Number of training samples | Number of testing samples |
|---|---|---|---|---|
| 13 | Web query | Business (B) | 1500 | 1200 |
| 14 | | Computers (C) | 1500 | 1000 |
| 15 | | Education (E) | 2210 | 2500 |
| 16 | | Health (H) | 1180 | 1190 |
| 17 | | Sports (S) | 1420 | 660 |
| 18 | Amazon sentiment reviews | Books (B) | 1000 | 1000 |
| 19 | | DVDs (D) | 1000 | 1000 |
| 20 | | Electronics (E) | 1000 | 1000 |
| 21 | | Kitchen (H) | 1000 | 1000 |

methods in this task, we construct 12 pairs of cross-domain sentiment classification tasks as shown in Table 6, e.g., we use the reviews from domain A as the training data and then predict the sentiment of the reviews in the domain B.

### 5.3.2. *Experimental results on text datasets*

Tables 4–6 and Fig. 6 show the means and standard deviations of classification accuracies of different methods on the above DAL tasks. From these results, we can make several interesting observations as follows:

(1) Tables 4–6 summarize the average domain adaptation performance of different methods on all datasets. Our methods have obtained very promising results. The major limitation of LMPROJ, DTSVM and KMM is that they only consider the first-order statistics and thus cannot generalize well. However, our methods definitely consider both the second-order and the first-order statistics between the source and target domains, making them yield better generalization results. It can be observed that our methods, particularly $\mu$-DAKSVM, significantly outperform other methods. These empirical results again show that considering second-order statistics as well as first-order statistics can help us improve the domain adaptation performance.

(2) SVM and TSVM have the worst performance on almost all learning tasks compared to other classifiers, which is consistent with the experimental results of the syntactic datasets above. Though obtaining better classification on both 20Newsgroup and Reuters datasets, TSVM exhibits worse performance on two web text classification tasks than other methods. It is worth noting that we obtain slightly better results for SVM and TSVM than those typically reported in the previous literature (e.g. [1,3,5,49]) on the same datasets used

**Table 4**
Means and standard deviations (%) of classification accuracies (ACC) of all methods on the 20Newsgroups, Reuters data sets, and email spam filtering datasets. Each result in the table is best among all the results obtained using different parameters.

| Datasets | | Methods | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | SVM | TSVM | CDCS | LWE | LMPROJ | DTSVM | KMM | DAKSVM | $\mu$-DAKSVM |
| Reuters | 1 | 80.20* (±0.45) | 81.84* (±1.26) | 88.50 (±4.32) | 83.42* (±2.01) | 84.63* (±2.82) | **88.77* (±2.14)** | 85.43* (±3.02) | 87.64 (±0.42) | 88.13 (±0.50) |
| | 2 | 71.35* (±2.18) | 75.80* (±1.72) | 73.90* (±2.14) | 69.70* (±0.08) | 80.20* (±1.16) | 81.52 (±0.56) | 79.38* (±0.01) | 79.97 (±1.06) | **82.64 (±0.00)** |
| | 3 | 65.36* (±1.67) | 69.80 (±0.46) | 64.00* (±0.32) | 68.52* (±1.49) | 70.80* (±1.38) | 74.12* (±3.26) | 72.19* (2.42) | 74.40 (±1.13) | **74.90 (±0.42)** |
| 20NG | 4 | 72.53* (±3.42) | 76.75* (±0.54) | 69.80* (±0.48) | **85.24 (±1.81)** | 82.52* (±0.86) | 83.52* (±2.74) | 78.11* (±2.80) | 84.68 (±0.63) | 85.02 (±1.08) |
| | 5 | 70.10* (±2.62) | 73.40* (±2.02) | **82.92 (±1.06)** | 78.60* (±0.34) | 79.30* (±2.76) | 80.5* (±2.82) | 79.11 (±1.00) | 82.36 (±0.15) | 82.67 (±0.30) |
| | 6 | 75.40* (±0.51) | 83.90 (±1.14) | 64.00* (±3.1) | 87.20 (±2.10) | 86.34* (±3.10) | **90.23 (±0.82)** | 87.52* (±2.69) | 88.81 (±1.74) | 88.81 (±0.6) |
| | 7 | 78.00* (±0.04) | 81.20* (±0.06) | 70.84* (±1.62) | 75.32* (±0.47) | 84.68 (±2.11) | 84.84 (±1.49) | 81.47* (±3.27) | 85.12 (±0.04) | **85.37 (±0.80)** |
| | 8 | 83.80* (±1.13) | 85.24* (±0.18) | 82.72* (±0.34) | 88.30 (±1.08) | 85.40* (±1.08) | 91.76 (±1.68) | 89.46* (±2.07) | 89.58 (±0.3) | **91.84 (±0.10)** |
| | 9 | 92.70* (±0.40) | 88.74* (±0.70) | 90.20* (±1.16) | 94.00* (±2.06) | 93.43* (±0.79) | 94.13 (±0.4) | 92.50* (±0.78) | 96.72 (±0.60) | **96.78 (±0.20)** |
| Spam filtering | 10 | 96.08 (±0.10) | 96.21 (±0.22) | 83.28* (±0.20) | 93.51* (±0.40) | 93.21* (±0.52) | 96.89 (±0.1) | 96.21 (±0.10) | 96.49 (±0.03) | **97.19 (±0.06)** |
| | 11 | 96.89 (±0.0) | 97.0 (±0.10) | 92.14* (±1.02) | 98.74 (±0.4) | 94.0* (±0.00) | **97.65 (±0.20)** | 97.13 (±0.05) | 97.25 (±0.4) | 97.25 (±0.41) |
| | 12 | 91.7* (±0.6) | 91.80* (±0.3) | 90.02* (±0.3) | 88.78* (±0.01) | 88.79* (±0.24) | 94.50 (±0.60) | 91.8* (±0.00) | 93.20 (±0.20) | **93.85 (±0.10)** |

* The performance of $\mu$-DAKSVM is statistically significant compared with other 7 classifiers at $p$-value $\leq 0.05$.

**Table 5**
Means and standard deviations (%) of classification accuracies (ACC) of all methods on the Web query data set. Each result in the table is best among all the results obtained using different parameters.

| Methods | Datasets | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Web query data | | | | | | | | | |
| | B–C | B–H | B–H | B–S | C–E | C–H | C–S | E–H | E–S | H–S |
| SVM | 82.52* (±1.14) | 85.93* (±0.03) | 90.94* (±2.67) | 87.25 (±0.44) | 87.34 (±4.12) | 93.19* (±3.10) | 84.68* (±2.01) | 92.55* (±0.4) | 81.59* (±0.2) | 93.45* (±1.04) |
| TSVM | 82.64 (±0.2) | 85.42* (±0.2) | 91.19* (±1.06) | 82.60* (±2.50) | 83.35* (±1.52) | 89.70* (±2.00) | 92.01* (±3.24) | 93.11 (±0.03) | 77.93* (±2.44) | 80.84* (±1.60) |
| CDCS | 84.34 (±2.74) | 82.86* (±0.33) | 96.44 (±3.16) | 85.40* (±2.22) | 78.70* (±0.50) | 91.28* (±1.60) | 89.40* (±0.62) | 92.76* (±1.10) | **85.55 (±0.00)** | 91.25* (±0.40) |
| LWE | 83.26* (±0.74) | 86.72 (±3.02) | 93.20* (±1.27) | 82.56* (±0.80) | 85.80 (±2.28) | 93.66 (±1.40) | 84.20* (±4.56) | 94.40* (±0.72) | 79.46* (±3.31) | 94.71* (±0.20) |
| LMPROJ | 84.68 (±0.4) | 86.48* (±2.33) | 94.82* (±1.08) | 85.78* (±0.86) | 88.52* (±3.26) | 92.00* (±1.09) | 86.12* (±4.20) | 95.38* (±2.02) | 82.70* (±1.17) | 95.00* (±2.48) |
| DTSVM | 84.22* (±3.12) | **88.36* (±1.32)** | 96.61 (±0.08) | 86.39* (±2.16) | **90.15 (±0.26)** | 94.08 (±0.40) | 95.16 (±0.82) | 93.08* (±0.00) | 83.29* (±0.42) | 97.33* (±1.28) |
| KMM | 83.92* (±0.74) | 86.86 (±0.30) | 95.12* (±0.6) | 81.22* (±3.06) | 85.52* (±1.90) | 93.00* (±0.01) | 84.82* (±2.70) | 95.11 (±0.02) | 80.10* (±0.2) | 96.32* (±1.42) |
| DAKSVM | 86.36 (±0.40) | 87.82 (±0.56) | 98.23 (±1.02) | 90.46 (±0.03) | 88.78 (±0.00) | 95.10 (±0.6) | 96.94 (±0.90) | 96.81 (±0.12) | 83.20 (±0.62) | **98.27 (±0.2)** |
| $\mu$-DAKSVM | **86.76 (±0.00)** | 87.87 (±0.2) | **98.75 (±0.4)** | **91.02 (±1.01)** | 88.78 (±0.00) | **95.42 (±0.80)** | **97.54** (±0.05) | **96.81** (±0.3) | 84.31 (±0.5) | **98.27 (±0.4)** |

* The performance of $\mu$-DAKSVM is statistically significant compared with other 7 classifiers at $p$-value $\leq 0.05$.

**Table 6**
Means and standard deviations (%) of classification accuracies (ACC) of all methods on the Sentiment reviews data set. Each result in the table is best among all the results obtained using different parameters.

| Methods | Datasets | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sentiment reviews data | | | | | | | | | | | |
| | D–B | E–B | K–B | B–D | E–D | K–D | B–E | D–E | K–E | B–K | D–K | E–K |
| SVM | 71.29* (±2.14) | 67.89* (±1.32) | 67.72* (±0.74) | 75.69* (±3.22) | 67.81* (±0.50) | 71.77* (±4.22) | 68.83* (±3.80) | 69.31* (±0.62) | 78.77* (±0.34) | 73.13* (±2.18) | 73.68* (±4.18) | 80.46 (±0.22) |
| TSVM | 71.52* (±0.72) | 68.27 (±0.40) | 69.96* (±2.18) | 74.84* (±2.50) | 64.51* (±3.58) | 71.63* (±3.04) | 71.13* (±1.48) | 70.52* (±1.22) | 80.16 (±0.30) | 76.14* (±2.54) | 76.57* (±0.96) | 80.99 (±0.12) |
| CDCS | 69.33* (±0.44) | 72.30 (±1.05) | 74.34 (±0.60) | 82.30* (±0.14) | 73.00* (±0.54) | 80.66 (±0.82) | 76.80 (±0.40) | 70.36* (±2.08) | 84.07* (±0.41) | 79.02 (±0.36) | 80.58* (±1.16) | **86.47 (±3.05)** |
| LWE | 74.54* (±1.40) | 69.10 (±0.20) | 77.60 (±1.01) | 78.70* (±0.82) | 69.40* (±2.60) | 78.21* (±1.36) | 78.90* (±0.08) | 75.67 (±1.15) | 84.73 (±0.68) | 78.79* (±1.06) | 83.19* (±0.01) | 79.89 (±2.76) |
| LMPROJ | 76.71* (±0.61) | 71.29* (±1.34) | 75.80* (±1.80) | 83.65* (±1.74) | 73.20* (±0.20) | 81.14* (±1.52) | 75.20 (±0.32) | 77.30 (±2.26) | 81.66* (±0.92) | 80.04 (±0.00) | 85.33 (±0.44) | 81.38* (±0.80) |
| DTSVM | 78.41 (±0.30) | 72.58 (±0.40) | **78.88 (±0.66)** | **86.69 (±1.27)** | 74.62* (±0.25) | 82.49* (±0.50) | 79.21 (±1.14) | 77.54 (±0.05) | 83.97 (±0.20) | **81.72 (±2.02)** | 87.07 (±0.84) | 82.56* (±1.48) |
| KMM | 74.02* (±1.31) | 69.58* (±2.04) | 77.18 (±0.56) | 81.79* (±0.07) | 73.16* (±0.05) | 82.06* (±1.12) | 70.88* (±2.06) | 75.82* (±0.42) | 83.50 (±0.20) | 79.94* (±1.70) | 84.27* (±1.46) | 80.07* (±0.40) |
| DAKSVM | 78.54 (±0.40) | 73.09 (±0.01) | 77.82 (±1.06) | 86.34 (±0.6) | 78.02 (±0.00) | 83.07 (±0.56) | 78.79 (±0.02) | 79.21 (±0.72) | **84.25 (±0.2)** | 81.33 (±0.43) | 88.14 (±0.6) | 84.57 (±0.05) |
| $\mu$-DAKSVM | **78.72 (±0.25)** | **73.74 (±0.56)** | 78.03 (±0.50) | 86.34 (±0.71) | **78.43 (±0.34)** | **84.21 (±0.20)** | **79.36 (±0.18)** | **79.54 (±0.00)** | **84.25 (±0.3)** | 81.68 (±0.08) | **89.36 (±0.82)** | 84.89 (±0.25) |

* The performance of $\mu$-DAKSVM is statistically significant compared with other 7 classifiers at $p$-value $\leq 0.05$.

in our experiments. This is because in our simulations instead of selecting a default parameter on the training data to be performed and in order to make our comparison fair, we report the best results over a set of parameters for SVM and TSVM.

(3) In Tables 4–6 and Fig. 6, we can also observe that although the seven methods, i.e., CDCS, LWE, LMPROJ, DTSVM, KMM, DAKSVM and its variation $\mu$-DAKSVM, exhibit comparable classification capability on all text datasets, the proposed DAKSVM and its variation $\mu$-DAKSVM always keep significantly high classification accuracy in most cases, which implies that it is more stable than other methods, particularly on two web text classification datasets such as web query and sentiment reviews datasets.

(4) The results in Tables 4–6 and Fig. 6 also show that the proposed DAKSVM and its variation $\mu$-DAKSVM perform relatively better than MMD-based methods LMPROJ and KMM in almost all datasets, which justifies that the only emphasis on minimizing distribution mean discrepancy between both domains is far from sufficiency for DAL. 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. Besides, it is worth mentioning that DTSVM also obtains fairly robust performance on almost all datasets by adopting multiple kernel learning scheme. A possible explanation is that multiple kernel learning skill can improve learning capability for DAL.

(5) $\mu$-DAKSVM is obviously superior than DAKSVM in classification accuracy for almost all these datasets, which demonstrates that parameter $\mu$can be used to enhance the generalization capability of DAKSVM. Therefore, we use $\mu$-DAKSVM instead of DAKSVM for the performance evaluation hereafter.

(6) In order to verify whether the proposed methods are significantly better than the other methods, we also performed the paired two-tailed $t$-test [39] on the classification results of the 10 runs to calculate the statistical significance of the proposed method $\mu$-DAKSVM. The smaller the $p$-value, the more significant the difference of the two average results is, and a $p$-value of 0.05 is a typical threshold which is considered to be statistically significant. Thus, in Tables 4–6, if the $p$-value of each dataset is less than 0.05, the corresponding results will be denoted "$*$". Therefore, as shown in Tables 4–6, we can clearly conclude that the proposed $\mu$-DAKSVM significantly outperforms other methods in most datasets.

### 5.4. Multi-class classification: DAL on face recognition

In this section, in order to assess the effectiveness of the proposed methods in multi-class classification problems, we investigate the performance of the proposed methods LS-DAKSVM and $\mu$-DAKSVM for face recognition on two benchmarking face databases, namely, Yale and ORL [29,30]. The Yale face database was constructed at the Yale Center for Computation Vision and Control. There are 165 images of 15 individuals in this database where each person has 11 images. The images demonstrate face variations under different lighting conditions (left-right, center-light, right-light) and facial expressions (normal, happy, sad, sleepy, surprised and wink) with or without glasses. Each image was cropped to have a size of 32 × 32 pixels in our experiment. We randomly select 8 images of each individual to construct the source domain dataset. The ORL database contains 400 images grouped into 40 distinct subjects each of which has 10 different images. The images were captured at different times,

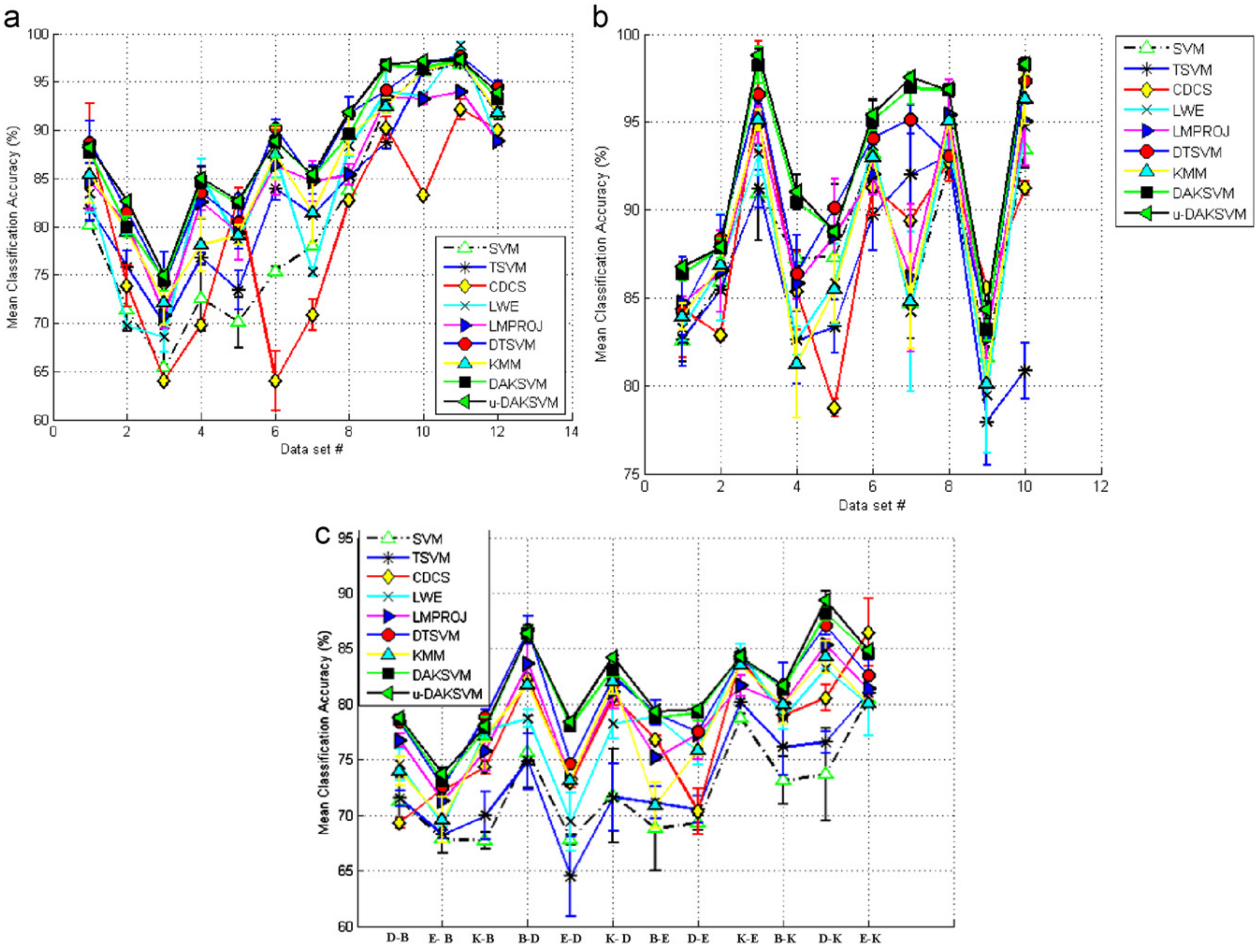

**Fig. 6.** Means and standard deviations (%) of classification accuracies (ACC) of all methods on text datasets. (a) Text datasets: Reuters, 20Newsgroups and mail spam filtering, (b) web query dataset and (c) sentiment classification dataset.

and for some subjects, the images may vary in facial expressions and facial details. All the images were taken against a dark homogeneous background with the tolerance for some side movement of about 20 pixels. The original images are all sized $112 \times 92$ pixels with 256 Gy levels, which are further down-sampled to $32 \times 32$ pixels in our experiment. We randomly select 8 images of each individual to construct the source domain training set. Fig. 7(a) and (c) shows the cropped images of one person in Yale and ORL face databases, respectively.

The target datasets are generated by rotating anticlockwise the original source domain dataset 3 times by $10^\circ$, $30^\circ$, and $50^\circ$, 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. Thus we construct 3 face domain adaptation learning problems for each face database. Fig. 7(b) and (d) shows the face samples with rotation angle $10^\circ$ of the two databases.

We compare the performance of LS-DAKSVM and $\mu$-DAKSVM with CDCS, LWE, DTSVM and LMPROJ. In order to carry out a comprehensive comparison, we also employ the baseline method LS-SVM for face recognition with different distributions. For the above multi-class classification tasks, $\mu$-DAKSVM, CDCS, LWE, LS-SVM, DTSVM and LMPROJ adopt one against one (OAO) multi-class separation strategy to finish the corresponding multi-class classification tasks. For each evaluation, 10 rounds of experiments are repeated with randomly selected training data, and the average result is recorded as the final classification accuracy in Table 7. Several insights can be obtained from these results as follows:

(1) The overall accuracy of LS-SVM is lower than any other classifier in all DAL tasks, which is consistent with SVM.
(2) With the increase in rotation angle, the classification performance of all classifiers descends gradually. However, LS-DAKSVM seems to decrease more slowly than other methods. Exceptionally, CDCS and DTSVM exhibit competitive performance to some extent compared to other methods, particularly on more complex datasets.
(3) As shown in Table 7, the LS-DAKSVM method delivers more stable results across all the datasets and is highly competitive in most of the datasets. It obtains the best classification accuracy more times than any other method. Hence, as discussed in the above section, LS-DAKSVM possesses overall DAL advantages over other methods in the sense of both computational complexity and classification accuracy.
(4) Table 7 also shows that although LS-DAKSVM seems to have overall advantage over $\mu$-DAKSVM in classification accuracy, $\mu$-DAKSVM is actually considerably comparable to LS-DAKSVM.

### 5.5. *Sensitivity analysis of parameters*

In this section, in order to explicitly explain the parameters' influence on the classification performance of the proposed

**Fig. 7.** Face image samples from the Yale and ORL face databases. (a) Yale faces of an object, (b) yale faces of an object with rotation angle 10°, (c) ORL faces of an object and (d) ORL faces of an object with rotation angle 10°.

**Table 7**
Means and standard deviations (%) of classification accuracies (ACC) of all methods on Yale and ORL with different rotation angles. The best results among all the results obtained with different parameters are listed in the table.

| Face data | | Method | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | LS-SVM | LMPROJ | LWE | CDCS | DTSVM | LS-DAKSVM | $\mu$-DAKSVM |
| Yale | 10° | 61.78 ( ± 3.45) | 68.45 ( ± 0.56) | 63.78 ( ± 2.48) | 62.47 ( ± 1.10) | 68.77 ( ± 0.54) | **70.24** ( ± 0.24) | 69.93 ( ± 0.2) |
| | 30° | 58.37 ( ± 2.74) | 64.13 ( ± 1.07) | 61.66 ( ± 1.01) | 60.70 ( ± 0.4) | 67.28 ( ± 2.34) | **66.47** ( ± 0.5) | 66.2 ( ± 0.6) |
| | 50° | 52.29 ( ± 2.12) | 62.08 ( ± 1.18) | 58.78 ( ± 0.41) | 60.20 ( ± 0.34) | **65.63** ( ± 1.14) | 63.00 ( ± 0.4) | 63.7 ( ± 0.34) |
| ORL | 10° | 76.30 ( ± 1.00) | 85.94 ( ± 1.40) | 80.90 ( ± 0.6) | 84.64 ( ± 0.2) | 84.84 ( ± 0.00) | **86.28** ( ± 0.04) | 84.18 ( ± 0.44) |
| | 30° | 70.72 ( ± 3.04) | 82.00 ( ± 0.76) | 79.33 ( ± 1.20) | **83.71** ( ± 2.10) | 83.19 ( ± 0.01) | 83.10 ( ± 1.06) | 83.40 ( ± 0.00) |
| | 50° | 65.70 ( ± 0.62) | 78.65 ( ± 0.20) | 72.22 ( ± 3.54) | 79.91 ( ± 1.03) | 80.01 ( ± 1.14) | **81.46** ( ± 0.02) | 78.10 ( ± 1.68) |

method, we give the experimental results of the sensitivity of three parameters $C$, $\lambda$, and $\gamma$ on accuracy in Fig. 8. In our experiments, we take $\gamma_0=10$ in terms of Theorem 5. In each case, two parameters are fixed at their best values while the third parameter is varied to generate the corresponding figures. Here, we first show the representative results for the three parameters on a couple of datasets, including the 2nd Reuters dataset and the first web query, in Fig. 8(a)–(c). In the sequel, to demonstrate the validity of the conclusion in Proposition 1, we investigate the generalization performance of our method in the cases of fixing and changing the parameter $\gamma$, respectively, denoted by $\gamma=1$ and $\gamma\in[1,\gamma_0]$. To be distinguishable and for simplicity in Fig. 8, we denote DAKSVM by N-DAKSVM in the case of $\gamma=1$, which corresponds to the naive version of DAKSVM (or Naive DAKSVM, N-DAKSVM). Fig. 8(d) shows the learning performance comparison between DAKSVM and N-DAKSVM on the twelve text classification tasks (including Reuters, 20NG, and email spam filtering datasets) in the two cases mentioned above. In summary, from Fig. 8(a)–(d), we can observe several interesting results as follows:

(1) As shown in Fig. 8(a), the proposed method is considerably sensitive to regularization parameter $C$ for a wide range of value. As $C$ varied smoothly, the accuracy of the proposed method significantly changed accordingly. This verifies the importance of $C$ to be tuned.

(2) In Fig. 8(b), it is shown that when $\lambda=0$, i.e., ignoring of distribution scatter discrepancy between source and target domains, the proposed method cannot achieve the optimal performance. As $\lambda$ 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 $\lambda=1$, i.e., ignoring of distribution mean discrepancy between source and target domains, the performance degrades significantly. In terms of the analysis mentioned above, we can conclude that in domain adaptation, learning a classifier by only minimizing the distribution mean distance or scatter distance between two different domains may not be enough, and only when simultaneously considering the distribution mean and scatter distances between two different domains, can we obtain better classification performance.

(3) In Fig. 8(c), as explained in Theorem 5, with a comparatively small $\gamma$ (e.g. $\gamma\in[1,2]$), i.e. a large Gaussian kernel bandwidth, the proposed method tends to have the decrease in the convergence rate of the distribution scatter between two different domains due to the increase of the distribution scatter of intra-domains, while with a comparatively large $\gamma$ (e.g. $\gamma\in[4,+\infty)$), i.e. a small Gaussian kernel bandwidth, the proposed method leads to the class overlapping of intra-domains due to the high cohesion of data distributions of intra-domains. Both cases can lead to degrade the classification performance of the proposed method. Only in its some moderate range of values (e.g.$\gamma\in[1.5,3.5]$), the proposed method achieves relatively better performance.

(4) From Fig. 8(d), we can see that naive DAKSVM cannot obtain the optimal performance on all classification tasks. However, the proposed DAKSVM can gain significantly better

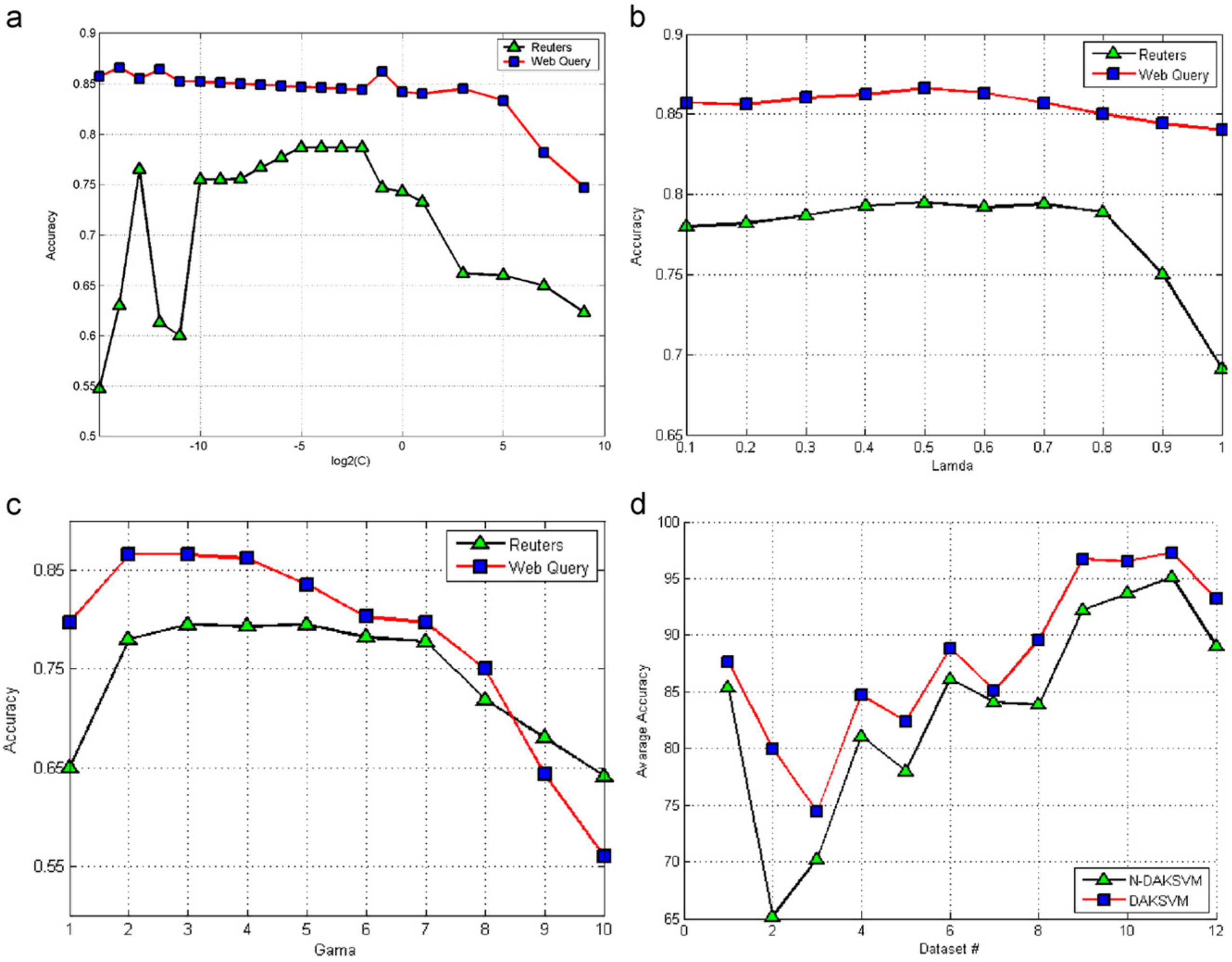

**Fig. 8.** Parameter sensitivity analysis. (a) Sensitivity of parameter $C$, (b) sensitivity of parameter $\lambda$, (c) sensitivity of parameter $\gamma$ and (d) sensitivity of kernel bandwidth tuned by $\gamma$.

performance by adaptively tuning the parameter $\gamma$ on almost all classification tasks. This conclusion further verifies the conclusion in Proposition 1.

## 6. Conclusions and future work

In this paper, we attempt to address domain adaptation learning problems by proposing DAKSVM and its extensions. DAKSVM extends the principle of SVMs to the domain adaptation framework by simultaneously considering both mean and scatter discrepancy between two distributions of source and target domains. Our work here justifies the importance of distribution scatter consistency between domains in DAL study. It is worth noting that the proposed method is designed to address a problem conceptually different from those faced by transductive and semi-supervised SVMs, which have been used 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) testing samples. With extensive experiments on toy and real-world datasets, we demonstrate the effectiveness of the proposed methods and compare them with several state-of-the-art methods. Our results demonstrate the effectiveness of the viewpoint of using such regularization as mentioned in the paper 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 along the following directions. First, we plan to validate the capability of the proposed methods more extensively through other kernel functions or multiple kernel learning. Second, this work only investigates the case where the source, target and auxiliary data sources share the same feature space. We plan to extend our methods for heterogeneous transfer learning [5]. 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 the near future, we also expect to investigate an effective validation strategy to validate the solutions consistent with the source and target domains.

## Acknowledgments

This work was supported in part by the Hong Kong Polytechnic University under Grants 1-ZV5V and G-U724, and by the National Natural Science Foundation of China under Grants 60903100, 60975027, and 61170122, and by the Natural Science Foundation of Jiangsu Province under Grant BK2009067, 2011NSFJS plus its Key Grant, JiangSu 333 expert engineering Grant (BRA2011142), and 2011 Postgraduate Student's Creative Research Fund of

Jiangsu Province under Grant CXZZ11-0483. Finally, the authors wish to acknowledge the anonymous reviewers for their helpful comments.

## Appendix 1. Proof of Theorem 3

**Proof.** Given a nonempty data matrix $\mathbf{X}=(\boldsymbol{x}_{i\,i=1}^{\,n},\boldsymbol{z}_{j\,j=1}^{\,m})$, $x_i\in\mathbf{X}_s$, $z_j\in\mathbf{X}_t$, and let us consider a nonlinear function $\varphi:\boldsymbol{x}\rightarrow\varphi(\boldsymbol{x})$ mapping $\boldsymbol{x}$ in the primal input space into $\varphi(\boldsymbol{x})$ in the feature space. Then, the data matrix $\mathbf{X}$ in the input space can be represented as $\varphi(\mathbf{X})=(\varphi(\boldsymbol{x}_i)_{i=1}^{n},\varphi(\boldsymbol{z}_j)_{j=1}^{m})$ in the feature space. On the analysis of the normal weight vector $w$ in the linear function $f(\boldsymbol{x})=\boldsymbol{w}^T\varphi(\boldsymbol{x})$ in the kernel space, we know that $\boldsymbol{w}$ is related to both source domain samples and target domain samples. And by using Theorem 2, $\boldsymbol{w}$ in the feature space can be formulated as

$$\boldsymbol{w}=\sum_{i=1}^{n}\beta_i\varphi(\boldsymbol{x}_i)+\sum_{j=1}^{m}\beta_j\varphi(\boldsymbol{z}_j), \tag{26}$$

where $\boldsymbol{\beta}=(\beta_1,\ldots,\beta_m,\ldots,\beta_{m+n})^T$ denotes the weight vector. Hence, $\boldsymbol{w}=\varphi(\mathbf{X})\boldsymbol{\beta}$. Thereby, Eq. (3) can be reformulated as

$$\begin{aligned}\gamma_{KM}(f,\mathbf{X}_s,\mathbf{X}_t)&=\boldsymbol{w}^T\left\|\frac{1}{n}\sum_{i=1}^{n}\varphi(\boldsymbol{x}_i)-\frac{1}{m}\sum_{j=1}^{m}\varphi(\boldsymbol{z}_j)\right\|^2\boldsymbol{w}\\&=\left\|\frac{1}{n}\sum_{j=1}^{n+m}\beta_j^T\varphi(\boldsymbol{x}_j)^T\sum_{i=1}^{n}\varphi(\boldsymbol{x}_i)-\frac{1}{m}\sum_{i=1}^{n+m}\beta_i^T\varphi(\boldsymbol{x}_i)^T\sum_{j=1}^{m}\varphi(\boldsymbol{x}_j)\right\|^2=\boldsymbol{\beta}^T\boldsymbol{\Omega}_1\beta,\end{aligned} \tag{27}$$

where $\boldsymbol{\Omega}_1$ is a $(n+m)\times(n+m)$ symmetrical positive semi-definite kernel matrix defined as

$$\boldsymbol{\Omega}_1=\frac{1}{n^2}\mathbf{K}_s[1]^{n\times n}\mathbf{K}_s^T+\frac{1}{m^2}\mathbf{K}_t[1]^{m\times m}\mathbf{K}_t^T-\frac{1}{nm}(\mathbf{K}_s[1]^{n\times m}\mathbf{K}_t^T+\mathbf{K}_t[1]^{m\times n}\mathbf{K}_s^T), \tag{28}$$

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

By the same way, Eq. (4) can be further reformulated as

$$\begin{aligned}&\gamma_{KS}(f,\mathbf{X}_s,\mathbf{X}_t)\\&\quad=\boldsymbol{w}^T\left|\frac{1}{n}\sum_{i=1}^{n}\varphi(\mathbf{x}_i)[\varphi(\mathbf{x}_i)]^T-\frac{1}{m}\sum_{j=1}^{m}\varphi(\mathbf{z}_j)[\varphi(\mathbf{z}_j)]^T\right|\boldsymbol{w}\\&\quad=\left|\frac{1}{n}\sum_{j,k}^{n+m}\beta_j^T\varphi(\mathbf{x}_j)^T\sum_{i=1}^{n}\varphi(\mathbf{x}_i)[\varphi(\mathbf{x}_i)]^T\beta_k\varphi(\mathbf{x}_k)\right.\\&\quad\left.-\frac{1}{m}\sum_{j,k}^{n+m}\beta_j^T\varphi(\mathbf{x}_j)^T\sum_{i=1}^{m}\varphi(\mathbf{z}_i)[\varphi(\mathbf{z}_i)]^T\beta_k\varphi(\mathbf{x}_k)\right|\\&\quad=\left|\frac{1}{n}\sum_{j,k}^{n+m}\beta_j^T\beta_k\sum_{i=1}^{n}k_\sigma(\mathbf{x}_j,\mathbf{x}_i)k_\sigma(\mathbf{x}_i,\mathbf{x}_k)\right.\\&\quad\left.-\frac{1}{m}\sum_{j,k}^{n+m}\beta_j^T\beta_k\sum_{i=1}^{m}k_\sigma(\mathbf{x}_j,\mathbf{z}_i)k_\sigma(\mathbf{z}_i,\mathbf{x}_k)\right|\\&\quad=\boldsymbol{\beta}^T\left|\frac{1}{n}\mathbf{K}_s\mathbf{K}_s^T-\frac{1}{m}\mathbf{K}_t\mathbf{K}_t^T\right|\boldsymbol{\beta}=\boldsymbol{\beta}^T\boldsymbol{\Omega}_2\boldsymbol{\beta}\end{aligned}$$

where $\boldsymbol{\Omega}_2$ is a $(n+m)\times(n+m)$ symmetrical positive semi-definite kernel matrix, which is defined as

$$\boldsymbol{\Omega}_2=\left|\frac{1}{n}\mathbf{K}_s\mathbf{K}_s^T-\frac{1}{m}\mathbf{K}_t\mathbf{K}_t^T\right|. \tag{29}$$

Let $\boldsymbol{\Omega}=(1-\lambda)\boldsymbol{\Omega}_1+\lambda\boldsymbol{\Omega}_2$. We know that Theorem 3 holds.

## Appendix 2. Proof of Theorem 4

**Proof.** The Lagrangian function corresponding to Eqs. (11) and (12) may be represented as

$$\begin{aligned}L(\boldsymbol{w},\rho,\xi,b,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\eta})&=\frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\Omega}\boldsymbol{\beta}+C\sum_{i=1}^{N}\xi_i\\&\quad-\sum_{i=1}^{n}\alpha_i\left[y_i\left(\sum_{j=1}^{N}\beta_jk_\sigma(\boldsymbol{x}_j,\boldsymbol{x}_i)+b\right)-1+\xi_i\right]-\sum_{i=1}^{n}\eta_i\xi_i,\end{aligned} \tag{30}$$

where $N=n+m$ and $\alpha_i\geq 0,\eta_i\geq 0$ are Lagrange multiplier coefficients. According to K.K.T., we have

$$\frac{\partial L}{\partial b}=-\sum_{i=1}^{n}\alpha_iy_i=0\rightarrow\sum_{i=1}^{n}\alpha_iy_i=0,\quad i=1,\ldots,n \tag{31}$$

$$\frac{\partial L}{\partial\xi_i}=C-\alpha_i-\eta_i=0\rightarrow 0\leq\alpha_i\leq C,\quad i=1,\ldots,n, \tag{32}$$

Substituting Eqs. (31) and (32) into Eq. (30), we obtain the following new Lagrange function:

$$L(\boldsymbol{w},\rho,\xi,b,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\eta},\gamma)=\frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\Omega}\boldsymbol{\beta}-\sum_{i=1}^{n}\alpha_i\left[y_i\sum_{j=1}^{N}\beta_jk_\sigma(\boldsymbol{x}_j,\boldsymbol{x}_i)-1\right]. \tag{33}$$

In Eq. (33), the Lagrange function has to be minimized with respect to the primal variables and maximized with respect to the dual variables $\boldsymbol{\beta}$. We compute the corresponding partial derivatives with respect to $\boldsymbol{\beta}$ and set it to 0, thus obtaining the following equation:

$$\frac{\partial L}{\partial\boldsymbol{\beta}}=0\rightarrow\boldsymbol{\beta}=(\boldsymbol{\Omega})^{-1}\mathbf{K}_s\tilde{\mathbf{Y}}\boldsymbol{\alpha}, \tag{34}$$

where $\tilde{\mathbf{Y}}=\mathrm{diag}(y_1,y_2,\ldots,y_n)$, $\boldsymbol{y}_i\in\mathbf{Y}_s$. By substituting Eq. (34) into Eq. (33) and by Eqs. (31) and (32), we can derive Theorem 4.

## Appendix 3. Proof of Proposition 1

**Proof.** As the proposed metric GPMDD consists of two metrics, namely, the projected mean metric and the projected scatter metric on RKHS embedding distributions between domains, we discuss them accordingly. First, by Definition 3, the projected scatter metric on RKHS embedding distributions between domains can be defined as

$$\mathbf{w}^T\left|\frac{1}{n}\sum_{i=1}^{n}\varphi(\mathbf{x}_i)\varphi(\mathbf{x}_i)^T-\frac{1}{m}\sum_{j=1}^{m}\varphi(\mathbf{z}_j)\varphi(\mathbf{z}_j)^T\right|\mathbf{w}. \tag{35}$$

In terms of Eq. (35), with the Parzen window kernel density estimator, we can have the following observation:

$$\begin{aligned}&\int\varphi(\mathbf{x})^T\left|\frac{1}{n}\sum_{i=1}^{n}\varphi(\mathbf{x}_i)\varphi(\mathbf{x}_i)^T-\frac{1}{m}\sum_{j=1}^{m}\varphi(\mathbf{z}_j)\varphi(\mathbf{z}_j)^T\right|\varphi(\mathbf{x})dx\\&\quad=\int\left|\frac{1}{n}\sum_{i=1}^{n}k_\sigma(\mathbf{x},\mathbf{x}_i)k_\sigma(\mathbf{x}_i,\mathbf{x})-\frac{1}{m}k_\sigma(\mathbf{x},\mathbf{z}_j)k_\sigma(\mathbf{z}_j,\mathbf{x})\right|dx\\&\quad=\int\left|\frac{1}{n}\sum_{i=1}^{n}k_{\sigma/\sqrt{2}}(\mathbf{x},\mathbf{x}_i)-\frac{1}{m}\sum_{j=1}^{m}k_{\sigma/\sqrt{2}}(\mathbf{x},\mathbf{z}_j)\right|dx\\&\quad=\int\left|p_1(\mathbf{x},\sigma/\sqrt{2})-p_2(\mathbf{x},\sigma/\sqrt{2})\right|dx,\end{aligned} \tag{36}$$

where $p_k(x,\sigma)$ $(k=1,2)$ denotes the Parzen window kernel density estimator with kernel bandwidth $\sigma$ for different domains. In terms of

Eq. (36), we can see that the projected scatter metric on RKHS embedding distributions between domains essentially measures the difference of two corresponding distributions with kernel bandwidth $\sigma/\sqrt{2}$.

Next, by the same way and Definition 2, we can obtain the projected mean metric on RKHS embedding distributions between domains as follows:

$$\mathbf{w}^T \| \frac{1}{n}\sum_{i=1}^{n} \varphi(\mathbf{x}_i) - \frac{1}{m}\sum_{j=1}^{m} \varphi(\mathbf{z}_j) \|^2 \mathbf{w}$$

$$= \mathbf{w}^T \left( \frac{1}{n}\sum_{i=1}^{n} \varphi(\mathbf{x}_i) - \frac{1}{m}\sum_{j=1}^{m} \varphi(\mathbf{z}_j)) \right) \left( \frac{1}{n}\sum_{i=1}^{n} \varphi(\mathbf{x}_i) - \frac{1}{m}\sum_{j=1}^{m} \varphi(\mathbf{z}_j) \right)^T \mathbf{w}$$

With the Parzen window kernel density estimator, we can have the following observation:

$$\int \varphi(\mathbf{x})^T \| \frac{1}{n}\sum_{i=1}^{n} \varphi(\mathbf{x}_i) - \frac{1}{m}\sum_{j=1}^{m} \varphi(\mathbf{z}_j) \|^2 \varphi(\mathbf{x}) dx$$

$$= \int \varphi(\mathbf{x})^T \left( \frac{1}{n}\sum_{i=1}^{n} \varphi(\mathbf{x}_i) - \frac{1}{m}\sum_{j=1}^{m} \varphi(\mathbf{z}_j) \right) \left( \frac{1}{n}\sum_{i=1}^{n} \varphi(\mathbf{x}_i) - \frac{1}{m}\sum_{j=1}^{m} \varphi(\mathbf{z}_j) \right)^T \varphi(\mathbf{x}) dx$$

$$= \int \left( \frac{1}{n}\sum_{i=1}^{n} k_\sigma(\mathbf{x},\mathbf{x}_i) - \frac{1}{m}\sum_{j=1}^{m} k_\sigma(\mathbf{x},\mathbf{z}_j) \right)^2 dx$$

$$= \int (p_1(\mathbf{x},\sigma) - p_2(\mathbf{x},\sigma))^2 dx \tag{37}$$

From Eq. (37), we can see that the projected mean metric on RKHS embedding distributions between domains essentially measures the difference of two corresponding distributions with kernel bandwidth $\sigma$ rather than $\sigma/\sqrt{2}$ in Eq. (36). Let us recall the common knowledge that **w** in SVM in essence plays a role of dimensionality reduction and thus $\sigma$ reflecting the scatter of a dataset should tend to become smaller for its dimension-reduced dataset by **w**. In terms of Eq. (36), the projected scatter metric on RKHS embedding distributions between domains can just reflect such a tendency in the sense of $\sigma/\sqrt{2}$. This fact demonstrates that the projected mean metric is insensitive to the inter-domain discriminative (or scatter) structure variation and thus the projected scatter metric on RKHS embedding distributions between domains can provides us more observable information, which is different from those provided by the projected mean metric on RKHS embedding distributions between domains. Furthermore, in order to reflect such a tendency more, we can introduce a tunable parameter $\gamma$ ($\geq 1$) to control the kernel bandwidth as $\sigma/\gamma$ in Eqs. (35) and (36) such that the scatter discrepancy between domains can be adaptively degraded a lot. In summary, because the proposed GPMDD is the linear convex combination of the above metrics, Proposition 1 about GPMDD holds.

## Appendix 4. Proof of Theorem 6

**Proof.** The Lagrangian function corresponding to Eq. (18) Eq. (19) can be formulated as

$$L_\lambda(\boldsymbol{w},b,\xi,\boldsymbol{\alpha}) = \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 + \frac{1}{2}\boldsymbol{\beta}^T\overline{\boldsymbol{\Omega}}\boldsymbol{\beta} - \sum_{i=1}^{n} \alpha_i \left( \sum_{j=1}^{n+m} \beta_j k_{\sigma/\gamma}(\boldsymbol{x}_j,\boldsymbol{x}_i) + b + \xi_i - y_i \right), \tag{38}$$

where $\alpha_i$ is Lagrangian multiplier. We compute the corresponding partial derivatives with respect to optimal variables and set them to 0, thus obtaining the following equations:

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = 0 \rightarrow \boldsymbol{\beta} = (\overline{\boldsymbol{\Omega}})^{-1}\mathbf{K}_s\boldsymbol{\alpha}, \tag{39}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{n} \alpha_i = 0, \tag{40}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \rightarrow \xi_i = \frac{\alpha_i}{C}, \tag{41}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow \boldsymbol{\beta}^T\mathbf{K}_s + b + \xi_i = y_i. \tag{42}$$

By using Eqs. (40)–(43) and eliminating variables $\boldsymbol{\beta}$ and $\xi_i$, we have:

$$\begin{bmatrix} 0 & \mathbf{1}_n^T \\ \mathbf{1}_n & \tilde{\boldsymbol{\Omega}} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{Y}_s \end{bmatrix}, \tag{43}$$

where $1_n = [1,\ldots,1]^T$, $\alpha = [\alpha_1,\ldots,\alpha_n]^T$, $Y_s = [y_1,\ldots,y_n]^T$, $\tilde{\boldsymbol{\Omega}} = \mathbf{K}_s^T(\overline{\boldsymbol{\Omega}})^{-1}\mathbf{K}_s + (\mathbf{I}_n/C)$, $\mathbf{I}_n$ is a $n$-dimensional identity matrix.

## References

[1] Brian Quanz, Jun Huan, Large margin transductive transfer learning, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM), New York, 2009, pp. 1327–1336.

[2] Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, et al., Hilbert space embeddings and metrics on probability measures, Journal of Machine Learning Research 11 (3) (2010) 1517–1561.

[3] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, Qiang Yang, Domain adaptation via transfer component analysis, IEEE Transactions on Neural Networks 22 (2) (2011) 199–210.

[4] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. Wortman, Learning bounds for domain adaptation, in: Proceedings of the NIPS, 2007.

[5] Evan Wei Xiang, Bin Cao, Derek Hao Hu, Qiang Yang, Bridging domains using world wide knowledge for transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (6) (2010) 770–783.

[6] Lorenzo Bruzzone, Mattia Marconcini, Domain adaptation problems: a DASVM classification technique and a circular validation strategy, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (5) (2010) 770–787.

[7] Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Kernel methods in machine learning, Annals of Statistics 36 (2007) 1171–1220.

[8] Bharath K. Sriperumbudur, Kenji Fukumizu, Arthur Gretton, et al., Kernel choice and classifiability for RKHS embeddings of probability distributions, Advances in Neural Information Processing Systems 22 (2010) 1750–1758. (MIT Press).

[9] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, Analysis of representations for domain adaptation, in: Proceedings of the NIPS, 2007.

[10] A. Gretton, Z. Harchaoui, K. Fukumizu, B. Sriperumbudur, A fast, consistent kernel two-sample test, Advances in Neural Information Processing Systems 22 (2010) 673–681. (MIT Press).

[12] M. Belkin, P. Niyogi, V. Sindhwani, P. Bartlett., Manifold regularization: a geometric framework for learning from examples, Journal of Machine Learning Research 7 (2399) . 2434.

[13] C.-C. Chang, C.-J. Lin., Training $\nu$-support vector classifiers: theory and algorithms, Neural Computation 13 (9) (2001) 2119–2147.

[14] J. Gao, W. Fan, J. Jiang, J. Han, Knowledge transfer via multiple model local structure mapping, in: Proceedings of the 14th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2008.

[15] T. Joachims, Transductive inference for text classification using support vector machines, in: I. Bratko, S. Dzeroski (Eds.), Proceedings of ICML-99, 16th International Conference on Machine Learning, Morgan Kaufmann Publishers, 1999, pp. 200–209.

[16] X. Ling, W. Dai, G. Xue, Q. Yang, Y. Yu, Spectral domain transfer learning, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM New York, NY, USA, 2008.

[17] V. Vapnik, Statistical learning theory, John Wiley and Sons, 1998.

[18] Wenyuan Dai, Gui-Rong Xue, Q. Yang, Y. Yu, Co-clustering based classification for out-of-domain documents, in: Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, San Jose, California, USA, August 2007, pp. 210–219.

[19] Y. Wu, Y. Liu, Robust truncated hinge loss support vector machines, Journal of the American Statistical Association 102 (479) (2007) 974–983.

[20] X. Zhu, Semi-Supervised Learning Literature Survey, Technical Report, Department of Computer Science, University of Wisconsin, Madison, 2008.

[22] S.J. Pan, Q. Yang., A survey on transfer learning, IEEE Transactions on Knowledge Data Engineering 22 (10) (2010) 1345–1359.

[23] J. Blitzer, R. McDonald, F. Pereira, Domain adaptation with structural learning, in: Proceedings of the Conference on Empirical Methods Natural Language, Sydney, Australia, July 2006, pp. 120–128.
[24] T. Kanamori, S. Hido, M. Sugiyama, A least-squares approach to direct importance estimation, Journal of Machine Learning Research 10 (2009) 1391–1445.
[25] A.J. Smola, A. Gretton, L. Song, B. Schölkopf, A Hilbert space embedding for distributions, in: Proceedings of the 18th International Conference on Algorithmic Learning Theory, Sendai, Japan, October 2007, pp. 13–31.
[26] J. Blitzer, M. Dredze, F. Pereira, Biographies, bollywood. Boom-boxes and blenders: domain adaptation for sentiment classification, in: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '07), June 2007, pp. 440–447.
[27] X.H. Phan, M.L. Nguyen, S. Horiguchi, Learning to classify short and sparse text and web with hidden topics from large-scale data collections, in: Proceedings of the 17th International Conference on World Wide Web (WWW '08), April 2008, pp. 91–100.
[28] S.M. Beitzel, E.C. Jensen, O. Frieder, D.D. Lewis, A. Chowdhury, A. Kolcz, Improving automatic query classification via semi-supervised learning, in: Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM '05), November 2005, pp. 42–49.
[29] Jun Gao, Shi-Tong Wang, Zhao-Hong Deng, Global and local preserving based semi-supervised support vector machine, Acta Electronica Sinica 38 (7) (2010) 1626–1634. (in Chinese).
[30] Deng Cai, Xiaofei He, Jiawei Han, Orthogonal Laplacianfaces for face recognition, IEEE Transactions on Image Processing 15 (11) (2006) 3608–3614.
[31] S. Szedmak, J. Shawe-Taylor, Muticlass Learning at One-class Complexity, Technical Report No. 1508, School of Electronics and Computer Science, Southampton, UK, 2005.
[32] B. Scholkopf, R. Herbrich, A.J. Smola, A generalized representer theorem, in: Proceedings of the COLT'2001, Springer Press, Amsterdam, 2001, pp. 416–426.
[33] Yishay Mansour, Mehryar Mohri, Afshin Rostamizadeh, Domain adaptation: learning bounds and algorithms, COLT, 2009.
[35] B. Schölkopf, A.J. Smola, R. Williamson, P.L. Bartlett., New support vector algorithms, Neural Computation 12 (5) (2000) 1207–1245.
[36] L.G. Abril, C. Angulo, F. Velasco, J.A. Ortega, A note on the bias in SVMs for multi-classification, IEEE Transactions on Neural Networks 19 (4) (2008) 723–725.
[37] S. Satpal, S. Sarawagi, Domain adaptation of conditional probability models via feature subsetting, in: Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, 2007.
[38] J. Huang, A. Smola, A. Gretton, K.M. Borgwardt, B. SchÄolkopf, Correcting sample selection bias by unlabeled data, in: Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, 2006.
[39] E. Alpaydin, Introduction to machine learning, The MIT Press, Cambridge, MA, USA, 2004.
[40] J.A.K. Suykens, J. Vandewalle., Least squares support vector machine classifiers, Neural Processing Letters 9 (3) (1999) 293–300.
[41] Xiaoming Wang, Fu-lai Chung, Shitong Wang, On minimum class locality preserving variance support vector machine, Pattern Recognition 43 (2010) 2753–2762.
[42] S. Zafeiriou, A. Tefas, I. Pitas, Minimum class variance support vector machines, IEEE Transactions on Image Processing 16 (10) (2007) 2551–2564.
[43] Z. Harchaoui, F. Bach, E. Moulines, Testing for homogeneity with kernel fisher discriminant analysis, in: Proceedings of the NIPS 20, MIT Press, 2008.
[44] K.M. Borgwardt, A. Gretton, M.J. Rasch, H.-P. Kriegel, B. Schӧlkopf, A.J. Smola, Integrating structured biological data by kernel maximum mean discrepancy, Bioinformatics (ISMB) 22 (14) (2006) e49–e57.
[45] E. Zhong, W. Fan, J. Peng, K. Zhang, J. Ren, D.S. Turaga, O. Verscheure, Cross domain distribution adaptation via kernel mapping, in: Proceedings of the KDD, 2009, pp. 1027–1036.
[46] Bo Chen, Wai Lam, Ivor W. Tsang, Tak-Lam Wong, Location and scatter matching for dataset shift in text mining, in: Proceedings of the IEEE International Conference on Data Mining (IEEE ICDM 2010), December 2010, Sydney, Australia.
[47] S.J. Pan, J.T. Kwok, Q. Yang, Transfer learning via dimensionality reduction, in: Proceedings of the Association for the Advancement of Artificial Intelligence, 2008, pp. 677–682.
[49] L. Duan, I.W. Tsang, D. Xu, et al., Domain transfer SVM for video concept detection, in: Proceedings of the IEEE International Conference Computer Vision and Pattern Recognition, 2009, pp. 1375–1381.
[50] V. Sindhwani, M. Belkin, P. Niyogi, The geometric basis of semi-supervised learning, in: O. Chapelle, B. Scholkopf, A. Zien (Eds.), Semi-Supervised Learning, 2006 (Book Chapter).
[51] I.W. Tsang, J.T. Kwok, Very large scale manifold regularization using core vector machines, NIPS 2005 Workshop on Large Scale Kernel Machines, 2005.
[52] S. Bickel, ECML-PKDD discovery challenge 2006 overview, in: Proceedings of the ECML/PKDD Discovery Challenge Workshop, 2006.
[53] B. Chen, W. Lam, I. Tsang, T.-L. Wong, Extracting discriminative concepts for domain adaptation in text mining, in: Proceedings of KDD, 2009, pp. 179–188.
[54] W. Jiang et al., Cross-domain learning methods for high-level visual concept classification, in: Proceedings of the ICIP, 2008.
[55] L. Duan, I.W. Tsang, D. Xu, Domain transfer multiple kernel learning, IEEE Transactions on Pattern Analysis and Machine Intelligence (2012) 465–479.
[56] L. Duan, D. Xu, I.W. Tsang, Domain adaptation from multiple sources: a domain-dependent regularization approach, IEEE Transaction on Neural Networks and Learning Systems (2012) 504–518.
[57] Lixin Duan, Ivor W. Tsang, Dong Xuet al., Domain adaptation from multiple sources via auxiliary classifiers, in: Proceedings of the 26th International Conference on Machine Learning (ICML 2009), Montreal, Quebec, June 2009.
[58] Chun-Wei Seah, Ivor W. Tsang, Yew-Soon Ong, 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, Spain, September 2010.
[59] P. Luo, F. Zhuang, H. Xiong, Y. Xiong, Q. He, Transfer learning from multiple source domains via consensus regularization, in: Proceedings of the ACM Conference on Information and Knowledge Management, Napa Valley, CA, October 2008, pp. 103–112.
[60] M.T. Rosenstein, Z. Marx, L.P. Kaelbling., To transfer or not to transfer. in advances in neural information processing systems, MIT Press, Cambridge, MA, 2005.

**Jianwen Tao** received the B.S. degree in 1995 and the M.S. degree in 1998, all in computer science from Huazhong University of Science and Technology. Currently, he is an associate professor and pursuing a Ph.D. degree in the School of Digital Media, Jiangnan University. His current research interests include pattern recognition and information retrieval.

**Fu-Lai Chung** received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1987, and the M.Phil. and Ph.D. degrees from the Chinese University of Hong Kong, Shatin, Hong Kong, in 1991 and 1995, respectively. In 1994, he joined the Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong, where he is currently an Associate Professor. He is also a Guest Professor in the School of Digital Media, Jiangnan University, Wuxi, China. He has published more than 50 journal papers in the areas of soft computing, data mining, machine intelligence, and multimedia. His current research interests include novel feature selection techniques, text data mining, and fuzzy system modeling.

**Shitong Wang** received the M.S. degree in computer science from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1987. In the past several years, he was a Research Scientist at London University and Bristol University, U.K., Hiroshima International University, Japan, and Hong Kong University of Science and Technology and Hong Kong Polytechnic University, Hong Kong. He is currently a Full Professor in the School of Digital Media, Jiangnan University, Wuxi, China. His current research interests include artificial intelligence (AI), neurofuzzy systems, pattern recognition, and image processing. He is the author or coauthor of more than 80 papers in international/national journals, and has authored seven books.