Domain Adaptation from Multiple Sources: A Domain-Dependent Regularization Approach

Lixin Duan, Dong Xu, *Member, IEEE*, and Ivor Wai-Hung Tsang

***Abstract*— In this paper, we propose a new framework called domain adaptation machine (DAM) for the multiple source domain adaption problem. Under this framework, we learn a robust decision function (referred to as *target classifier*) for label prediction of instances from the target domain by leveraging a set of *base classifiers* which are prelearned by using labeled instances either from the source domains or from the source domains and the target domain. With the base classifiers, we propose a new domain-dependent regularizer based on *smoothness assumption*, which enforces that the target classifier shares similar decision values with the relevant base classifiers on the unlabeled instances from the target domain. This newly proposed regularizer can be readily incorporated into many kernel methods (e.g., support vector machines (SVM), support vector regression, and least-squares SVM (LS-SVM)). For domain adaptation, we also develop two new domain adaptation methods referred to as FastDAM and UniverDAM. In FastDAM, we introduce our proposed domain-dependent regularizer into LS-SVM as well as employ a sparsity regularizer to learn a sparse target classifier with the support vectors only from the target domain, which thus makes the label prediction on any test instance very fast. In UniverDAM, we additionally make use of the instances from the source domains as *Universum* to further enhance the generalization ability of the target classifier. We evaluate our two methods on the challenging TRECIVD 2005 dataset for the large-scale video concept detection task as well as on the 20 newsgroups and email spam datasets for document retrieval. Comprehensive experiments demonstrate that FastDAM and UniverDAM outperform the existing multiple source domain adaptation methods for the two applications.**

***Index Terms*— Domain adaptation machine, domain-dependent regularizer, multiple source domain adaptation.**

I. Introduction

IT IS well known that the collection of labeled instances is expensive and time consuming. However, the classifiers learned with a small number of labeled training data are not robust and therefore cannot generalize well. To this end, many domain adaptation methods were recently proposed[1] [1]–[5] to learn robust classifiers with only a few or even no labeled instances from the target domain by leveraging a large amount of labeled training data from other domains (referred to as auxiliary/source domains). These methods have demonstrated that the labeled instances collected from other domains are also useful for classifying the instances from the target domain in many real applications, such as sentiment classification, text categorization, WiFi localization, and video concept detection.

To utilize all training instances from the source and target domains, Blitzer *et al.* [2] proposed a structural correspondence learning algorithm to induce the correspondences among features from different domains. They employed a heuristic technique to select the pivot features that appear frequently in both domains. Daumé III [3] proposed a feature replication (FR) method to augment features for domain adaptation. The augmented features are then used to construct a kernel function for kernel methods. Yang *et al.* [5] proposed the adaptive support vector machine (A-SVM) to learn a new SVM classifier $f^T(\mathbf{x})$ for the target domain, which is adapted from an existing classifier $f^s(\mathbf{x})$ trained with the instances from a source domain.

However, numerous unlabeled instances in the target domain are not exploited in the above domain adaptation methods [1], [3], [5]. As shown in [4] and [6]–[12], such unlabeled instances can also be employed to improve generalization performances. When there are only a few or even no labeled instances available in the target domain, the classifiers can be trained with labeled instances from the source domain. In such an extreme case, several domain adaptation methods [13]–[16] were proposed to cope with the inconsistency of data distribution (such as *covariate shift* [14] or *sampling selection bias* [13]). These methods re-weighted training instances from the source domain by leveraging the unlabeled data from the target domain such that the statistics of instances from both domains are matched.

Recently, several domain adaptation methods [4], [17] were proposed to learn robust classifiers with diverse training data from multiple source domains. Luo *et al.* [17] proposed to maximize the consensus of predictions from multiple sources. However, some source domains may not be useful for knowledge adaptation. The brute-force transfer of knowledge without domain selection may degrade the classification performance in the target domain [18], which is a well-known open problem termed as *negative transfer* [19].

Some researchers also theoretically studied the domain adaptation problem [20]–[25]. By assuming the distribution of the target domain to be a weighted combination of the

Manuscript received June 20, 2010; revised October 6, 2011; accepted October 7, 2011. This work was supported in part by the Singapore A*STAR - Science and Engineering Research Council under Grant 082 101 0018 and the Singapore Nanyang Technological University Academic Research Fund Tier-1 Research under Grant RG15/08.

The authors are with the School of Computer Engineering, Nanyang Technological University, 639798, Singapore (e-mail: S080003@ntu.edu.sg; dongxu@ntu.edu.sg; ivortsang@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2011.2178556

[1]Domain adaptation is different from semisupervised learning (SSL). SSL methods employ both labeled and unlabeled data for better classification, in which the labeled and unlabeled data are assumed to come from the same domain.

source distributions, Mansour *et al.* [23] proved that the loss of the target classifier has an upper bound. Crammer *et al.* [22] assumed that the distributions of multiple sources are the same, but the labelings of the data from different sources may be different from each other. And they derived a bound on the error in the target domain by minimizing the empirical error on the data from any subset of the sources. Ben-David *et al.* [20] introduced a formal model for domain adaptation using a generalization upper bound on the errors of the training data. This bound is based on the distance between the feature distributions of the instances from the source and target domains, which is measured by a so-called $d_{\mathcal{A}}$ distance as introduced in [26]. Following [20], Mansour *et al.* [24] extended the $d_{\mathcal{A}}$ distance [20], [26] by introducing a so-called discrepancy distance to measure the mismatch of data distributions, and they also provided Rademacher complexity bounds for a broad family of loss functions. Blitzer *et al.* [21] proposed uniform convergence bounds by additionally considering a limited number of labeled training data from the target domain. In their recent work [25], Ben-David *et al.* further analyzed three types of assumptions that are widely used in the existing domain adaptation methods, and they also mathematically discussed under which assumptions a domain adaptation method can work. For more details on the theory of the domain adaptation problems, the interested readers may refer to [27].

In this paper, we focus on the setting with multiple source domains, which is referred to as *multiple source domain adaptation*. We propose a new domain adaptation framework called the domain adaptation machine (DAM) to learn a robust decision function (referred to as *target classifier*) for label prediction of instances in the target domain by leveraging a set of prelearned classifiers (referred to as *base classifiers*). In our framework, any types of classifiers such as the standard SVM classifier learned with the labeled instances from the source domains or the FR classifier [3] learned with the labeled instances from the source domains and the target domain can be readily used as base classifiers. Motivated from manifold regularization (MR) [7] and the graph-based multitask learning (MTL) [28]–[30], with the base classifiers, we propose a new domain-dependent regularizer based on smooth assumption, which enforces that the learned target classifier should have similar decision values on the unlabeled instances of the target domain with the prelearned base classifiers from relevant source domains. This newly proposed regularizer can be readily introduced to many kernel methods [31] such as SVM, support vector regression (SVR), least-squares SVM (LS-SVM) and so on, and extend these algorithms to the corresponding domain adaptation methods. To the best of our knowledge, this paper and its initial conference version [4] are the first to introduce this domain-dependent regularizer for domain adaptation.

Under this framework, we develop two new methods referred to as FastDAM and UniverDAM. In FastDAM, we incorporate the proposed domain-dependent regularizer into LS-SVM. We also employ a sparsity regularizer based on the ϵ-insensitive loss to enforce the sparsity of the target classifier with the support vectors only from the target domain such that the label prediction on any test instance is very fast in FastDAM. Recent work [32], [33] has indicated that the instances that do not belong to the positive class and the negative class can be used as an additional data collection called *Universum* [34] to improve the generalization ability of SVM for the binary classification task. However, how to choose/construct Universum is problem-dependent. For example, Weston *et al.* [33] experimentally demonstrated that one can make use of symbols (e.g., uppercase and lowercase letters) as Universum for digit classification. For a more detailed introduction of Universum, we refer the readers to [34]. Note that while the instances from the source domains are with different data distributions when compared with that of the target domain, it is reasonable to assume that the distributions of the samples from the source domain and the target domain should overlap to some extent. We therefore use the instances from the source domains as *Universum* for domain adaptation to further enhance the generalization ability of the target classifier. Specially, we introduce our newly proposed regularizer based on smoothness assumption and another regularizer based on the Universum from source instances into LS-SVM in UniverDAM.

We evaluate our two DAM-based methods in two multiple domain adaptation related applications, i.e., video concept detection and document retrieval. In the video concept detection task, the experimental results on the large TRECVID 2005 dataset demonstrate that our proposed FastDAM significantly outperforms other domain adaptation methods. Moreover, with the utilization of the sparsity regularizer, the prediction of FastDAM is much faster than other domain adaptation methods, making it suitable for the large-scale video concept detection task. In the document retrieval task, we compare our two methods with other baseline methods on the 20 newsgroups and the email spam datasets. The comprehensive experiments on the two datasets also demonstrate the effectiveness of FastDAM and UniverDAM. UniverDAM achieves the best document retrieval performances on both datasets because of the successful utilization of *Universum* (i.e., the instances from multiple source domains).

A preliminary version of this paper appeared in [4] which focused on the FastDAM algorithm using auxiliary classifiers (i.e., SVMs learned with the labeled instances from source domains) as the base classifiers. In this paper, we additionally propose a new method called UniverDAM and discuss the connection between FastDAM and UniverDAM. Moreover, we further employ the FR classifiers [3], which are learned with the labeled instances from the source domains and the target domain, as the base classifiers in our DAM framework and report more experimental results using two new datasets (i.e., 20 newsgroups and email spam) for document retrieval.

The rest of this paper is organized as follows. We briefly review the related work in Section II. We then introduce our proposed framework DAM and two methods FastDAM and UniverDAM in Section III. The connections between the proposed two methods and the related works are discussed in Section IV. The experimental results are reported in Section V. Finally, we conclude this paper in Section VI.

II. Brief Review of Related Work

Let us represent the labeled and unlabeled instances from the target domain as $D_l^T = (\mathbf{x}_i^T, y_i^T)|_{i=1}^{n_l}$ and $D_u^T = \mathbf{x}_i^T|_{i=n_l+1}^{n_l+n_u}$, respectively, where y_i^T is the label of $\mathbf{x}_i^T$. We also define $D^s = (\mathbf{x}_i^s, y_i^s)|_{i=1}^{n_s}$ as the dataset from the sth source domain, where $s = 1, \ldots, P$ and P is the total number of source domains. Also, we assume the dimension of each instance $\mathbf{x}$ to be d. Moreover, we denote the dataset from the target domain as $D^T = D_l^T \cup D_u^T$ with the size $n_T = n_l + n_u$. In the sequel, the transpose of vector/matrix is denoted by the superscript $'$. Let us also define $\mathbf{I}_n$ as the $n \times n$ identity matrix and $\mathbf{0}_n, \mathbf{1}_n \in \Re^n$ as the column vectors of all zeros and all ones, respectively. The inequality $\mathbf{u} = [u_1, \ldots, u_n]' \geq \mathbf{v} = [v_1, \ldots, v_n]'$ means that $u_i \geq v_i$ for $i = 1, \ldots, n$. And diag $(\mathbf{u}')$ represents a diagonal matrix with u_i as its ith diagonal entry.

A. Domain Adaptation Using Prelearned Classifiers

Yang *et al.* [5] proposed the A-SVM in which a target classifier $f^T(\mathbf{x})$ is adapted from the existing auxiliary classifiers $f^s(\mathbf{x})$'s trained with the instances from the auxiliary sources. Specifically, the target decision function is formulated as

$$f^T(\mathbf{x}) = \sum_{s=1}^{P} \gamma_s f^s(\mathbf{x}) + \Delta f(\mathbf{x}) \tag{1}$$

where the perturbation function $\Delta f(\mathbf{x})$ is learned by using the labeled data D_l^T from the target domain, and $\gamma_s \in [0, 1]$ is the weight of each auxiliary classifier f^s and $\sum_{s=1}^{P} \gamma_s = 1$. Usually in the domain adaptation problems, there are only a limited number of labeled training instances from the target domain. Therefore, the adaptation process of A-SVM (i.e., learning of the target decision function) is very fast. In [5], equal weights are used for all auxiliary classifiers in the experiments. As shown in [5], the perturbation function can be formulated by $\Delta f(\mathbf{x}) = \sum_{i=1}^{n_l} \alpha_i^T y_i^T k(\mathbf{x}_i^T, \mathbf{x})$, where α_i^T is the coefficient of the ith labeled instance in the target domain, and $k(\cdot, \cdot)$ is a kernel function induced from the nonlinear feature mapping $\phi(\cdot)$. In addition, the authors assumed that the auxiliary classifiers are also learned with the same kernel function, namely, $f^s(\mathbf{x}) = \sum_{i=1}^{n_s} \alpha_i^s y_i^s k(\mathbf{x}_i^s, \mathbf{x})$, where α_i^s is the learned coefficient of the ith instance from the sth source domain. Then the decision function (1) becomes

$$f^T(\mathbf{x}) = \sum_{s=1}^{P} \gamma_s \sum_{i=1}^{n_s} \alpha_i^s y_i^s k(\mathbf{x}_i^s, \mathbf{x}) + \sum_{i=1}^{n_l} \alpha_i^T y_i^T k(\mathbf{x}_i^T, \mathbf{x}) \tag{2}$$

which is the sum of a set of weighted kernel evaluations between the test instance x and all labeled instances $\mathbf{x}_i^T$ and $\mathbf{x}_i^s$, respectively, from the target domain and all the source domains. Thus, the prediction using (2) is inefficient in large-scale applications with a large amount of test instances. In addition, it is unclear how to use the valuable unlabeled data D_u^T in the target domain in A-SVM.

Schweikert *et al.* [18] also made use of the prelearned classifiers for domain adaptation. They formally presented a so-called multiple convex combination method to linearly combine auxiliary classifiers together with the target classifier. Similar to A-SVM, each auxiliary classifier f^s is learned by using SVM with the labeled training data from one source domain. And the target classifier f^T is also obtained by simply learning a SVM classifier using the labeled training instances only from the target domain. Then the final classifier $f(\mathbf{x})$ is formulated as follows:

$$f(\mathbf{x}) = \gamma f^T(\mathbf{x}) + \frac{1-\gamma}{P} \sum_{s=1}^{P} f^s(\mathbf{x}) \tag{3}$$

where γ is the weight to balance the two terms.

B. Regularizers

Belkin *et al.* [7] extended regularized least squares (RLS) and SVM to Laplacian RLS and Laplacian SVM for SSL by adding a geometrically based regularizer, which enforces nearby points in a high-density region to have similar decision values (i.e., the so-called manifold smoothness assumption in SSL). Let us denote $G = \{X, S\}$ as an undirected weighted graph with vertex set X and similarity matrix $\mathbf{W} \in \Re^{n \times n}$, in which n is the total number of labeled and unlabeled training samples and each element w_{ij} of the real symmetric matrix $\mathbf{W}$ represents the similarity of a pair of vertices. The proposed regularizer in [7] is as follows:

$$\Omega_M(f) = \sum_{i,j=1}^{n} w_{ij} \left(f(\mathbf{x}_i) - f(\mathbf{x}_j)\right)^2 \tag{4}$$

where $f(\mathbf{x})$ is the decision function. This regularizer can be rewritten as $\Omega_M(f) = \boldsymbol{f}'\mathbf{L}\boldsymbol{f}$, where $\boldsymbol{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]'$, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian matrix and $\mathbf{D}$ is a diagonal matrix with the diagonal elements as $d_{ii} = \sum_{j=1}^{n} w_{ij}$.

A graph-based regularizer is also proposed in [28]–[30] for MTL. It is based on two MTL functions f^i and f^j of the ith task and the jth task in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$

$$\Omega_{\mathcal{G}}(f^1, f^2, \ldots, f^K) = \frac{1}{2} \sum_{i,j:\,(f^i, f^j) \in \mathcal{G}} \gamma_{ij} \|f^i - f^j\|_{\mathcal{H}}^2 \tag{5}$$

where K is the total number of tasks and γ_{ij} defines the relevance between the ith task and the jth task of a graph $\mathcal{G}$ and the graph $\mathcal{G}$ represents the weighted connectivity of tasks.

Weston *et al.* [33] proposed a new data-dependent regularizer in combination with SVM for binary-class problems, in which an additional dataset is employed in the learning process to improve the generalization ability of the learned classifier. This additional dataset, which does not belong to the positive and the negative class, is called as *Universum* as suggested by Vapnik in [34]. In [33], Weston *et al.* defined the data-dependent regularizer based on the Universum with the quadratic loss function as follows:

$$\Omega_{\mathfrak{U}}(f) = \sum_{i=1}^{n} (f(\mathbf{x}_i))^2 \tag{6}$$

where $f(\mathbf{x})$ is the decision function and n is the size of the Universum.

III. DAM Framework

In this section, we introduce our proposed framework DAM as well as the two methods FastDAM and UniverDAM for multiple source domain adaptation.

A. Smoothness Assumption for Domain Adaptation

In MR [7], the decision function in (4) is enforced to be smooth on the data manifold, namely, the two nearby instances in a high-density region should share similar decision values. For domain adaptation, we similarly assume that the target classifier $f^T(x)$ should have similar decision values on the unlabeled samples in the target domain with the precomputed base classifiers. For the ith instance $\mathbf{x}_i$ in the target domain, we denote $f_i^T = f^T(\mathbf{x}_i)$ and $f_i^s = f^s(\mathbf{x}_i)$, where f^s represents the sth base classifier. In our DAM framework, any types of classifiers can be readily used as base classifiers. In our experiments, we test our framework with two types of classifiers for f^s: 1) the standard SVM classifier learned by using the labeled instances from the sth source domain, and 2) the FR classifier trained with the labeled instances from the sth source domain and the target domain. For the unlabeled target instances D_u^T in the target domain, let us define the decision values from the target classifier and the sth base classifier as $\boldsymbol{f}_u^T = [f_{n_l+1}^T, \ldots, f_{n_T}^T]'$ and $\boldsymbol{f}_u^s = [f_{n_l+1}^s, \ldots, f_{n_T}^s]'$, respectively.

Let us also define γ_s as the weight for measuring the distribution relevance between the sth source domain and the target domain (see Section V-B for more discussions on γ_s). If the sth source domain and the target domain are relevant (i.e., γ_s is large), we enforce that f_i^s should be close to f_i^T on the unlabeled instances in the target domain. Note that the source domains are assumed to be independent of each other in this paper.

Motivated by MR [7] and graph-based multitask learning [28]–[30] (see the corresponding regularizers in Section II-B), we propose a domain-dependent regularizer for the target classifier f^T as below.

Definition 1: Domain-dependent regularizer for domain adaptation

$$\Omega_{\mathcal{A}}(f^T) = \frac{1}{2}\sum_{s=1}^{P}\gamma_s \sum_{i=n_l+1}^{n_T}\left(f_i^T - f_i^s\right)^2 = \frac{1}{2}\sum_{s=1}^{P}\gamma_s \|\boldsymbol{f}_u^T - \boldsymbol{f}_u^s\|^2. \quad (7)$$

It is worth mentioning the differences between the regularizers defined in our DAM and MTL. First, MTL simultaneously learns all task functions $f^1, \ldots, f^K$ (see (5)) and each two of the task functions are compared in the same RKHS $\mathcal{H}$. In contrast, the base classifiers f^s's in (7) are assumed to be precomputed, and DAM focuses on the learning of the target classifier only. Moreover, different kernels (or RKHS) or even different learning methods can be employed to train the base classifiers and the target classifier in DAM. Second, it is still unclear how to exploit the unlabeled samples through the regularizer (5) in MTL. In contrast, the unlabeled instances D_u^T from the target domain are used in DAM (see (7) and Fig. 1).

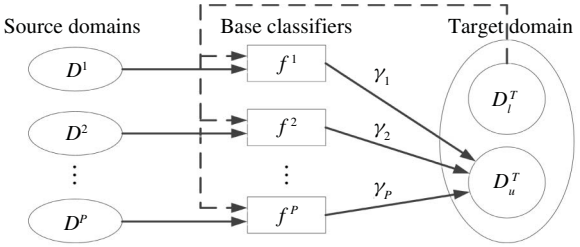

Fig. 1. Base classifiers learned by using the labeled training instances from the source domains (and the target domain as well). For each unlabeled instance $\mathbf{x}$ in D_u^T, we define its *virtual label* $\tilde{y} = \sum_{s=1}^{P}\tilde{\gamma}_s f^s(\mathbf{x})$ as a weighted summation of the decision values $f^s(\mathbf{x})$'s from the base classifiers f^s's on $\mathbf{x}$, where $\tilde{\gamma}_s = \gamma_s / \sum_{s=1}^{P}\gamma_s$.

As shown in our experiments in Section V, our proposed domain-dependent regularizer generally works well on the real-world datasets such as the TRECVID, 20 newsgroups, and email spam datasets.

B. Proposed Framework

We propose to simultaneously minimize the loss of the labeled training data from the target domain as well as different regularizers defined on the unlabeled data, such as the newly proposed domain-dependent regularizer $\Omega_{\mathcal{A}}(f^T)$ in (7) and the data-dependent regularizer $\Omega_{\mathfrak{U}}(f^T)$ in (6) for the Universum. The proposed framework DAM is then formulated as follows:

$$\min_{f^T}\ \Omega(f^T) + \lambda_L\Omega_L(f^T) + \lambda_D\Omega_D(f^T) \quad (8)$$

where $\lambda_L, \lambda_D > 0$ are tradeoff parameters, $\Omega(f^T)$ is a regularizer to control the complexity of the target classifier f^T, $\Omega_L(f^T)$ is a loss function of the target classifier f^T on the labeled instances of the target domain, and the last term $\Omega_D(f^T)$ represents different regularizers such as $\Omega_{\mathcal{A}}(f^T)$ and $\Omega_{\mathfrak{U}}(f^T)$. Note that different types of loss functions can be readily used as $\Omega_L(f^T)$ in our DAM framework for domain adaptation (e.g., the hinge loss in SVM).

In this paper, we model $\Omega_L(f^T)$ in (8) as the square error of the target classifier f^T on the labeled instances D_l^T in the target domain, which is analogous to the LS-SVM [35]. Note that the experimental results in [35] show that LS-SVM is comparable with SVM using the hinge loss. We consider two regularizers to define $\Omega_D(f^T)$. In FastDAM, we use the regularizer $\Omega_{\mathcal{A}}(f^T)$ in (7) to model $\Omega_D(f^T)$, while in UniverDAM, we additionally incorporate the regularizer $\Omega_{\mathfrak{U}}(f^T)$ in (6) by treating the instances from the source domains as Universum.

C. DAM with Fast Prediction

With the domain-dependent regularizer $\Omega_{\mathcal{A}}(f^T)$ in (7), we rewrite (8) as follows:

$$\min_{f^T}\ \Omega(f^T) + \frac{\lambda_D}{2}\sum_{s=1}^{P}\gamma_s \sum_{i=n_l+1}^{n_T}\left(f_i^T - f_i^s\right)^2 + \frac{\lambda_L}{2}\sum_{i=1}^{n_l}\left(f_i^T - y_i^T\right)^2. \quad (9)$$

1) Non-Sparse Solution:

Theorem 1: Assume that the target decision function is in the form of $f^T(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x})$ and the regularizer $\Omega(f^T) = \|\mathbf{w}\|^2/2$. Then, the solution f^T of the optimization problem (9) is

$$f^T(\mathbf{x}) = \lambda_D \sum_{s=1}^{P} \gamma_s \sum_{i=n_l+1}^{n_T} f^s(\mathbf{x}_i^T)\tilde{k}(\mathbf{x}_i^T, \mathbf{x}) + \sum_{i=1}^{n_l} \alpha_i^T \tilde{k}(\mathbf{x}_i^T, \mathbf{x}) \tag{10}$$

where

$$\tilde{k}(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) - \mathbf{k}'_{\mathbf{x}_i}(\mathbf{I}_{n_u} + \mathbf{M}\mathbf{K}_u)^{-1}\mathbf{M}\mathbf{k}_{\mathbf{x}_j} \tag{11}$$

is the kernel function for domain adaptation, $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)'\phi(\mathbf{x}_j)$ is the inner product between $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$, $\mathbf{k}_{\mathbf{x}} = [k(\mathbf{x}_{n_l+1}^T, \mathbf{x}), \ldots, k(\mathbf{x}_{n_T}^T, \mathbf{x})]'$, $\mathbf{K}_u = [k(\mathbf{x}_i^T, \mathbf{x}_j^T)] \in \Re^{n_u \times n_u}$ is the kernel matrix defined on the unlabeled data from the target domain, $\mathbf{M} = \lambda_D \sum_{s=1}^{P} \gamma_s \mathbf{I}_{n_u}$, and α_i^T is the coefficient for the ith labeled instances in the target domain.

Proof: The proof can be easily derived by solving the dual problem of (9). ∎

Note that similar to [28], [30], the solution of the target decision function f^T in (10) is non-sparse. All the base classifiers f^s's need to be used for predicting the labels of the target instances, making it inefficient for large-scale applications (e.g., video concept detection). Moreover, similar to the manifold kernel defined in [36], the kernel for domain adaptation in (11) involves the matrix inversion of a matrix $\mathbf{I}_{n_u} + \mathbf{M}\mathbf{K}_u$, which is computationally infeasible when n_u is large.

2) Sparse Solution: As shown in [37] and [38], the use of the ϵ-insensitive loss function in SVR can usually lead to a sparse representation of the decision function.[2] To obtain the sparse solution, we therefore introduce an additional term in (9), which regulates the approximation quality and the sparsity of the decision function. Moreover, we also assume that the regularizer $\Omega(f^T) = \|\mathbf{w}\|^2/2$ for the penalty of function complexity of f^T. The optimization problem (9) is then rewritten as

$$\min_{f_i^T, \mathbf{w}, b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n_T} \ell_\epsilon(\mathbf{w}'\phi(\mathbf{x}_i) + b - f_i^T) + \frac{\lambda_L}{2}\|\boldsymbol{f}_l^T - \mathbf{y}_l^T\|^2 + \frac{\lambda_D}{2}\sum_{s=1}^{P}\gamma_s\|\boldsymbol{f}_u^T - \boldsymbol{f}_u^s\|^2 \tag{12}$$

where C is another tradeoff parameter, $\boldsymbol{f}_l^T = [f_1^T, \ldots, f_{n_l}^T]'$ is the vector of the target decision function on the labeled instances D_l^T from the target domain, $\mathbf{y}_l^T = [y_1^T, \ldots, y_{n_l}^T]'$ is the label vector of the labeled training instances in the target domain, and $\ell_\epsilon(t)$ is ϵ-insensitive loss $\ell_\epsilon(t) = \begin{cases} |t| - \epsilon, & \text{if } |t| > \epsilon; \\ 0, & \text{otherwise.} \end{cases}$ Since ϵ-insensitive loss is nonsmooth, (12) is usually transformed as a constrained optimization problem, that is

$$\min_{f_i^T, \mathbf{w}, b, \xi_i, \xi_i^*} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n_T}(\xi_i + \xi_i^*) + \frac{\lambda_L}{2}\|\boldsymbol{f}_l^T - \mathbf{y}_l^T\|^2 + \frac{\lambda_D}{2}\sum_{s=1}^{P}\gamma_s\|\boldsymbol{f}_u^T - \boldsymbol{f}_u^s\|^2 \tag{13}$$

$$\text{s.t.} \quad \mathbf{w}'\phi(\mathbf{x}_i^T) + b - f_i^T \le \epsilon + \xi_i, \quad \xi_i \ge 0 \tag{14}$$

$$f_i^T - \mathbf{w}'\phi(\mathbf{x}_i^T) - b \le \epsilon + \xi_i^*, \quad \xi_i^* \ge 0 \tag{15}$$

where ξ_i's and ξ_i^*'s are slack variables for ϵ-insensitive loss. In the experiment, we fix $\epsilon = 0.1$, which is also the default value in the online toolbox LIBSVM [39].

Detailed Derivation: Let us represent $\boldsymbol{f}^T = [\boldsymbol{f}_l^{T'}, \boldsymbol{f}_u^{T'}]'$. By introducing the Lagrange multipliers α_i's and η_i's (resp. α_i^*'s and η_i^*'s) for the constraints in (14) (resp. (15)), we set the derivatives of the Lagrangian of (13) w.r.t. the primal variables ($\boldsymbol{f}^T$, $\mathbf{w}$, b, ξ_i, and ξ_i^*) to zeros, respectively, and we obtain

$$\boldsymbol{f}^T = \tilde{\mathbf{y}} + \text{diag}\left(\left[\frac{1}{\lambda_L}\mathbf{1}'_{n_l}, \frac{1}{p\lambda_D}\mathbf{1}'_{n_u}\right]\right)(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \tag{16}$$

and $\mathbf{w} = \boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \boldsymbol{\alpha})$, $\mathbf{1}'_{n_T}\boldsymbol{\alpha} = \mathbf{1}'_{n_T}\boldsymbol{\alpha}^*$, $\mathbf{0}_{n_T} \le \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \le C\mathbf{1}_{n_T}$, where $\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y}_l^T \\ \sum_{s=1}^{P}\tilde{\gamma}_s \boldsymbol{f}_u^s \end{bmatrix}$, $\tilde{\gamma}_s = \gamma_s / \sum_{s=1}^{P}\gamma_s$ is the normalized weight for the s-th base classifier, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{n_T}]'$ and $\boldsymbol{\alpha}^* = [\alpha_1^*, \ldots, \alpha_{n_T}^*]'$ are the vectors of the dual variables, and $\boldsymbol{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_{n_T})]$. Substituting them back into the Lagrangian, we arrive at the following dual formulation:

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)'\tilde{\mathbf{K}}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \tilde{\mathbf{y}}'(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \epsilon\mathbf{1}'_{n_T}(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*)$$
$$\text{s.t.} \quad \mathbf{1}'_{n_T}\boldsymbol{\alpha} = \mathbf{1}'_{n_T}\boldsymbol{\alpha}^*, \quad \mathbf{0}_{n_T} \le \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \le C\mathbf{1}_{n_T} \tag{17}$$

where $\tilde{\mathbf{K}} = \mathbf{K} + \text{diag}([(1/\lambda_L)\mathbf{1}'_{n_l}, (\frac{1}{p\lambda_D})\mathbf{1}'_{n_u}])$ is a transformed kernel matrix, $\mathbf{K} = \boldsymbol{\Phi}'\boldsymbol{\Phi}$ and $p = \sum_{s=1}^{P}\gamma_s$.

Parametric prediction: From the Karush–Kuhn–Tucker (KKT) condition in (16), we can obtain the vector of the target decision values $\boldsymbol{f}^T$. Moreover, the decision value of any unlabeled data D_u^T in the target domain is given as $f^T(\mathbf{x}_i) = \sum_{s=1}^{P}\tilde{\gamma}_s f^s(\mathbf{x}_i) + (\alpha_i - \alpha_i^*)/\lambda_D$, $\forall i = n_l + 1, \ldots, n_T$, which is similar to that of A-SVM when we set the perturbation function Δf in A-SVM for the unlabeled instance $\mathbf{x}_i$ as $\Delta f(x_i) = (\alpha_i - \alpha_i^*)/\lambda_D$. However, $f^T(\mathbf{x}_i)$ also involves the ensemble outputs from the base classifiers. Alternatively, we use the parametric form of the target decision function for label prediction on any test instance $\mathbf{x}$ by

$$f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b = \sum_{i:\,\alpha_i - \alpha_i^* \ne 0}(\alpha_i - \alpha_i^*)k(\mathbf{x}_i^T, \mathbf{x}) + b \tag{18}$$

which is a linear combination of $k(\mathbf{x}_i, \mathbf{x})$'s only without involving any base classifiers. According to the KKT conditions, if a target instance $\mathbf{x}_i^T$ has the value $\left|\mathbf{w}'\phi(\mathbf{x}_i^T) + b - f_i^T\right|$ less than ϵ, then their corresponding coefficient $(\alpha_i - \alpha_i^*)$ in (18) becomes zero. Therefore, with the use of ϵ-insensitive loss function, the computation for the prediction using the sparse representation in (18) can be greatly reduced when compared with A-SVM. We therefore refer to this sparse-solution version of DAM as FastDAM.

[2]For the samples that fall within the ϵ-tube of the ϵ-insensitive loss, the corresponding dual variables α_i's will be zeros, which makes the solution sparse. For more details, refer to [37].

D. DAM with the Universum

Since the classification hyperplane learned with only a limited number of labeled training instances in the target domain may overfit these data, the generalization ability of the learned FastDAM classifier may be degraded. As shown in the traditional transductive and SSL methods [10], [12], [36], [40], unlabeled data can be employed to improve the classification performance. However, these algorithms require that the unlabeled data come from the same distribution of the labeled training data. Recently, Vapnik [34] proposed the use of an additional unlabeled dataset called *Universum* to enhance the generalization ability of the learned classifier for the binary classification tasks. In contrast to the traditional transductive and SSL methods, it does not assume that the unlabeled data (i.e., Universum) should come from the same distribution of the training data. Weston *et al.* [33] introduced a new regularizer into the SVM framework based on the Universum. The proposed method, referred to as $\mathfrak{U}$-SVM, maximizes not only the margin between two classes but also the number of contradictions on the Universum. For a detailed definition of contradiction on the Universum, refer to [33], [34]. Sinz *et al.* [32] discovered the connection between the Universum-related algorithm in [33] and other algorithms including SVM in a projected subspace, Fisher discriminant analysis, and oriented principal components analysis.

Recall that the learned classifier by using labeled instances from the source and target domains may not perform well due to the distribution mismatch [4], [5], [9], [41], [42]. As mentioned in [33] and [34], Universum can be drawn from a distribution that is different from, yet close to, that of the labeled training samples. In this paper, we use the instances in the source domains as the Universum because of the following two aspects: 1) the distributions of the samples from the source domains and the target domain are different but overlap to some extent, and 2) the Universum regularizer $\Omega_{\mathfrak{U}}(f^T)$ in (6) is a data-dependent regularizer and it can be used to control the complexity of the learned classifier. To this end, we model $\Omega_D(f^T)$ by using two regularizers $\Omega_{\mathcal{A}}(f^T)$ and $\Omega_{\mathfrak{U}}(f^T)$, namely

$$\Omega_D(f^T) = \frac{1}{2}\sum_{s=1}^{P}\gamma_s \sum_{i=n_l+1}^{n_T}\left(f^T(\mathbf{x}_i^T) - f^s(\mathbf{x}_i^T)\right)^2 + \frac{\theta}{2}\sum_{s=1}^{P}\sum_{i=1}^{n_s}\left(f^T(\mathbf{x}_i^s)\right)^2 \tag{19}$$

where $\theta > 0$ is a tradeoff parameter.

Denote $\lambda_{D_1} = \lambda_D$ and $\lambda_{D_2} = \theta\lambda_D$. Substituting (19) back into the DAM framework (8), we arrive at the following optimization problem:

$$\begin{aligned}\min_{f^T}\ \ & \Omega(f^T) + \frac{\lambda_L}{2}\sum_{i=1}^{n_l}\left(f^T(\mathbf{x}_i^T) - y_i^T\right)^2 \\ & + \frac{\lambda_{D_1}}{2}\sum_{s=1}^{P}\gamma_s\sum_{i=n_l+1}^{n_T}\left(f^T(\mathbf{x}_i^T) - f^s(\mathbf{x}_i^T)\right)^2 \\ & + \frac{\lambda_{D_2}}{2}\sum_{s=1}^{P}\sum_{i=1}^{n_s}\left(f^T(\mathbf{x}_i^s)\right)^2.\end{aligned} \tag{20}$$

1) Detailed Solution: Note that the optimization problem (20) can be solved through the least-squares method. However, it is computationally infeasible as stated in Section III-C.1. Again, we employ ϵ-insensitive loss to regulate the approximation quality and the sparsity of the target decision function. Let $\boldsymbol{f}_T^T = [f^T(\mathbf{x}_1^T), \ldots, f(\mathbf{x}_{n_T}^T)]' = [\boldsymbol{f}_l^{T'}, \boldsymbol{f}_u^{T'}]'$, $\boldsymbol{\xi}_T = [\xi_{T,1}, \ldots, \xi_{T,n_T}]'$, and $\boldsymbol{\xi}_T^* = [\xi_{T,1}^*, \ldots, \xi_{T,n_T}^*]'$ for the target domain, $\boldsymbol{f}_s^T = [f^T(\mathbf{x}_1^s), \ldots, f^T(\mathbf{x}_{n_s}^s)]'$, $\boldsymbol{\xi}_s = [\xi_{s,1}, \ldots, \xi_{s,n_s}]'$ and $\boldsymbol{\xi}_s^* = [\xi_{s,1}^*, \ldots, \xi_{s,n_s}^*]'$ for the sth source domain. With the ϵ-insensitive loss, we can rewrite (20) as the following constrained optimization problem which is referred to as UniverDAM:

$$\begin{aligned}\min_{\substack{f^T,\mathbf{w},b,\xi_{T,i}\\ \xi_{T,i}^*,\xi_{s,i},\xi_{s,i}^*}}\ & \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\mathbf{1}_{n_T}'(\boldsymbol{\xi}_T + \boldsymbol{\xi}_T^*) + \sum_{s=1}^{P}\mathbf{1}_{n_s}'(\boldsymbol{\xi}_s + \boldsymbol{\xi}_s^*)\right) \\ & + \frac{\lambda_L}{2}\|\boldsymbol{f}_l^T - \mathbf{y}_l^T\|^2 + \frac{\lambda_{D_1}}{2}\sum_{s=1}^{P}\gamma_s\|\boldsymbol{f}_u^T - \boldsymbol{f}_u^s\|^2 \\ & + \frac{\lambda_{D_2}}{2}\sum_{s=1}^{P}\|\boldsymbol{f}_s^T\|^2\end{aligned} \tag{21}$$

s.t. For the target domain, $i = 1, \ldots, n_T$

$$\mathbf{w}'\phi(\mathbf{x}_i^T) + b - f^T(\mathbf{x}_i^T) \le \epsilon_T + \xi_{T,i},\ \ \xi_{T,i} \ge 0 \tag{22}$$

$$f^T(\mathbf{x}_i^T) - \mathbf{w}'\phi(\mathbf{x}_i^T) - b \le \epsilon_T + \xi_{T,i}^*,\ \ \xi_{T,i}^* \ge 0 \tag{23}$$

For the sth source domain, $i = 1, \ldots, n_s$:

$$\mathbf{w}'\phi(\mathbf{x}_i^s) + b - f^T(\mathbf{x}_i^s) \le \epsilon_s + \xi_{s,i},\ \ \xi_{s,i} \ge 0 \tag{24}$$

$$f^T(\mathbf{x}_i^s) - \mathbf{w}'\phi(\mathbf{x}_i^s) - b \le \epsilon_s + \xi_{s,i}^*,\ \ \xi_{s,i}^* \ge 0. \tag{25}$$

Let us define $\boldsymbol{f}^T = [\boldsymbol{f}_T^{T'}, \boldsymbol{f}_1^{T'}, \ldots, \boldsymbol{f}_P^{T'}]'$, $\boldsymbol{\xi} = [\boldsymbol{\xi}_T', \boldsymbol{\xi}_1', \ldots, \boldsymbol{\xi}_P']'$, $\boldsymbol{\xi}^* = [\boldsymbol{\xi}_T^{*\prime}, \boldsymbol{\xi}_1^{*\prime}, \ldots, \boldsymbol{\xi}_P^{*\prime}]'$, $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_T', \boldsymbol{\alpha}_1', \ldots, \boldsymbol{\alpha}_P']'$, $\boldsymbol{\alpha}^* = [\boldsymbol{\alpha}_T^{*\prime}, \boldsymbol{\alpha}_1^{*\prime}, \ldots, \boldsymbol{\alpha}_P^{*\prime}]'$ and $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_T, \boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_P]$. We introduce the Lagrangian multipliers $\boldsymbol{\alpha}_T = [\alpha_{T,1}, \ldots, \alpha_{T,n_T}]'$ and $\boldsymbol{\eta}_T = [\eta_{T,1}, \ldots, \eta_{T,n_T}]'$ (resp. $\boldsymbol{\alpha}_T^* = [\alpha_{T,1}^*, \ldots, \alpha_{T,n_T}^*]'$ and $\boldsymbol{\eta}_T^* = [\eta_{T,1}^*, \ldots, \eta_{T,n_T}^*]'$) for the constraints of the target domain in (22) (resp. [23]), as well as the Lagrangian multipliers $\boldsymbol{\alpha}_s = [\alpha_{s,1}, \ldots, \alpha_{s,n_s}]'$ and $\boldsymbol{\eta}_s = [\eta_{s,1}, \ldots, \eta_{s,n_s}]'$ (resp. $\boldsymbol{\alpha}_s^* = [\alpha_{s,1}^*, \ldots, \alpha_{s,n_s}^*]'$ and $\boldsymbol{\eta}_s^* = [\eta_{s,1}^*, \ldots, \eta_{s,n_s}^*]'$) for the constraints of the sth source domain in (24) (resp. (25)). By setting the derivatives of the Lagrangian of (21) w.r.t. the primal variables $\boldsymbol{f}^T, \mathbf{w}, b, \boldsymbol{\xi}$ and $\boldsymbol{\xi}^*$ to zero, we obtain the following solutions:

$$\boldsymbol{f}^T = \tilde{\mathbf{y}} + \operatorname{diag}\left(\left[\frac{1}{\lambda_L}\mathbf{1}_{n_l}', \frac{1}{p\lambda_{D_1}}\mathbf{1}_{n_u}', \frac{1}{\lambda_{D_2}}\mathbf{1}_{\sum_{s=1}^{P}n_s}'\right]\right)(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \tag{26}$$

$\mathbf{w} = \boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \boldsymbol{\alpha})$, $\mathbf{1}_N'\boldsymbol{\alpha} = \mathbf{1}_N'\boldsymbol{\alpha}^*$ and $\mathbf{0}_N \le \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \le C\mathbf{1}_N$, where $\tilde{\mathbf{y}} = \left[\mathbf{y}_l^{T\prime}, \sum_{s=1}^{P}\tilde{\gamma}_s\boldsymbol{f}_u^{s\prime}, \mathbf{0}_{\sum_{s=1}^{P}n_s}'\right]'$, $N = n_T + \sum_{s=1}^{P}n_s$ is the total number of training instances and $\tilde{\gamma}_s = \gamma_s/\sum_{s=1}^{P}\gamma_s$ is the normalized weight for the sth base classifier. Similar to the derivation of FastDAM in Section III-C.2, we arrive at the dual problem of UniverDAM as follows:

$$\begin{aligned}\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*}\ & \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)'\tilde{\mathbf{K}}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \tilde{\mathbf{y}}'(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \boldsymbol{\epsilon}'(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) \\ \text{s.t.}\ & \mathbf{1}_N'\boldsymbol{\alpha} = \mathbf{1}_N'\boldsymbol{\alpha}^*,\quad \mathbf{0}_N \le \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \le C\mathbf{1}_N\end{aligned} \tag{27}$$

where $\tilde{\mathbf{K}} = \mathbf{K} + \text{diag}([(1/\lambda_L)\mathbf{1}'_{n_l}, (\frac{1}{p\lambda_{D_1}})\mathbf{1}'_{n_u}, (1/\lambda_{D_2})\mathbf{1}'_{\sum_{s=1}^{P} n_s}])$ is a transformed kernel matrix, $\mathbf{K} = \mathbf{\Phi}'\mathbf{\Phi}$, $p = \sum_{s=1}^{P} \gamma_s$, and $\boldsymbol{\epsilon} = [\epsilon_T \mathbf{1}'_{n_T}, \epsilon_1 \mathbf{1}'_{n_1}, \ldots, \epsilon_P \mathbf{1}'_{n_P}]'$. In the experiments, we fix all ϵ_s and ϵ_T as 0.1.

2) Parametric Prediction: The solution (26) to the target decision function f^T is transductive. Namely, it is restricted to the training data only and it cannot be used for the prediction of newly coming test instances. To this end, similarly as in Section III-C2, we introduce a parametric form of the target decision function f^T for label prediction. With the dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ obtained from (27), the target decision function f^T on any test instance $\mathbf{x}$ is formulated as $f^T(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b = \sum_{i:\, \alpha_{T,i} - \alpha^*_{T,i} \neq 0} (\alpha_{T,i} - \alpha^*_{T,i}) k(\mathbf{x}_i^T, \mathbf{x}) + \sum_{s=1}^{P} \sum_{i:\, \alpha_{s,i} - \alpha^*_{s,i} \neq 0} (\alpha_{s,i} - \alpha^*_{s,i}) k(\mathbf{x}_i^s, \mathbf{x}) + b$.

IV. Discussion

A. Connection Between FastDAM and UniverDAM

We first discuss the connection between FastDAM and UniverDAM and introduce a theorem on the degradation of UniverDAM into FastDAM. One can observe that, when $\alpha_{s,i} = \alpha^*_{s,i} = 0$, the dual optimization problem (27) of UniverDAM reduces to the dual optimization problem (17) of FastDAM. The following theorem shows that UniverDAM can be reduced to FastDAM under some conditions.

Theorem 2: The optimization problem (21) of UniverDAM will be reduced to the optimization problem (13) of FastDAM, if each ϵ_s $(s = 1, \ldots, P)$ satisfies the following condition:

$$\epsilon_s > \max_{i=1,\ldots,n_s} \left| \mathbf{w}'\phi(\mathbf{x}_i^s) + b - f^T(\mathbf{x}_i^s) \right|. \tag{28}$$

Proof: The proof is given in the Appendix. ■

Recall that Universum is initially proposed for the binary-class problem [32]–[34], which means that the positive instances are from one class and the negative instances are from another class. The Universum is an additional data collection belonging to neither class, which is used as prior knowledge to help find the optimal margin and enhance the generalization ability of the learned classifier. However, it is not clear how to make use of the Universum in the multiclass setting in which the training data are from multiple classes.

B. Connection to SVR

Under the framework of DAM, we have introduced two methods, namely, FastDAM and UniverDAM. Surprisingly, for both FastDAM and UniverDAM, the dual forms of (17) and (27) do not involve any expensive matrix operation as in [28]–[30] and can be reduced to a form which is quite close to the dual of ϵ-SVR

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \ & \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)'\mathbf{K}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \mathbf{y}'(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \epsilon \mathbf{1}'(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) \\ \text{s.t.} \ & \boldsymbol{\alpha}'\mathbf{1} = \boldsymbol{\alpha}^{*\prime}\mathbf{1}, \ \ \mathbf{0} \leq \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \leq C\mathbf{1} \end{aligned} \tag{29}$$

except for the kernel matrix $\mathbf{K}$, the regression label vector $\mathbf{y}$, and the parameter vector $\epsilon\mathbf{1}$. For the ease of presentation, we take the dual (17) of FastDAM as an example, and note that the dual (27) of UniverDAM can be similarly analyzed. To transform ϵ-SVR into FastDAM, the kernel matrix $\mathbf{K}$ and the regression label vector $\mathbf{y}$ in (29) are replaced by the transformed kernel matrix $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{y}}$ in (17), respectively. In experiments, we normalize the sum of γ_s to 1 (i.e., $p = 1$). So the transformed kernel matrix is $\mathbf{K} + \text{diag}([(1/\lambda_L)\mathbf{1}'_{n_l}, (1/\lambda_D)\mathbf{1}'_{n_u}])$, which is similar to the automatic relevance determination kernel used in a Gaussian process, where λ_L and λ_D are the parameters to control the noise of output. Moreover, the ith item of the last n_u entries of $\tilde{\mathbf{y}}$ is $\tilde{y}_i = \sum_{s=1}^{P} \tilde{\gamma}_s f^s(\mathbf{x}_i)$, which can be explained as the virtual label generated by a weighted summation of the decision values $f^s(\mathbf{x}_i)$'s from the base classifiers f^s's on the unlabeled instance $\mathbf{x}_i$ in D_u^T (also see Fig. 1). Moreover, the objective function of FastDAM in (12) can be solved efficiently by using state-of-the-art SVM solvers such as LIBSVM [39]. When compared with the original formulation in (9), the calculation of the matrix inversion in (11) is avoided.

C. Discussion on Related Work

Our proposed DAM framework is different from MTL [28]–[30], [43]. DAM focuses on learning the target decision classifier only by leveraging the existing base classifiers, and the computational cost in the learning stage is reduced, especially for FastDAM. In addition, according to the definition of our domain-dependent regularizer in (7), the base classifiers can be trained with different kernels and even different learning methods.

The most related work to DAM is A-SVM [5], in which the new SVM classifier is adapted from the existing auxiliary classifiers. However, DAM is different from A-SVM in two aspects. First, A-SVM did not exploit the unlabeled data D_u^T in the target domain. In contrast, the unlabeled instances D_u^T in the target domain are employed in DAM (see the domain-dependent regularizer defined in (7)). Second, A-SVM employed auxiliary classifiers for the label prediction of the instances in the target domain. In contrast, the target classifier learned in DAM (see (18)) is in a sparse representation of the target instances only. In Table I, we summarize the comparisons between our two methods (i.e., FastDAM and UniverDAM) and other domain adaptation methods (i.e., FR [3], A-SVM [5], multiple convex combination of SVM (MCC-SVM) [18], and multiple KMM (Multi-KMM) [18]) which will be evaluated in the experiments.

Finally, DAM also differs from other SSL methods. SSL methods generally assumed that the labeled and unlabeled samples come from the same domain. In contrast, DAM does not enforce such a constraint.

V. Experiments

In the experiments, we evaluate our two methods FastDAM and UniverDAM for two multiple domain adaptation related applications: 1) video concept detection, and 2) document retrieval.

A. Descriptions of Datasets

We conduct experiments on three datasets which are suitable for the multiple source domain adaptation applications.

TABLE I
SUMMARY OF THE COMPARISONS BETWEEN OUR TWO METHODS (i.e., FASTDAM AND UNIVERDAM) AND OTHER DOMAIN ADAPTATION METHODS

	Smoothness assumption	Prelearned classifiers	Source data	Labeled target data	Unlabeled target data	Fast adaptation	Fast prediction
FR [3]	×	×	✓	✓	×	×	×
A-SVM [5]	×	✓	×	✓	×	✓	×
MCC-SVM [18]	×	✓	×	✓	×	✓	×
Multi-KMM [18]	×	×	✓	Optional	✓	×	×
FastDAM	✓	✓	×	Optional	✓	✓	✓
UniverDAM	✓	✓	✓	Optional	✓	×	×

TABLE II
DESCRIPTION OF THE TRECVID 2005 DATASET

Domain	Source domains					Target domain
Channel	CNN_ENG	MSNBC_ENG	NBC_ENG	CCTV4_CHN	NTDTV_CHN	LBC_ARB
# keyframes	11 025	8905	9322	10 896	6481	15 272

We use the challenging TRECVID 2005 dataset for the video concept detection task and employ two text datasets (i.e., 20 newsgroups and email spam) for document retrieval.

1) TRECVID 2005 Dataset: The TRECVID[3] video corpus is one of the largest annotated video benchmark datasets for research purposes. The TRECVID 2005 dataset contains 61 901 keyframes extracted from 108 hours of video programs from six different broadcast channels, including three English channels (CNN, MSNBC, and NBC), two Chinese channels (CCTV and NTDTV), and one Arabic channel (LBC). The total number of keyframes in each channel is listed in Table II. Thirty-six semantic concepts are chosen from the large-scale concept ontology for multimedia-lite lexicon [44], which cover the dominant visual concepts present in broadcast news videos including objects, locations, people, events, and programs. And these concepts have been manually annotated to describe the visual content of the keyframes in the TRECVID 2005 dataset.

As shown in [5], the data distributions of six channels are quite different, making it suitable for evaluating domain adaptation methods. In this paper, three English channels and two Chinese channels are used as the source domains, and the Arabic channel is used as the target domain D^T. The training dataset comprises all the labeled samples from the source domains as well as the labeled samples (i.e., D_l^T) from the target domain, in which 10 samples per concept are randomly chosen. The remaining samples in the target domain are used as the test dataset. Moreover, from the test dataset we randomly select 4000 instances as the unlabeled training data. We only sample the instances from the target domain once because of the very high computational cost in the large TRECIVD 2005 dataset.

Three low-level global features, namely, grid color moment (225 dim.), Gabor texture (48 dim.), and edge direction histogram (73 dim.), are used to represent the diverse content of keyframes, because of their consistent and good performance reported in TRECVID [5], [45]. Yanagawa *et al.* [46] have made the three types of features extracted from the TRECVID 2005 dataset publicly available. The three types of features are then put together to form a 346-D feature to represent each keyframe.

2) Twenty Newsgroups Dataset: The 20 newsgroups dataset[4] contains 18 774 documents, and has a hierarchical structure with 6 main categories and 20 subcategories. We choose the instances from three main categories with at least four subcategories and generate three settings for evaluating multiple source domain adaptation algorithms. For each setting, we consider one main category as the positive class and use another one as the negative class, and employ all the labeled instances from two subcategories (i.e., one from the positive class and the other from the negative class) to construct one domain. In the experiments, we have three source domains and one target domain (see Table III for the detailed settings). The training dataset comprises all the labeled samples from the source domains as well as $2m$ labeled samples from the target domain, in which m positive and m negative instances are randomly chosen. The remaining samples in the target domain are used as the unlabeled training data and the test data. In the experiments, we set m as $0, 2, 4, 6, 10, 15$, and 20. We repeat the experiments 10 times with different randomly sampled instances from the target domain and report the means and the standard deviations. The word-frequency feature is used to represent each document.

3) Email Spam Dataset: The email spam dataset[5] contains a set of 4000 publicly available labeled emails as well as three email sets (each has 2500 emails) annotated by three different users. Therefore, the data distributions of the three user-annotated email sets and the publicly available email set differ from each other. For each of the four datasets, one-half of the emails are *non-spam* (labeled as 1) and the other half of them are *spam* (labeled as -1). In our experiments, we consider the three user-annotated sets as three source domains, and employ the publicly available email set as the target

[3]Available at: http://www-nlpir.nist.gov/projects/trecvid.

[4]Available at: http://people.csail.mit.edu/jrennie/20Newsgroups/.

[5]Available at: http://www.ecmlpkdd2006.org/challenge.html.

TABLE III
DESCRIPTION OF THE 20 NEWSGROUPS DATASET

	Source domains	Target domain
rec versus sci	rec.autos & sci.crypt rec.motorcycles & sci.electronics rec.sport.baseball & sci.med	rec.sport.hockey & sci.space
comp versus rec	comp.graphics & rec.autos comp.os.ms-windows.misc & rec.motorcycles comp.sys.ibm.pc.hardware & rec.sport.baseball	comp.sys.mac.hardware & rec.sport.hockey
sci versus comp	sci.crypt & comp.graphics sci.electronics & comp.os.ms-windows.misc sci.med & comp.sys.ibm.pc.hardware	sci.space & comp.sys.mac.hardware

TABLE IV
DESCRIPTION OF THE EMAIL SPAM DATASET

Domain	Source domains			Target domain
Email sets	User1 (U00)	User2 (U01)	User3 (U02)	Public set
# emails	2500	2500	2500	4000

domain (see Table IV for more details). The training dataset comprises all the labeled samples from the source domains as well as 20 labeled samples from the target domain, in which 10 positive and 10 negative instances are randomly chosen. The remaining samples in the target domain are used as the unlabeled training data and also as the test data. We repeat the experiments 10 times with different randomly sampled instances from the target domain and report the means and the standard deviations. Again, we use the word-frequency feature to represent each document.

B. Experimental Setup

For performance evaluation, we use noninterpolated average precision (AP) [47]. Any base classifiers can be readily used in our DAM framework. In the experiments, we test our methods using two types of base classifiers: 1) the standard SVM classifier learned by using the labeled instances from one source domain, which is the same as the auxiliary classifier in A-SVM [5], and 2) the FR classifier trained with the labeled instances from one source domain and the target domain. On the TRECVID 2005 and email spam datasets, we use FR classifiers as base classifiers in FastDAM and UniverDAM. And on the 20 newsgroups dataset, we use SVM classifiers learned from source domains as base classifiers.

Recall that FastDAM and UniverDAM both make use of the virtual labels $\tilde{y}$ (i.e., the weighted decision values from the base classifiers) of the unlabeled instances from the target domain (see Section IV-B for more details). However, the unlabeled instances from the target domain may bring ambiguity in the learning of the target classifier if their virtual labels are close to zero. To alleviate such side effect of the ambiguity as well as accelerate the learning process, we discard the unlabeled instances from the target domain with the virtual labels ranging from -0.3 to 0.3 and only employ the remaining unlabeled instances. In the experiment, we empirically fix the thresholds as -0.3 and 0.3. We will investigate how to automatically determine the thresholds in the future.

1) Detailed Setup for Video Concept Detection: We compare our method FastDAM with the baseline SVM and other four domain adaptation methods: MCC-SVM [18], FR [3], A-SVM [5], and Multi-KMM [18]. We do not test UniverDAM for the video concept detection task because the instances from each source domain come from multiple classes, which violates the basic assumption on Universum stated in Section V. It is also worth mentioning that UniverDAM can achieve comparable results with FastDAM because FastDAM is a special case of UniverDAM (see Section IV-A for the details about the connection between FastDAM and UniverDAM).

In this paper, we focus on the multiple source domain setting. For the baseline SVM algorithm, we report the results for two cases: 1) in SVM_T, we only use the training instances from the target domain (i.e., D_l^T) for SVM learning, and 2) in SVM_S, we *equally fuse*[6] the decision values of five base classifiers independently trained with the labeled instances from five source domains. MCC-SVM, FR, A-SVM, and Multi-KMM can cope with the training samples from multiple source domains. For MCC-SVM, as in [18], we equally fuse the decision values of six SVM classifiers independently trained with the labeled instances from the target domain and five source domains. And similarly for FR, we also equally fuse the decision values of five base classifiers with each classifier learned using the labeled instances from one source domain and the target domain. Considering we only have a limited number of labeled training samples from the target domain (i.e., 10 samples per class), for Multi-KMM [18] we shift the samples from each source domain toward the mean of the target samples without considering the class label information. We also empirically set the parameter α in Multi-KMM as 1. The Multi-KMM classifier is finally learned by using the shifted samples from the source domains and the labeled data from the target domain. Considering that Multi-KMM and FastDAM can take advantage of both labeled and unlabeled data, we use a semisupervised setting in this paper. In practice, 4000 test instances from the target domain are randomly sampled as D_u^T for Multi-KMM and Fast-

[6]For each of the P source domains, we train one SVM by using the corresponding labeled samples in the source domain. Then, for each test instance $\mathbf{x}$, the decision values from the P SVM classifiers are converted into the probability values by using the sigmoid function (i.e., $g(t) = 1/(1+\exp(-t))$ as suggested in [48]. Finally, we average the P probability values as the final prediction of the test instance $\mathbf{x}$.

TABLE V
MAPS (%) OF ALL METHODS OVER 36 CONCEPTS ON THE TRECVID 2005 DATASET

	SVM_T	SVM_S	MCC-SVM	FR	Multi-KMM	A-SVM	FastDAM_50	FastDAM_200
MAP	25.5	26.4	28.8	30.4	26.7	28.1	**32.2**	**32.6**

DAM, which are used as unlabeled data during the learning process.

For all methods, we train one-versus-others SVM classifiers with the fixed regularization parameter $C = 1$. For FastDAM, we fix the tradeoff parameters $\lambda_L = \lambda_D = 100$. Gaussian kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$) is used as the default kernel in SVM_T, SVM_S, MCC-SVM, FR, Multi-KMM, and FastDAM, where γ is set to $1/d = 0.0029$ ($d = 346$ is the feature dimension). For A-SVM, we train 50 auxiliary classifiers by independently using 5 sources and 10 kernel parameters for the Gaussian kernel, which are set as $1.2^{\delta}\gamma$, where $\delta \in \{-0.5, 0, 0.5, \ldots, 4\}$. We also report two results for FastDAM: 1) in FastDAM_50, we exploit 50 base classifiers independently learned by using the labeled training data from each source domain and the target domain with the Gaussian kernel and the same ten kernel parameters, and 2) in FastDAM_200, we additionally employ another three types of kernels: Laplacian kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sqrt{\gamma}\|\mathbf{x}_i - \mathbf{x}_j\|)$), inverse square distance kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = 1/(\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2 + 1)$), and inverse distance kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = 1/(\sqrt{\gamma}\|\mathbf{x}_i - \mathbf{x}_j\| + 1)$). Then, for FastDAM, there are in total 200 base classifiers from 5 sources, 4 types of kernels, and 10 kernel parameters.

In A-SVM and FastDAM, we also need to determine the weight γ_s for the s-th base classifier. For fair comparison, we set

$$\gamma_s = \mathrm{e}^{-\beta \mathrm{DIST}_k^2(D^s, D^T)} \tag{30}$$

where $\beta > 0$ is the bandwidth parameter to control the spread of $\mathrm{DIST}_k(D^s, D^T)$ and $\mathrm{DIST}_k(D^s, D^T)$ is the *maximum mean discrepancy* (MMD) [49] for measuring the data distributions between the sth the source domain and the target domain. MMD is an effective nonparametric distance metric for comparing data distributions in the RKHS, namely, $\mathrm{DIST}_k(D^s, D^T) = \left\| (1/n_s) \sum_{i=1}^{n_s} \phi(\mathbf{x}_i^s) - (1/n_T) \sum_{i=1}^{n_T} \phi(\mathbf{x}_i^T) \right\|_{\mathcal{H}}^2$. In this paper, we adopt MMD owing to its effectiveness and simplicity [9], [41], [49]. In the experiment, we further normalize the sum of the weights γ_s's as 1 and empirically set $\beta = 100$.

2) Detailed Setup for Document Retrieval: The 20 newsgroups dataset and the email spam dataset are used for document retrieval. For this application, we compare our two methods FastDAM and UniverDAM with the baseline SVMs (i.e., SVM_T, SVM_S), FR, MCC-SVM, and Multi-KMM. We use the same settings for these methods as in Section V-B1.

In our methods, linear kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i' \mathbf{x}_j$) and polynomial kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i' \mathbf{x}_j + 1)^a$) are considered as the base kernels to train the base classifiers, where $a = 1.1, 1.2, \ldots, 1.5$ for both datasets. And the linear kernel is used as the default kernel because of its good performance on text datasets. Therefore, we have in total 18 base classifiers from 3 sources and 6 base kernels. In the default setting, the regularization parameter C is set as 1 for all methods, and we set $\lambda_L = \lambda_D = 1$ for FastDAM and $\lambda_L = \lambda_{D_1} = \lambda_{D_2} = 1$ for UniverDAM on both datasets. As in the video concept detection task, we use (30) to determine the weight γ_s for the sth base classifier, and we further normalize the sum of the weights γ_s's as 1. We empirically set β as 10 000 for the 20 newsgroups dataset and β as 100 for the email spam dataset. In Section V-D, we take the 20 newsgroups dataset as an example to analyze the performance variations of different methods with respect to all these parameters $C, \lambda_L, \lambda_D, \lambda_{D_1}, \lambda_{D_2}$, and β.

C. Performance Comparisons

1) Results of Video Concept Detection: The MAPs over all 36 concepts on the TRECVID 2005 dataset is given in Table V. From the table, we observe that the domain adaptation methods MCC-SVM, FR, Multi-KMM, and A-SVM outperform SVM_T and SVM_S, which demonstrates that the instances from source domains and target domain can be used to improve generalization performance in the target domain. MCC-SVM and A-SVM achieve similar performance in terms of MAP over 36 concepts. Multi-KMM is worse than MCC-SVM, FR, and A-SVM, possibly because it is difficult to estimate the means to be shifted with many source domains.

In our initial version of this paper [4], we have shown that our FastDAM_50 and FastDAM_200 using the auxiliary classifiers (i.e., the SVM_S classifiers) can respectively improve the performance from 26.4% (SVM_S) to 29.8% and 30.9% and the prediction of FastDAM is also much faster than other domain adaptation methods because of the sparse solution. In Table V, we report the results of FastDAM using the FR classifiers as the base classifiers. The MAPs of FastDAM_50 and FastDAM_200 using the better base classifiers are further improved to 32.2% and 32.6%, respectively, which are better than SVM_T, SVM_S and other cross-domain learning methods. When compared with FR (resp. A-SVM), the relative MAP improvements of FastDAM_50 and FastDAM_200 are 5.9% and 7.2% (resp. 14.6% and 16.0%), respectively. These results clearly demonstrate that FastDAM can learn a robust target classifier for domain adaptation by leveraging a set of prelearned base classifiers.

We additionally report the results of SVM_S, MCC-SVM, and FR in the single-source domain setting in which all the instances from five source domains are considered as one source domain. The MAPs of SVM_S, MCC-SVM, and FR in the single source domain setting are 23.4%, 28.4%, and 28.7%, respectively, and they are worse than the results from the multiple source domain setting reported in Table V.

TABLE VI

MEANS AND STANDARD DEVIATIONS (%) OF APS OF ALL METHODS WITH m POSITIVE AND m NEGATIVE TRAINING INSTANCES FROM THE TARGET DOMAIN ON THE 20 NEWSGROUPS DATASET. THE RESULTS SHOWN IN BOLDFACE ARE SIGNIFICANTLY BETTER THAN THE OTHERS, JUDGED BY THE t-TEST WITH A SIGNIFICANCE LEVEL AT 0.01

(a) rec versus sci

m	SVM_T	SVM_S	MCC-SVM	FR	Multi-KMM	FastDAM	UniverDAM
0	–	95.44 ± 0.00	–	–	97.58 ± 0.00	97.77 ± 0.00	**99.69±0.00**
2	67.58 ± 6.82	95.44 ± 0.01	95.45 ± 0.26	93.87 ± 2.86	97.69 ± 0.07	97.81 ± 0.05	**99.69±0.01**
4	73.56 ± 7.46	95.44 ± 0.02	95.61 ± 0.41	94.87 ± 2.63	97.81 ± 0.09	97.87 ± 0.05	**99.69±0.01**
6	75.77 ± 5.79	95.45 ± 0.03	95.66 ± 0.40	95.34 ± 1.58	97.91 ± 0.07	97.93 ± 0.07	**99.70±0.01**
10	81.55 ± 6.87	95.45 ± 0.03	95.82 ± 0.56	95.94 ± 1.43	98.04 ± 0.08	98.01 ± 0.11	**99.71±0.02**
15	87.58 ± 4.72	95.46 ± 0.05	96.03 ± 0.43	96.69 ± 0.95	98.18 ± 0.09	98.04 ± 0.12	**99.72±0.01**
20	90.52 ± 3.70	95.46 ± 0.05	96.17 ± 0.51	97.31 ± 0.87	98.31 ± 0.09	98.11 ± 0.13	**99.72±0.01**

(b) comp versus rec

m	SVM_T	SVM_S	MCC-SVM	FR	Multi-KMM	FastDAM	UniverDAM
0	–	97.53 ± 0.00	–	–	98.78 ± 0.00	99.03 ± 0.00	**99.82±0.00**
2	67.96 ± 10.19	97.53 ± 0.01	97.49 ± 0.41	97.34 ± 1.25	98.80 ± 0.03	99.03 ± 0.01	**99.82±0.00**
4	75.53 ± 8.28	97.52 ± 0.01	97.47 ± 0.54	97.36 ± 1.03	98.83 ± 0.02	99.04 ± 0.01	**99.82±0.00**
6	78.45 ± 9.18	97.52 ± 0.01	97.28 ± 0.65	97.31 ± 1.18	98.85 ± 0.03	99.05 ± 0.02	**99.82±0.00**
10	83.71 ± 5.60	97.53 ± 0.03	97.47 ± 0.44	97.86 ± 0.58	98.90 ± 0.04	99.06 ± 0.03	**99.82±0.01**
15	88.71 ± 5.72	97.52 ± 0.04	97.58 ± 0.53	98.47 ± 0.37	98.97 ± 0.04	99.07 ± 0.03	**99.82±0.01**
20	92.50 ± 3.56	97.51 ± 0.04	97.79 ± 0.38	98.52 ± 0.36	99.01 ± 0.03	99.08 ± 0.03	**99.82±0.01**

(c) sci versus comp

m	SVM_T	SVM_S	MCC-SVM	FR	Multi-KMM	FastDAM	UniverDAM
0	–	91.94 ± 0.00	–	–	91.16 ± 0.00	94.68 ± 0.00	**98.44±0.00**
2	66.95 ± 6.50	91.93 ± 0.02	92.07 ± 0.25	89.34 ± 2.30	91.51 ± 0.17	94.77 ± 0.04	**98.46±0.02**
4	73.17 ± 6.33	91.94 ± 0.02	92.32 ± 0.36	90.41 ± 2.35	91.92 ± 0.17	94.92 ± 0.11	**98.50±0.02**
6	76.69 ± 6.10	91.94 ± 0.03	92.56 ± 0.31	91.63 ± 1.93	92.33 ± 0.23	95.04 ± 0.13	**98.53±0.02**
10	81.65 ± 3.54	91.94 ± 0.03	92.94 ± 0.43	93.13 ± 1.72	92.85 ± 0.38	95.28 ± 0.15	**98.59±0.03**
15	85.28 ± 2.88	91.93 ± 0.04	93.43 ± 0.49	94.48 ± 1.31	93.53 ± 0.33	95.64 ± 0.23	**98.68±0.05**
20	88.22 ± 2.55	91.93 ± 0.06	93.76 ± 0.51	95.27 ± 0.95	94.04 ± 0.28	95.87 ± 0.25	**98.73±0.04**

2) Results of Document Retrieval: Table VI shows the means and standard deviations of APs of all methods on the 20 newsgroups dataset. When the number of positive and negative training instances (i.e., m) from the target domain increases, the performances of most methods improve in terms of the means of APs. We observe that SVM_S achieves good results by only using the labeled instances from the source domains, possibly because some source domains are highly relevant to the target domain. This conjecture is also supported by measuring the distances between the source domains and the target domain with the MMD criterion. We therefore use the SVM_S classifiers as the base classifiers in our FastDAM and UniverDAM. Multi-KMM is generally better than SVM_S, MCC_SVM, and FR, which demonstrates that Multi-KMM can successfully shift the means of source domains toward the target domain on this dataset. Our method FastDAM outperforms other algorithms in most cases except that it performs slightly worse than Multi-KMM in three cases when setting $m = 10, 15$, and 20 (see setting (a) in Table VI) and in some cases FastDAM only performs slightly better than Multi-KMM and FR when setting $m = 10, 15$, and 20 (see settings (b) and (c) in Table VI). The explanation is that the existing domain adaptation algorithms such as Multi-KMM and FR can achieve good performances when there are more labeled target samples. UniverDAM achieves the best results in all the cases in terms of the means of APs, which demonstrates the effectiveness of our DAM framework. Moreover, UniverDAM is also significantly better than other methods judged by the t-test with a significance level at 0.01. It demonstrates that the data-dependent regularizer for the Universum suggested in [32] and [33] is suitable for this binary-class document retrieval problem in which the instances from the source domains can be effectively used as the Universum for domain adaptation. Since the SVM_S classifiers are used as the base classifiers in FastDAM and UniverDAM, our methods can successfully handle the extreme case in which there are no labeled instances in the target domain. In such an extreme case, we do not consider the loss of the labeled training data from the target domain in our DAM framework. However, other cross-domain learning methods such as MCC-SVM and FR cannot cope with such an extreme case.

Table VII lists the results of all methods on the email spam dataset. Since the performance of FR is much better than that of SVM_S, we use the FR classifiers as the base classifiers in FastDAM and UniverDAM on this dataset. In terms of the means of APs, our two methods outperform the other methods, and UniverDAM achieves the best result,

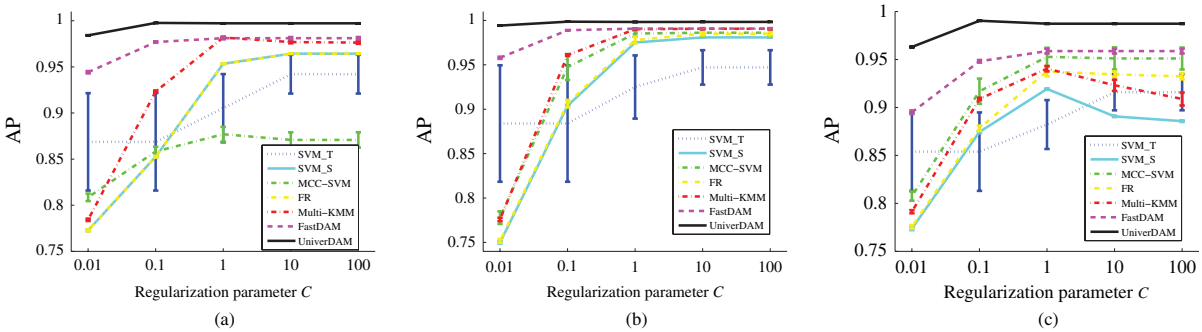

Fig. 2. Means and standard deviations (%) of APs of all methods on the 20 Newsgroups dataset with different regularization parameter $C = 0.01, 0.1, 1, 10,$ and 100. For better visualization, see the colored PDF file. (a) Rec versus sci, (b) comp versus rec, and (c) sci versus comp.

TABLE VII

MEANS AND STANDARD DEVIATIONS (%) OF APS OF ALL METHODS WITH 10 POSITIVE AND 10 NEGATIVE TRAINING INSTANCES FROM THE TARGET DOMAIN ON THE EMAIL SPAM DATASET. THE RESULTS SHOWN IN BOLDFACE ARE SIGNIFICANTLY BETTER THAN THE OTHERS, JUDGED BY THE t-TEST WITH A SIGNIFICANCE LEVEL AT 0.01

	SVM_T	SVM_S	MCC-SVM	FR	Multi-KMM	FastDAM	UniverDAM
AP	83.01 ± 3.08	68.62 ± 0.09	76.89 ± 2.75	84.13 ± 3.46	74.10 ± 1.36	85.36 ± 2.48	**87.05±2.94**

TABLE VIII

AVERAGE TRAINING AND TESTING TIME (IN SECONDS) OVER 10 ROUNDS OF EXPERIMENTS FOR ALL METHODS ON THE FIRST SETTING (i.e., REC VERSUS SCI) OF THE 20 NEWSGROUPS DATASET. NOTE THAT THE TRAINING TIME OF BOTH FASTDAM AND UNIVERDAM CONSISTS OF TWO PARTS (i.e., THE CALCULATION OF THE VIRTUAL LABELS AND THE LEARNING OF THE CLASSIFIER)

	SVM_T	SVM_S	MCC-SVM	FR	Multi-KMM	FastDAM	UniverDAM
Training	0.04	69.38	72.96	69.19	201.53	$133.41 + 8.87$	$133.41 + 284.26$
Testing	1.10	125.80	128.61	126.51	126.93	24.64	146.85

which again demonstrates the effectiveness of our methods. Moreover, UniverDAM is also significantly better than other methods judged by the t-test with a significance level at 0.01.

Moreover, we take the first setting (i.e., rec versus sci) of the 20 newsgroups dataset as an example to compare the average training and test time of all methods. All methods are performed on an IBM workstation (2.13 GHz CPU with 16 GB RAM), and we set $m = 20$ for all methods. The performances of all methods are shown in Table VIII. Note that for our methods FastDAM and UniverDAM we need to obtain the virtual labels of the unlabeled training instances in advance. We assume that the base classifiers (i.e., SVM_S) can be learned in an offline fashion. The calculation of the virtual labels takes 133.41 s on average. Because of the sparse solution, the prediction of FastDAM is much faster than the other methods except SVM_T which only uses 20 positive and 20 negative training instances. While a large number of unlabeled training data from the source domain are used as Universum, the testing time of UniverDAM is comparable with other methods (i.e., SVM_S, MCC-SVM, FR, and Multi-KMM).

D. Parameter Analysis for Different Methods

In this subsection, we evaluate the performance variations with respect to the regularization parameter C used in all methods, the parameters λ_L, λ_D, and β used in FastDAM, and the parameters $\lambda_L, \lambda_{D_1}, \lambda_{D_2}$, and β used in UniverDAM by using the 20 newsgroups dataset in which 20 positive and 20 negative labeled instances from the target domain are used for training. In the default setting, we set the regularization parameter $C = 1$ for all methods, $\lambda_L = \lambda_D = \lambda_{D_1} = \lambda_{D_2} = 1$, and $\beta = 10\,000$ for both FastDAM and UniverDAM. When evaluating the performance variations with respect to one parameter, we fix the other parameters as their default values.

1) Performance Variations w.r.t. the Regularization Parameter C: We compare all methods on the three settings of the 20 newsgroups dataset by using different C in Fig. 2, where C is set as $0.01, 0.1, 1, 10,$ and 100. We observe that the performances of most methods tend to saturate when C becomes large. Our method FastDAM is generally better than other methods, and UniverDAM consistently achieves the best performances by using different C's. Moreover, the large improvement of UniverDAM over the other methods clearly demonstrates the successful utilization of the source domain data as the Universum. We have similar observations when using different numbers of training samples from the target domain.

2) Performance Variations w.r.t. the Tradeoff Parameters $\lambda_L, \lambda_D, \lambda_{D_1}$, and λ_{D_2}: We conduct two experiments to study the tradeoff parameters $\lambda_L, \lambda_D, \lambda_{D_1}$, and λ_{D_2}. First, we evaluate the effectiveness of transferring source information. Specifically, we set $\lambda_D = \lambda_{D_1} = \lambda_{D_2}$ for our methods FastDAM and UniverDAM as $0.1, 1, 10, 100,$ and 1000. Other parameters

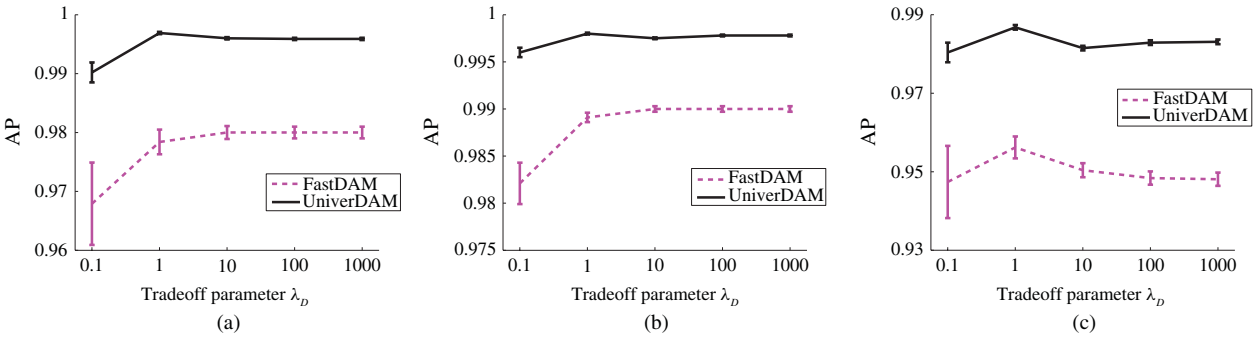

Fig. 3. Means and standard deviations (%) of APs of FastDAM and UniverDAM on the 20 newsgroups dataset with different $\lambda_D = \lambda_{D_1} = \lambda_{D_2} = 0.1, 1, 10, 100$, and 1000. (a) Rec versus sci, (b) comp versus rec, and (c) sci versus comp.

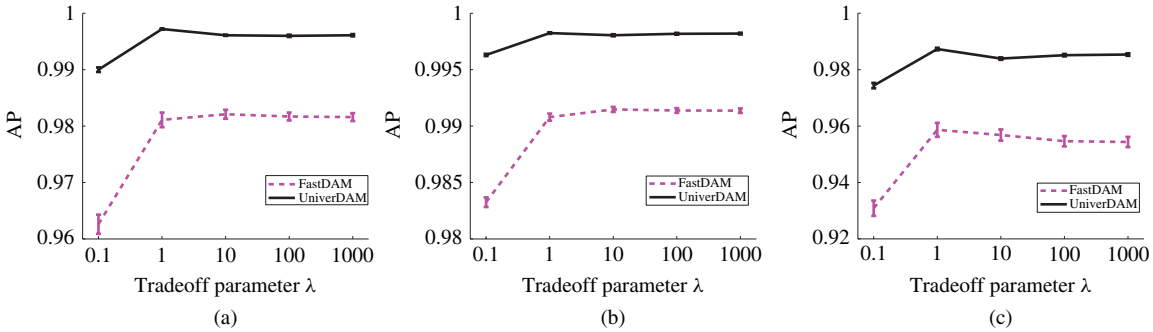

Fig. 4. Means and standard deviations (%) of APs of FastDAM and UniverDAM on the 20 newsgroups dataset with different tradeoff parameter $\lambda = 0.1, 1, 10, 100$, and 1000. (a) Rec versus sci, (b) comp versus rec, and (c) sci versus comp.

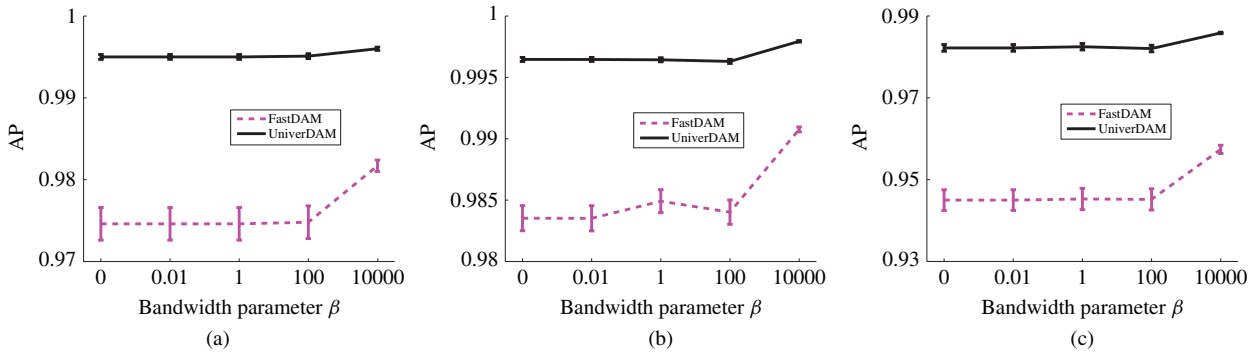

Fig. 5. Means and standard deviations (%) of APs of FastDAM and UniverDAM on the 20 newsgroups dataset with different bandwidth parameter $\beta = 0, 0.01, 1, 100$, and 10 000 in (30). (a) Rec versus sci, (b) comp versus rec, and (c) sci versus comp.

are set as default values. The performance variations with respect to different λ_D are shown in Fig. 3. When setting $\lambda_D \leq 1$, it can be observed from Fig. 3 that the performances of both FastDAM and UniverDAM increase when λ_D increases, which demonstrates that it is beneficial to utilize the unlabeled data for domain adaptation. We also observe that their performances become stable when setting $\lambda_D \geq 10$.

Second, we evaluate the performance variations with respect to all the tradeoff parameters. To avoid too many combinations of the parameters, we fix $\lambda_L = \lambda_D = \lambda_{D_1} = \lambda_{D_2} = \lambda$. We report the results of FastDAM and UniverDAM by using different λ in Fig. 4, where λ is set as $0.1, 1, 10, 100$, and 1000. From Fig. 4, we observe that the performances of both FastDAM and UniverDAM do not change much when setting $\lambda \geq 1$, and UniverDAM achieves the best performance when setting $\lambda = 1$.

3) Performance Variations w.r.t. the Bandwidth Parameter β: Recall that β is the bandwidth parameter for the calculation of γ_s (see (30)), and we normalize the sum of γ_s's to 1. In Fig. 5, we show the performances of FastDAM and UniverDAM by using different β, where β is set as $0, 0.01, 1, 100$, and 10 000. When setting $\beta = 0$, we have equal weights for all source domains (i.e., $\gamma_s = 1/P$, $\forall s = 1, \ldots, P$). From Fig. 5, we observe that both FastDAM and UniverDAM using $\beta = 10\,000$ achieve better performances when compared with $\beta = 0$, which demonstrates that it is beneficial to adopt the MMD criterion to measure the distribution mismatch between each source domain and the target domain. We also observe that UniverDAM achieves much better performances than FastDAM, which again demonstrates the effectiveness of the Universum constructed by using the source domain data for document retrieval.

VI. Conclusion

We have proposed a new framework, referred to as DAM, for multiple source domain adaptation. It learns a robust target classifier for predicting labels of test instances from the target domain by leveraging a set of prelearned base classifiers. Any types of classifiers, such as the standard SVM classifier learned with the labeled instances from the source domains or the FR classifier [3] learned with the labeled instances from the source domains and target domain, can be readily used as base classifiers in our framework. With the base classifiers, we introduced a new domain-dependent regularizer based on *smoothness assumption*, which enforces that the target classifier share similar decision values with the relevant base classifiers on the unlabeled instances from the target domain. This newly proposed regularizer can be readily combined with many kernel methods, such as SVM, SVR, LS-SVM, and so on, for domain adaptation. Under this framework, we also developed two methods, referred to as FastDAM and UniverDAM. In FastDAM, we incorporated the proposed domain-dependent regularizer into LS-SVM as well as employed a sparsity regularizer based on the ϵ-insensitive loss to enforce the sparsity on the target classifier. In FastDAM, the label prediction of test instances is very fast, making it suitable for large-scale applications (e.g., video concept detection) with a large number of test instances. In order to further enhance the generalization ability of the target classifier, in UniverDAM we additionally introduced another regularizer suggested in [32] and [33] into the objective function of FastDAM by treating the instances from the source domains as the Universum. We also showed that the final formulations of FastDAM and UniverDAM share a similar form to that of ϵ-SVR, which can be readily solved by using the state-of-the-art solvers such as LIBSVM [39]. Comprehensive experiments on the video concept detection and document retrieval tasks clearly demonstrate the effectiveness of our two methods.

In this paper, we adopted the simple but effective non-parametric criterion MMD [49] to define the weight γ_s in (30) which measures the distribution relevance between the sth source domain and the target domain. In the future, we will investigate other criteria in order to better measure the distribution mismatch between the source and target domains.

Appendix

Proof of Theorem 2

Proof: The KKT conditions of the proposed formulation of UniverDAM for the sth source domain are stated as follows:

$$\xi_{s,i}, \xi^*_{s,i} \geq 0 \qquad (31)$$

$$\alpha_{s,i}\left(\epsilon_s + \xi_{s,i} - \mathbf{w}'\phi(\mathbf{x}_i^s) - b + f^T(\mathbf{x}_i^s)\right) = 0 \qquad (32)$$

$$\alpha^*_{s,i}\left(\epsilon_s + \xi^*_{s,i} + \mathbf{w}'\phi(\mathbf{x}_i^s) + b - f^T(\mathbf{x}_i^s)\right) = 0 \qquad (33)$$

Assuming $\epsilon_s > \max_{i=1,\ldots,n_s}\left|\mathbf{w}'\phi(\mathbf{x}_i^s) + b - f^T(\mathbf{x}_i^s)\right|$, we have $\epsilon_s > \mathbf{w}'\phi(\mathbf{x}_i^s) + b - f^T(\mathbf{x}_i^s)$ and $\epsilon_s > -\mathbf{w}'\phi(\mathbf{x}_i^s) - b + f^T(\mathbf{x}_i^s)$. With the KKT conditions in (31), (32), and (33), it is easy to verify $\alpha_{s,i}, \alpha^*_{s,i} = 0$ for every training instance from the source domains.

Therefore, if $\epsilon_s > \max_{i=1,\ldots,n_s}\left|\mathbf{w}'\phi(\mathbf{x}_i^s) + b - f^T(\mathbf{x}_i^s)\right|$ holds for each source domain, the optimization problem (21) (resp. the dual form (27)) of UniverDAM can be simplified into the optimization problem (13) (resp. the dual form (17)), namely, UniverDAM reduces to FastDAM. ■

References

[1] P. Wu and T. G. Dietterich, "Improving SVM accuracy by training on auxiliary data sources," in *Proc. Int. Conf. Mach. Learn.*, Banff, AB, Canada, Jul. 2004, pp. 871–878.

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

[3] H. Daumé, III, "Frustratingly easy domain adaptation," in *Proc. Annu. Meeting Assoc. Comput. Linguist.*, Prague, Czech Republic, Jun. 2007, pp. 256–263.

[4] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua, "Domain adaptation from multiple sources via auxiliary classifiers," in *Proc. Int. Conf. Mach. Learn.*, Montreal, QC, Canada, Jun. 2009, pp. 289–296.

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

[6] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. Annu. Conf. Comput. Learn. Theory*, Madison, WI, Jul. 1998, pp. 92–100.

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

[8] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.

[9] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank, "Domain transfer SVM for video concept detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, Jun. 2009, pp. 1375–1381.

[10] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. Int. Conf. Mach. Learn.*, Bled, Slovenia, Jun. 1999, pp. 200–209.

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

[12] X. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, Tech. Rep. 1530, Jul. 2008.

[13] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Advances in Neural Information Processing Systems 19*. Cambridge, MA: MIT Press, 2007, pp. 601–608.

[14] A. J. Storkey and M. Sugiyama, "Mixture regression for covariate shift," in *Advances in Neural Information Processing Systems 19*. Cambridge, MA: MIT Press, 2007, pp. 1337–1344.

[15] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press, 2008, pp. 1433–1440.

[16] S. Bickel, C. Sawade, and T. Scheffer, "Transfer learning by distribution matching for targeted advertising," in *Advances in Neural Information Processing Systems 21*. Cambridge, MA: MIT Press, 2009.

[17] P. Luo, F. Zhuang, H. Xiong, Y. Xiong, and Q. He, "Transfer learning from multiple source domains via consensus regularization," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Napa Valley, CA, Oct. 2008, pp. 103–112.

[18] G. Schweikert, C. Widmer, B. Schölkopf, and G. Rätsch, "An empirical analysis of domain adaptation algorithms for genomic sequence analysis," in *Advances in Neural Information Processing Systems 21*. Cambridge, MA: MIT Press, 2009, pp. 1433–1440.

[19] M. T. Rosenstein, Z. Marx, and L. P. Kaelbling, "To transfer or not to transfer," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, Dec. 2005.

[20] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Advances in Neural Information Processing Systems 19*. Cambridge, MA: MIT Press, 2007, pp. 137–144.

[21] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press, 2008, pp. 129–136.
[22] K. Crammer, M. Kearns, and J. Wortman, "Learning from multiple sources," *J. Mach. Learn. Res.*, vol. 9, pp. 1757–1774, Aug. 2008.
[23] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Advances in Neural Information Processing Systems 21*. Cambridge, MA: MIT Press, 2009, pp. 1041–1048.
[24] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," *Comput. Res. Reposit.*, vol. abs/0902.3, no. 2007, pp. 1–12, 2009.
[25] S. Ben-David, T. Luu, T. Lu, and D. Pál, "Impossibility theorems for domain adaptation," *J. Mach. Learn. Res.*, vol. 9, pp. 129–136, May 2010.
[26] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *Proc. Int. Conf. Very Large Data Bases*, Toronto, ON, Canada, Aug. 2004, pp. 180–191.
[27] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, nos. 1–2, pp. 151–175, May 2010.
[28] T. Evgeniou, C. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *J. Mach. Learn. Res.*, vol. 6, pp. 615–637, Apr. 2005.
[29] T. Kato, H. Kashima, M. Sugiyama, and K. Asai, "Multi-task learning via conic programming," in *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press, 2008, pp. 737–744.
[30] D. Sheldon, "Graphical multi-task learning," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, Dec. 2008.
[31] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.
[32] F. H. Sinz, O. Chapelle, A. Agarwal, and B. Shölkopf, "An analysis of inference with the universum," in *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press, 2008, pp. 1369–1376.
[33] J. Weston, R. Collobert, F. Sinz, L. Bottou, and V. Vapnik, "Inference with the universum," in *Proc. Int. Conf. Mach. Learn.*, Pittsburgh, PA, Jun. 2006, pp. 1009–1016.
[34] V. Vapnik, "Transductive inference and semi-supervised learning," in *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006, pp. 454–472.
[35] T. Van Gestel, J. A. K. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. de Moor, and J. Vandewalle, "Benchmarking least squares support vector machine classifiers," *Mach. Learn.*, vol. 54, no. 1, pp. 5–32, Jan. 2004.
[36] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: From transductive to semi-supervised learning," in *Proc. Int. Conf. Mach. Learn.*, Bonn, Germany, Jun. 2005, pp. 824–831.
[37] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.
[38] I. W. Tsang and J. T. Kwok, "Large-scale sparsified manifold regularization," in *Advances in Neural Information Processing Systems 19*. Cambridge, MA: MIT Press, 2007, pp. 1401–1408.
[39] C.-C. Chang and C.-J. Lin. (2001). *LIBSVM: A Library for Support Vector Machines* [Online]. Available: http://www.csie.ntu.edu.tw/∼cjlin/libsvm
[40] T. D. Bie and N. Cristianini, "Semi-supervised learning using semi-definite programming," in *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006, pp. 119–135.
[41] L. Duan, D. Xu, I. W. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, Jun. 2010, pp. 1959–1966.
[42] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[43] S. Ozawa, A. Roy, and D. Roussinov, "A multitask learning model for online pattern recognition," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, pp. 430–445, Mar. 2009.
[44] M. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis, "Large-scale concept ontology for multimedia," *IEEE Multimedia Mag.*, vol. 13, no. 3, pp. 86–91, Jul.–Sep. 2006.
[45] S.-F. Chang, J. He, Y.-G. Jiang, E. E. Khoury, C.-W. Ngo, A. Yanagawa, and E. Zavesky, "Columbia University/VIREO-cityU/IRIT TRECVID2008 high-level feature extraction and interactive video search," in *Proc. NIST TRECVID Workshop*, Gaithersburg, MD, Nov. 2008, pp. 1–16.
[46] A. Yanagawa, W. Hsu, and S.-F. Chang, "Columbia University's baseline detectors for 374 LSCOM semantic visual concepts," Dept. Electr. Eng., Columbia University, New York, ADVENT Tech. Rep. 222-2006-8, Mar. 2007.
[47] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVID," in *Proc. ACM Int. Workshop Multimedia Inf. Retrieval*, Santa Barbara, CA, Oct. 2006, pp. 321–330.
[48] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. Cambridge, MA: MIT Press, 1999, pp. 61–74.
[49] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. 49–57, Jul. 2006.

Lixin Duan received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2008. He is currently pursuing the Ph.D. degree with the School of Computer Engineering, Nanyang Technological University, Singapore.

He was a recipient of the Microsoft Research Asia Fellowship in 2009 and the Best Student Paper Award at the IEEE International Conference on Computer Vision and Pattern Recognition in 2010.

Dong Xu (M'07) received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2001 and 2005, respectively.

He was with Microsoft Research Asia, Beijing, China, and the Chinese University of Hong Kong, Shatin, Hong Kong, for more than two years, while pursuing the Ph.D. degree. He was a Post-Doctoral Research Scientist with Columbia University, New York, NY, for one year. He is currently an Assistant Professor with Nanyang Technological University, Singapore. His current research interests include computer vision, statistical learning, and multimedia content analysis.

Dr. Xu was the co-author of a paper that won the Best Student Paper Award in the prestigious IEEE International Conference on Computer Vision and Pattern Recognition in 2010.

Ivor W. Tsang received his Ph.D. degree in computer science from the Hong Kong University of Science and Technology in 2007.

He is currently an Assistant Professor with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. He is also the Deputy Director of the Center for Computational Intelligence, NTU.

Dr. Tsang received the prestigious IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding 2004 Paper Award in 2006, and the second class prize of the National Natural Science Award 2008, China in 2009. His research also earned him the Best Paper Award at ICTAI'11, the Best Student Paper Award at CVPR'10, and the Best Paper Award from the IEEE Hong Kong Chapter of Signal Processing Postgraduate Forum in 2006. He was also conferred with the Microsoft Fellowship in 2005.