Domain Adaptation from Multiple Sources via Auxiliary Classifiers

Lixin Duan S080003@NTU.EDU.SG
Ivor W. Tsang IVORTSANG@NTU.EDU.SG
Dong Xu DONGXU@NTU.EDU.SG
School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798

Tat-Seng Chua CHUATS@COMP.NUS.EDU.SG
Department of Computer Science, National University of Singapore, 21 Lower Kent Ridge Road, Singapore 119077

Abstract

We propose a multiple source domain adaptation method, referred to as Domain Adaptation Machine (DAM), to learn a robust decision function (referred to as *target classifier*) for label prediction of patterns from the target domain by leveraging a set of pre-computed classifiers (referred to as *auxiliary/source classifiers*) independently learned with the labeled patterns from multiple source domains. We introduce a new data-dependent regularizer based on *smoothness assumption* into Least-Squares SVM (LS-SVM), which enforces that the target classifier shares similar decision values with the auxiliary classifiers from relevant source domains on the unlabeled patterns of the target domain. In addition, we employ a sparsity regularizer to learn a sparse target classifier. Comprehensive experiments on the challenging TRECVID 2005 corpus demonstrate that DAM outperforms the existing multiple source domain adaptation methods for video concept detection in terms of effectiveness and efficiency.

1. Introduction

Collection of labeled patterns requires expensive and time-consuming efforts of human annotators. Domain adaptation methods [1] were proposed (Wu & Dietterich, 2004; Blitzer et al., 2006; Daumé III, 2007) to learn robust classifiers with only a few or even no labeled patterns from the target domain by leveraging a large amount of labeled training data from other domains (referred to as auxiliary/source domains). These methods demonstrated that the labeled patterns collected from other domains are useful for classifying the patterns from the target domain in many real applications, such as sentiment classification, text categorization, WiFi localization and video concept detection.

To utilize all training patterns from the source and target domains, Blitzer et al. (2006) proposed Structural Correspondence Learning (SCL) 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 (2007) 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. (2007) proposed Adaptive Support Vector Machine (A-SVM) to learn a new SVM classifier $f^T(\mathbf{x})$ for target domain, which is adapted from an existing classifier $f^s(\mathbf{x})$ trained with the patterns from the source domain.

However, numerous unlabeled patterns in the target domain are not exploited in the above domain adaptation methods (Wu & Dietterich, 2004; Daumé III, 2007; Yang et al., 2007). As shown in (Joachims, 1999; Belkin et al., 2006), such unlabeled patterns can also be employed to improve the generalization performance. When there are only a few or even no labeled patterns available in the target domain, the classifiers can be trained with the patterns from the source domains. In such an extreme case, several domain adaptation methods (Huang et al., 2007; Storkey & Sugiyama, 2007) were proposed to cope with the inconsistency of data distribution (such as *covariate shift* (Storkey & Sugiyama, 2007) or *sampling selection bias* (Huang et al., 2007)). These methods re-weighted the training patterns from the source domain by leveraging the unlabeled data from the target domain such that the statistics of pat-

[1] Domain adaptation is different from Semi-Supervised 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.

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Montreal, Canada, 2009.

terns from both domains are matched.

Recently, several domain adaptation methods (Crammer et al., 2008; Luo et al., 2008; Mansour et al., 2009) were proposed to learn robust classifiers with the diverse training data from multiple source domains. Crammer et al. (2008) assumed that the distributions of multiple sources are the same, and the change of labels is due to the varying amount of noise. Luo et al. (2008) proposed to maximize the consensus of predictions from multiple sources. Mansour et al. (2009) estimated the data distribution of each source to reweight the patterns from different 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 (Schweikert et al., 2009), which is also known as *negative transfer* (Rosenstein et al., 2005).

Contribution : In this paper, we focus on the setting in which there are multiple source domains, which is referred to as *multiple source domain adaptation*. We develop a new domain adaptation method, referred to as Domain Adaptation Machine (DAM), to learn a robust decision function (referred to as *target classifier*) for label prediction of patterns in the target domain by leveraging a set of precomputed classifiers (referred to as *auxiliary/source classifiers*) independently learnt with the labeled samples from multiple source domains. Motivated from Manifold Regularization (MR) (Belkin et al., 2006) and the graph based Multi-Task Learning (MTL) (Evgeniou et al., 2005; Kato et al., 2008), a data-dependent regularizer based on *smoothness assumption* is proposed to enforce that the learned target classifier should have similar decision values with the auxiliary classifiers of relevant source domains on the unlabeled patterns of the target domain. In addition, we employ a sparsity regularizer to enforce the sparsity of the target classifier.

Significance : We test DAM for the challenging video concept detection task on the TRECVID 2005 data set, which is collected from six channels including three English channels, two Chinese channels and one Arabic channel. The data distributions of six channels are quite different, making it suitable for evaluating multiple source domain adaptation methods. Our comprehensive experiments demonstrate that DAM outperforms the existing domain adaptation methods. Moreover, with the sparsity regularizer, the prediction of DAM is much faster than other domain adaptation methods, making it suitable for large-scale applications such as video concept detection.

2. Domain Adaptation Machine

In the sequel, the transpose of vector / matrix is denoted by the superscript $'$. Let us also define $\mathbf{I}$ as the identity matrix and $\mathbf{0}, \mathbf{1} \in \Re^n$ as the zero vector and the vector of all ones, respectively. The inequality $\mathbf{u} = [u_1, \ldots, u_n]' \geq \mathbf{0}$ means that $u_i \geq 0$ for $i = 1, \ldots, n$.

We focus on the multiple source domain adaptation. Suppose there are plenty of unlabeled data and only few labeled data are available in the target domain. Denote the labeled and unlabeled patterns 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 $D^T = D_l^T \cup D_u^T$ as the data set from the target domain with the size $n_T = n_l + n_u$, and $D^s = (\mathbf{x}_i^s, y_i^s)|_{i=1}^{n_s}, s = 1, 2, \ldots, P$ as the data set from the s-th source domain, where P is the total number of source domains.

2.1. Domain Adaptation from Auxiliary Classifiers

Yang et al. (2007) proposed Adaptive SVM (A-SVM), in which a new SVM classifier $f^T(\mathbf{x})$ is adapted from the existing auxiliary classifiers $f^s(\mathbf{x})$'s trained with the patterns from the auxiliary sources. Specifically, the new decision function is formulated as:

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

where the perturbation function $\Delta f(\mathbf{x})$ is learned using the labeled data D_l^T from the target domain, and $\gamma_s \in (0, 1)$ is the weight[2]of each auxiliary classifier f^s and $\sum_s \gamma_s = 1$. As shown in (Yang et al., 2007), 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 i-th labeled pattern 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 learnt coefficient of the i-th pattern from the s-th source domain. Then the decision function (1) becomes:

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

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

2.2. Smoothness Assumption for Domain Adaptation

In Manifold Regularization (Belkin et al., 2006), the decision function is enforced to be smooth on the data manifold, namely, the two nearby patterns in a high-density region should share similar decision values. For domain adaptation, we also assume that the target classifier f^T should

[2]In (Yang et al., 2007), the equal weights γ_s are used for all auxiliary classifiers in the experiments.

have similar decision values with the pre-computed auxiliary classifiers. Let us define γ_s as the weight for measuring the relevance between the s-th source domain and the target domain (See Section 3.1 for more discussions on γ_s). For the i-th pattern $\mathbf{x}_i$, we also denote $f_i^T = f^T(\mathbf{x}_i)$ and $f_i^s = f^s(\mathbf{x}_i)$, where f^s is the auxiliary function of the s-th source domain. If two domains are relevant (*i.e.*, γ_s is large), we enforce f_i^s should be close to f_i^T.

For the unlabeled target patterns D_u^T in the target domain, let us define the decision values from the target classifier and the s-th auxiliary 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. We define a data-dependent regularizer for the target classifier f^T:

Definition 1. Data-Dependent Regularizer for Domain Adaptation

$$\Omega_D(\boldsymbol{f}_u^T) = \frac{1}{2}\sum_{i=n_l+1}^{n_T}\sum_s \gamma_s (f_i^T - f_i^s)^2 = \frac{1}{2}\sum_s \gamma_s \|\boldsymbol{f}_u^T - \boldsymbol{f}_u^s\|^2. \quad (3)$$

We note that (Evgeniou et al., 2005; Kato et al., 2008) similarly defined a graph based regularizer for Multi-Task Learning (MTL), which is based on two MTL functions f^s and f^T in the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$:

$$\Omega_{\mathcal{G}}(f^1, f^2, \ldots, f^P, f^T) = \frac{1}{2}\sum_{(f^s, f^T)\in\mathcal{G}} \gamma_{sT}\|f^s - f^T\|_{\mathcal{H}}^2, \quad (4)$$

where γ_{sT} defines the relevance between the s-th task and the T-th task of a graph $\mathcal{G}$ and the graph $\mathcal{G}$ represents the weighted connectivity of tasks.

The regularizers defined in our DAM and MTL are different in the following aspects: 1) MTL simultaneously learns two functions f^s and f^T (see (4)) and the two functions are compared in the same RKHS $\mathcal{H}$. In contrast, the auxiliary classifiers f^s's in (3) are assumed to be pre-computed, 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 auxiliary classifiers and the target classifier in DAM; 2) It is still unclear how to exploit the unlabeled samples through the regularizer (4) of MTL. In contrast, the unlabeled patterns D_u^T from the target domain are used in DAM (See Figure 1 and (3)).

2.3. Proposed Formulation

Based on the smoothness assumption for domain adaptation, we propose to minimize simultaneously the structural risk functional of Regularized Least-Squares (RLS) (*a.k.a.* Least-Squares SVM[3] (LS-SVM) (Van Gestel et al., 2004)), as well as the data-dependent regularizer defined in Section 2.2. The proposed method, namely Domain Adaptation Machine (DAM), is then formulated as follows:

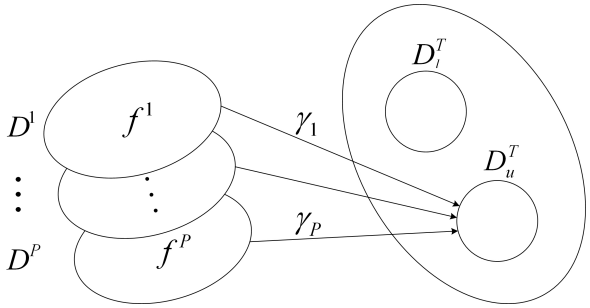

Figure 1. Label propagation from multiple auxiliary classifiers $f^1, f^2, \ldots, f^P$ to the unlabeled patterns in the target domain.

$$\min_{f^T} \Omega(f^T) + \frac{1}{2}\sum_{i=1}^{n_l}(f_i^T - y_i^T)^2 + \Omega_D(\boldsymbol{f}_u^T), \quad (5)$$

where $\Omega(f^T)$ is a regularizer to control the complexity of the target classifier f^T, the second term is the empirical error of the target classifier f^T on the target labeled patterns D_l^T, and the last term is our data-dependent regularizer.

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 $\Omega(f^T) = \frac{1}{2\theta}\|\mathbf{w}\|^2$, where $\theta > 0$ is a regularization parameter. Then, the solution f^T of the optimization problem (5) is*

$$f^T(\mathbf{x}) = \theta\sum_s \gamma_s \sum_{i=1}^{n_T} f^s(\mathbf{x}_i^T)\tilde{k}(\mathbf{x}_i^T, \mathbf{x}) + \sum_{i=1}^{n_l}\alpha_i^T\tilde{k}(\mathbf{x}_i^T, \mathbf{x}), \quad (6)$$

where

$$\tilde{k}(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) - \mathbf{k}'_{\mathbf{x}_i}(\mathbf{I} + \mathbf{MK})^{-1}\mathbf{M}\mathbf{k}_{\mathbf{x}_j} \quad (7)$$

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 $\phi(\mathbf{x}_j)$, $\mathbf{k}_{\mathbf{x}} = [k(\mathbf{x}_1^T, \mathbf{x}), \ldots, k(\mathbf{x}_{n_T}^T, \mathbf{x})]'$, $\mathbf{K} = [k(\mathbf{x}_i^T, \mathbf{x}_j^T)] \in \Re^{n_T\times n_T}$ is the kernel matrix defined on both labeled and unlabeled data in the target domain, $\mathbf{M} = \theta\sum_s \gamma_s\mathbf{I}$, and α_i^T is the coefficient for the i-th labeled patterns in the target domain.

Proof. Due to space limit, the proof is omitted here. □

Note, similar to (Evgeniou et al., 2005), the solution of the target decision function f^T is non-sparse. All the auxiliary classifiers f^s need to be used for predicting labels of the target patterns, making it inefficient for large-scale applications (*e.g.*, video concept detection). Moreover, similar to the manifold kernel defined in (Sindhwani et al., 2005), the kernel for domain adaptation (7) involves the matrix inversion of a matrix $(\mathbf{I} + \mathbf{MK})$, which is computationally infeasible when n_u is large.

[3]Experimental results show that LS-SVM is comparable with SVM (Van Gestel et al., 2004).

2.3.1. SPARSE SOLUTION

Recall that the use of the ϵ-insensitive loss function in Support Vector Regression (SVR) can usually lead to a sparse representation of the decision function. Therefore, to obtain the sparse solution, we introduce an additional term in (5), which regulates the approximation quality and the sparsity of the decision function. Moreover, we also assume that the regularizer $\Omega(f^T) = \frac{1}{2\theta}\|\mathbf{w}\|^2$ for the penalty of function complexity of f^T. The optimization problem (5) is then rewritten as:

$$\min_{f^T,\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n_T}\ell_\epsilon(\mathbf{w}'\phi(\mathbf{x}_i)+b-f_i^T) + \theta\left(\frac{1}{2}\|\boldsymbol{f}_l^T-\mathbf{y}_l\|^2+\Omega_D(\boldsymbol{f}_u^T)\right), \quad (8)$$

where $\boldsymbol{f}_l^T = [f_1^T,\ldots,f_{n_l}^T]'$ is the vector of the target decision function on the labeled patterns D_l^T from the target domain, $\mathbf{y}_l = [y_1,\ldots,y_{n_l}]'$ is the vector of true labels in the target domain, θ is a tradeoff parameter to control the empirical error of the label patterns from the target domain as well as the smoothness regularizer, C is another tradeoff parameter to control the difference between $f^T(\mathbf{x})$ and $\mathbf{w}'\phi(\mathbf{x})+b$, 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 non-smooth, (8) is usually transformed as a constrained optimization problem, that is:

$$\begin{aligned}\min_{\boldsymbol{f}^T,\mathbf{w},b,\xi_i,\xi_i^*} \ & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n_T}(\xi_i+\xi_i^*) \\ & +\frac{1}{2}\theta\left(\|\boldsymbol{f}_l^T-\mathbf{y}_l\|^2+\sum_s\gamma_s\|\boldsymbol{f}_u^T-\boldsymbol{f}_u^s\|^2\right) \\ \text{s.t.} \quad & \mathbf{w}'\phi(x_i)+b-f_i^T \le \epsilon+\xi_i, \ \ \xi_i\ge 0, \quad (9) \\ & f_i^T-\mathbf{w}'\phi(x_i)-b \le \epsilon+\xi_i^*, \ \ \xi_i^*\ge 0, \quad (10)\end{aligned}$$

where ξ_i's and ξ_i^*'s are slack variables for ϵ-insensitive loss.

2.3.2. DETAILED DERIVATION

By introducing the Lagrange multipliers α_i's and η_i's (resp. α_i^*'s and η_i^*'s) for the constraints of (9) (resp. (10)), we obtain the following Lagrangian:

$$\begin{aligned}L = & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n_T}(\xi_i+\xi_i^*) \\ & +\frac{1}{2}\theta\left(\|\boldsymbol{f}_l^T-\mathbf{y}_l\|^2+\sum_s\gamma_s\|\boldsymbol{f}_u^T-\boldsymbol{f}_u^s\|^2\right) \\ & -\sum_{i=1}^{n_T}\alpha_i(\epsilon+\xi_i+f_i^T-\mathbf{w}'\phi(\mathbf{x}_i)-b)-\sum_{i=1}^{n_T}\xi_i'\eta_i \\ & -\sum_{i=1}^{n_T}\alpha_i^*(\epsilon+\xi_i^*-f_i^T+\mathbf{w}'\phi(\mathbf{x}_i)+b)-\sum_{i=1}^{n_T}\xi_i^{*\prime}\eta_i^* \quad (11)\end{aligned}$$

Setting the derivative of (11) w.r.t. the primal variables ($\boldsymbol{f}^T, \mathbf{w}, b, \xi_i$ and ξ_i^*) to zeros, we have:

$$\boldsymbol{f}^T = \begin{bmatrix}\mathbf{y}_l \\ \sum_s\tilde{\gamma}_s\boldsymbol{f}^s\end{bmatrix} + \frac{\boldsymbol{\alpha}-\boldsymbol{\alpha}^*}{\theta}, \quad (12)$$

and $\mathbf{w} = \Phi(\boldsymbol{\alpha}^*-\boldsymbol{\alpha})$, $\boldsymbol{\alpha}'\mathbf{1} = \boldsymbol{\alpha}^{*\prime}\mathbf{1}$, $C\mathbf{1}\ge\boldsymbol{\alpha},\boldsymbol{\alpha}^*\ge\mathbf{0}$, where $\tilde{\gamma}_s = \frac{\gamma_s}{\sum_s\gamma_s}$ is the normalized weight for the s-th auxiliary 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 $\Phi = [\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_{n_T})]$. Substituting them back into (11), we arrive at the following dual formulation:

$$\begin{aligned}\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} & \frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)'\tilde{\mathbf{K}}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)+\tilde{\mathbf{y}}'(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)+\epsilon\mathbf{1}'(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*) \\ \text{s.t.} \quad & \boldsymbol{\alpha}'\mathbf{1}=\boldsymbol{\alpha}^{*\prime}\mathbf{1}, \ \ \mathbf{0}\le\boldsymbol{\alpha},\boldsymbol{\alpha}^*\le C\mathbf{1}, \quad (13)\end{aligned}$$

where

$$\tilde{\mathbf{K}} = \mathbf{K}+\frac{1}{\theta}\begin{bmatrix}I & \mathbf{0} \\ \mathbf{0} & \frac{1}{p}I\end{bmatrix} \quad (14)$$

is a transformed kernel matrix, and $p = \sum_s\gamma_s$, and

$$\tilde{\mathbf{y}} = \begin{bmatrix}\mathbf{y}_l \\ \sum_s\tilde{\gamma}_s\boldsymbol{f}^s\end{bmatrix}. \quad (15)$$

2.3.3. PARAMETRIC PREDICTION

From the Karush Kuhn Tucker (KKT) condition in (12), we can obtain the vector of the target decision values $\boldsymbol{f}^T$. Moreover, the decision value of unlabeled data D_u^T in the target domain is given as: $f^T(\mathbf{x}_i) = \sum_s\tilde{\gamma}_s f^s(\mathbf{x}_i) + \frac{\alpha_i-\alpha_i^*}{\theta}$, $\forall i = n_l+1,\ldots,n_T$, which is similar to that of A-SVM when we set the perturbation Δf in A-SVM for the unlabeled pattern $\mathbf{x}_i$ as $\Delta f(x_i) = \frac{\alpha_i-\alpha_i^*}{\theta}$. However, $f^T(\mathbf{x}_i)$ also involves the ensemble output from the auxiliary classifiers. Alternatively, we use the parametric form of the target decision function for label prediction on any test pattern by

$$f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x})+b = \sum_{\alpha_i-\alpha_i^*\neq 0}(\alpha_i^*-\alpha_i)k(\mathbf{x}_i,\mathbf{x})+b, \quad (16)$$

which is a linear combination of $k(\mathbf{x}_i,\mathbf{x})$'s only without involving any auxiliary classifiers. Here, $\mathbf{x}_i$ is the support vector from the target patterns with nonzero coefficient $\alpha_i^*-\alpha_i$, and the bias b can be obtained from the KKT conditions. According to the KKT conditions, if the patterns have the value $|\mathbf{w}'\phi(\mathbf{x}_i)+b-f_i^T|$ less than ϵ, then their corresponding coefficient in (16) becomes zero. Therefore, with the use of ϵ-insensitive loss function, the computation for the prediction using the sparse representation in (16) can be greatly reduced, when compared with that of A-SVM.

2.3.4. CONNECTION TO SVR

Surprisingly, the dual (13) does not involve any expensive matrix operation as in (Evgeniou et al., 2005; Kato et al.,

2008) and can be reduced to a form, which is similar to the dual of ϵ-SVR:

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

except that the kernel matrix $\mathbf{K}$ is replaced by the transformed kernel matrix $\tilde{\mathbf{K}}$, and the regression label vector $\mathbf{y}$ is replaced by $\tilde{\mathbf{y}}$ in (15). In experiments, we normalize the sum of γ_s to 1 (*i.e.*, $p = 1$). So the transformed kernel matrix is $\mathbf{K} + \mathbf{I}/\theta$, which is similar to Automatic Relevance Determination (ARD) kernel used in Gaussian Process, where θ is a parameter to control the noise of output. Note that the last n_u entries of $\tilde{\mathbf{y}}$ is similar to a virtual label $\tilde{y}_i = \sum_s \tilde{\gamma}_s f^s(\mathbf{x}_i)$ generated by the auxiliary classifiers f^s's on the unlabeled target data D_u^T (See Figure 1). Moreover, the sparsified DAM in (8) can be solved efficiently by using state-of-the-art SVM solvers such as LIBSVMfor large data sets, which takes $O(n_T^{2.3})$ training time and $O(n_T)$ memory storage only. When compared with the original formulation in (5), the calculation of the matrix inversion in (7) is avoided.

2.4. Discussions with Related Work

Our proposed DAM is different from MTL. DAM focuses on learning the target decision classifier only by leveraging the existing auxiliary classifiers, and the computational cost in learning stage is significantly reduced. In addition, according to the definition of our data-dependent regularizer in (3), the auxiliary classifiers can be trained with different kernels and even different learning methods.

The most related work to DAM is A-SVM (Yang et al., 2007), in which the new SVM classifier is adapted from the existing auxiliary classifiers. However, DAM is different from A-SVM in several aspects: 1) A-SVM did not exploit the unlabeled data D_u^T in the target domain. In contrast, the unlabeled patterns D_u^T in the target domain are employed in DAM (see the data-dependent regularizer defined in (3)); 2) A-SVM employed auxiliary classifiers for the label prediction of the patterns in the target domain. In contrast, the target classifier learned in DAM (See (16)) is in a sparse representation of the target patterns only. Therefore, as shown in the experiments, DAM is much faster than A-SVM in terms of the average total testing time.

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 assumption.

3. Experiments

We compare our proposed method DAM with the baseline SVM, Transductive SVM (T-SVM) (Joachims, 1999), and other four domain adaptation methods: Multiple Convex Combination of SVM (MCC-SVM) (Schweikert et al., 2009), Feature Replication (FR) (Daumé III, 2007), Adaptive SVM (A-SVM) (Yang et al., 2007) and multiple KMM (Multi-KMM) (Schweikert et al., 2009).

Table 1. Description of TRECVID 2005 data set

Domain	Channel	# keyframes
D^1	CNN_ENG	11,025
D^2	MSNBC_ENG	8,905
D^3	NBC_ENG	9,322
D^4	CCTV4_CHN	10,896
D^5	NTDTV_CHN	6,481
D^T	LBC_ARB	15,272

3.1. Experimental Setup

We use TRECVID 2005 dataset, which contains 61,901 keyframes extracted from 108 hours of video programmes 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 key-frames in six channels are listed in Table 1. We choose 36 semantic concepts from the LSCOM-lite lexicon (Naphade et al., 2005). The 36 concepts covers the dominant visual concepts present in broadcast news videos, and they have been manually annotated to describe the visual content of the keyframes in TRECVID 2005 data set.

As shown in (Yang et al., 2007), the data distributions of six channels are quite different, making it suitable for evaluating domain adaptation methods. In this work, 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 set comprises of all the labeled samples from the source domains as well as 360 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 data set.

Three low-level global features Grid Color Moment (225 dim.), Gabor Texture (48 dim.) and Edge Detection Histogram (73 dim.) are used to represent the diverse content of keyframes, because of their consistent, good performance reported in TRECVID (Yang et al., 2007). Then the three types of features are put together to form a 346-dimensional feature to represent each keyframe.

MCC-SVM, FR, A-SVM and Multi-KMM can cope with training samples from multiple source domains. For MCC-SVM, similarly as in (Schweikert et al., 2009), we equally fuse the decision values of six SVM classifiers independently trained with the labeled patterns from the target domain and five source domains. For the baseline SVM algorithm, we report the results for two cases: 1) in SVM_T, we only use the the training patterns from the target domain

(*i.e.*, D_l^T) for SVM learning; 2) in SVM_S, we equally fuse the decision values of five auxiliary classifiers independently trained with the labeled patterns from five source domains. Considering that DAM can take advantage of both labeled and unlabeled data, we use semi-supervised setting in this work. In practice, 4,000 test instances from the target domain are randomly sampled as D_u^T for DAM, which are used as unlabeled data during the learning process. For semi-supervised learning algorithm T-SVM, we also report two results: 1) in T-SVM_T, the labeled data from D_l^T as well as the 4,000 unlabeled patterns from D_u^T are employed for learning; 2) in T-SVM_ST, we equally fuse the decision values of six classifiers including the five auxiliary classifiers as well as the T-SVM_T classifier.

For all methods, we train one-versus-others SVM classifiers with the fixed regularization parameter $C = 1$. For DAM, we also fix the tradeoff parameter $\theta = 100$. Gaussian kernel (*i.e.*, $k(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$) is used as the default kernel in SVM_T, SVM_S, MCC-SVM, FR and Multi-KMM, where γ is set to $\frac{1}{d} = 0.0029$ ($d = 346$ is the feature dimension). For A-SVM, we train 50 auxiliary classifiers by independently using five sources and ten kernel parameters for Gaussian kernel, which are set as $1.2^\delta\gamma$, $\delta = \{-0.5, 0, 0.5, \ldots, 4\}$. We also report two results for DAM: 1) in DAM_50, we exploit the same 50 auxiliary classifiers as in A-SVM; 2) in DAM_200, we additionally employ another three types of kernels: Laplacian kernel (*i.e.*, $k(x_i, x_j) = \exp(-\sqrt{\gamma}\|x_i - x_j\|)$), inverse square distance kernel (*i.e.*, $k(x_i, x_j) = \frac{1}{\gamma\|x_i - x_j\|^2+1}$) and inverse distance kernel (*i.e.*, $k(x_i, x_j) = \frac{1}{\sqrt{\gamma}\|x_i - x_j\|+1}$). In total, there are 200 auxiliary classifiers from five sources, four types of kernels and ten kernel parameters.

In A-SVM and DAM, we need to determine γ_s in (3). For fair comparison, we set $\gamma_s = \frac{\exp(-\beta(\mathrm{Dist}_k(D^s, D^T))^\rho)}{\sum_s \exp(-\beta(\mathrm{Dist}_k(D^s, D^T))^\rho)} \in (0, 1)$, where $\beta = 1000$ and $\rho = 2$ are parameters 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)[4] for measuring the data distributions between the source and target domains.

3.2. Performance Comparisons

For performance evaluation, we use non-interpolated Average Precision (AP) (Smeaton et al., 2006), which has been used as the official performance metric in TRECVID since 2001. It corresponds to the multi-point average precision value of a precision-recall curve, and incorporates the effect of recall when AP is computed over the entire classification results.

[4]MMD proposed by Borgwardt et al. (2006) is a nonparametric distance metric for comparing data distributions in the RKHS, namely $\mathrm{Dist}_k(D^s, D^T) = \left\| \frac{1}{n_s}\sum_{i=1}^{n_s} \phi(\mathbf{x}_i^s) - \frac{1}{n_T}\sum_{i=1}^{n_T} \phi(\mathbf{x}_i^T) \right\|^2$.

The per-concept APs are shown in Figure 2 and the Mean Average Precision (MAP) over all 36 concepts is given in Table 2. From Table 2, 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 patterns from source domains and target domain can be used to improve generalization performance in the target domain. MCC-SVM, FR 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.

Our proposed method DAM outperforms all the other algorithms in terms of MAP, demonstrating that DAM learns a robust target classifier for domain adaptation by leveraging a set of pre-learnt auxiliary classifiers. In practice, DAM achieves the best results in 14 out of 36 concepts. When compared with SVM_T and MCC-SVM (the second best result), the relative MAP improvements of DAM_200 are 21.2% and 7.3%, respectively. When compared with A-SVM, the relative improvements of DAM_50 and DAM_200 are 6.0% and 10.0%, respectively. Moreover, the MAP performances of two single-source SVM models, which are trained based on labeled patterns from all the source domains and from all the source and target domains, are 23.4% and 28.4% respectively. They are inferior to SVM_S and MCC-SVM in multi-source setting.

Observed from Figure 2, SVM_T performs better than other domain adaptation methods for some concepts, such as "Weather", "Court", "Bus" and so on. For those concepts with few positive training samples, the data distributions of source and target domains based on the low-level features can be very different. Thus, the source domains cannot provide useful information to the target domain, and may lead to negative transfer.

Finally, we compare DAM with the semi-supervised learning algorithm T-SVM. The MAPs of T-SVM_T and T-SVM_ST over all 36 concepts are 22.6% and 25.1%, respectively, which is significantly worse than DAM. We also observe that T-SVM_T and T-SVM_ST are even worse than SVM_T in terms of MAP, possibly because T-SVM suffers from the local minima solution and the sampling selection bias of labeled patterns from the target domain.

3.3. Testing Time Comparisons

The testing time is crucial for large-scale applications with a large number of test patterns. All the experiments are performed on an IBM workstation (2.13 GHz CPU with 16 Gbyte RAM). In Table 3, we compare DAM with other methods in terms of the average support vectors (SVs) as well as the average total testing time on the test data set (about 15,000 samples) from Arabic channel over all 36 concepts. Because of the utilization of sparsity regulariz-

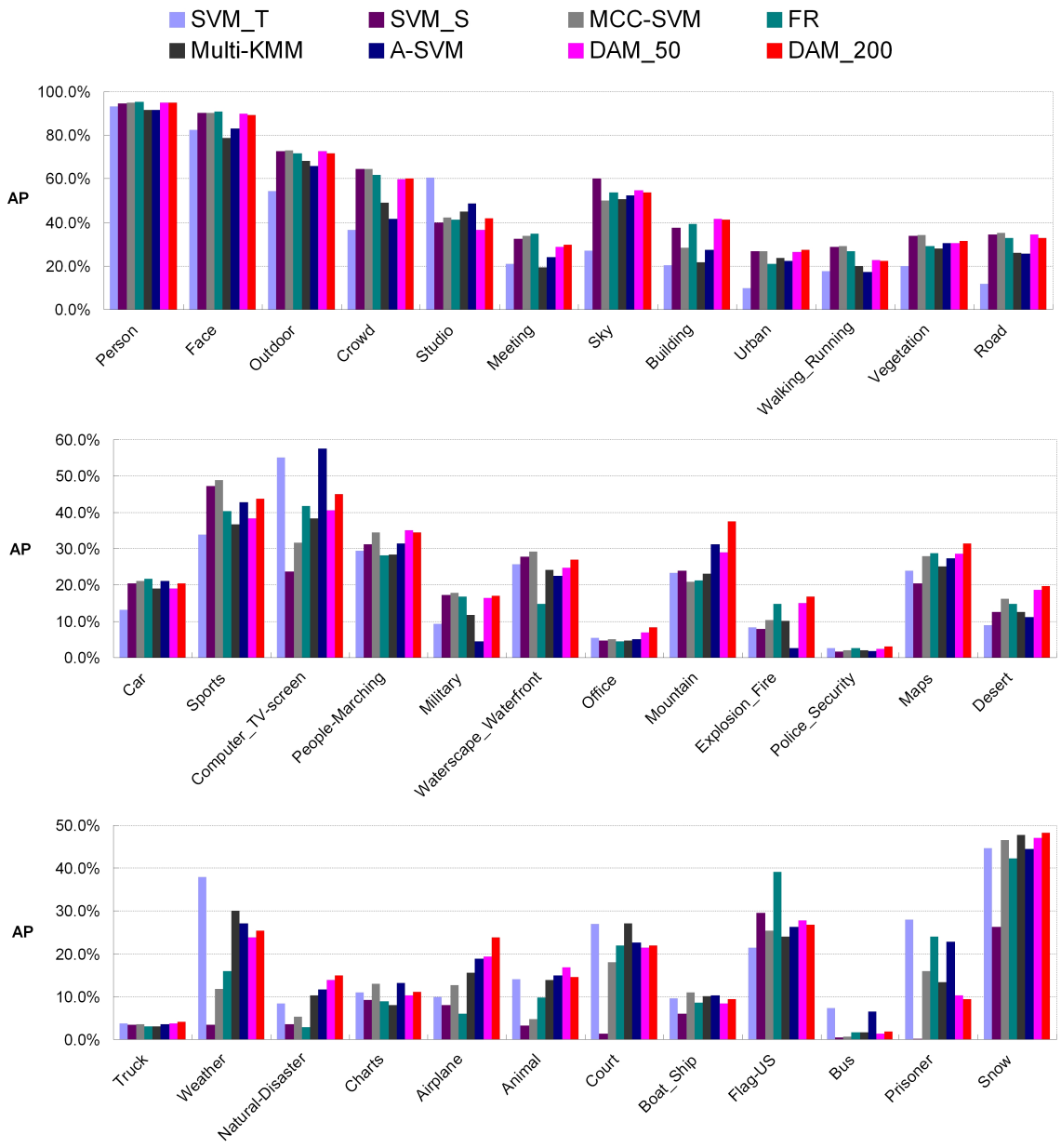

Figure 2. Performance Comparison of DAM with other methods in terms of Per-Concept Average Precision (AP).

Table 2. Performance comparisons of DAM with other methods in terms of Mean Average Precision (MAP) over all 36 concepts.

	SVM_T	SVM_S	MCC-SVM	FR	Multi-KMM	A-SVM	DAM_50	DAM_200
MAP	25.5%	26.4%	28.8%	28.7%	26.7%	28.1%	**29.8%**	**30.9%**

Table 3. Performance comparisons of DAM with other methods in terms of the average number of support vectors (SVs) and the average total testing time on the test data set (about 15,000 samples) from Arabic channel over all 36 concepts.

	SVM_T	SVM_S	MCC-SVM	FR	Multi-KMM	A-SVM	DAM_50	DAM_200
# SVs	95	7,029	7,124	7,238	2,167	86,104	561	380
Avg. Total Testing Time (s)	22	1,063	1,085	721	212	10,810	**52**	**35**

er, we observe that DAM has much fewer support vectors, when compared with SVM_S and other domain adaptation methods MCC-SVM, FR, Multi-KMM and A-SVM. DAM is as fast as SVM_T and it is much faster than SVM_S and other domain adaptation methods in terms of the average total testing time. We also observe that A-SVM is more than 300 times slower than DAM_200. Note A-SVM used all the 50 auxiliary models for label prediction with (2). In contrast, the prediction function in DAM is in a sparse representation of target patterns only.

4. Conclusion

We have proposed a new domain adaptation method, referred to as Domain Adaptation Machine (DAM), for the challenging video concept detection task. DAM learns a robust target classifier for predicting labels of the patterns from the target domain by leveraging a set of pre-learnt auxiliary classifiers based on the labeled samples from multiple source domains. We introduce smoothness assumption into the Least Squares SVM (LS-SVM) as a data-dependent regularizer such that the target classifier is enforced to share similar decision values with the auxiliary classifiers. We additionally exploit a sparsity regularizer to learn a sparse target classifier. The experiments on TRECVID 2005 data set demonstrate that DAM outperforms existing multiple source domain adaptation methods. DAM is also suitable for large-scale video concept detection task because of its efficiency for label prediction.

Acknowledgements

This material is based upon work funded by Singapore A*STAR SERC Grant (082 101 0018) and MOE AcRF Tier-1 Grant (RG15/08).

References

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, *12*, 2399–2434.

Blitzer, J., McDonald, R., & Pereira, F. (2006). Domain adaptation with structural correspondence learning. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 120–128.

Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., & Smola, A. J. (2006). Integrating structured biological data by kernel maximum mean discrepancy. *Proceedings of the 14th International Conference on Intelligent Systems for Molecular Biology*, 49–57.

Crammer, K., Kearns, M., & Wortman, J. (2008). Learning from multiple sources. *Journal of Machine Learning Research*, *9*, 1757–1774.

Daumé III, H. (2007). Frustratingly easy domain adaptation. *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, 256–263.

Evgeniou, T., Micchelli, C., & Pontil, M. (2005). Learnng multiple tasks with kernel methods. *Journal of Machine Learning Research*, *6*, 615–637.

Huang, J., Smola, A., Gretton, A., Borgwardt, K. M., & Schölkopf, B. (2007). Correcting sample selection bias by unlabeled data. *Advances in Neural Information Processing Systems 19*, 601–608.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proceedings of the Sixteenth International Conference on Machine Learning*, 200–209.

Kato, T., Kashima, H., Sugiyama, M., & Asai, K. (2008). Multi-task learning via conic programming. *Advances in Neural Information Processing Systems 20*, 737–744.

Luo, P., Zhuang, F., Xiong, H., Xiong, Y., & He, Q. (2008). Transfer learning from multiple source domains via consensus regularization. *Proceeding of the ACM 17th Conference on Information and Knowledge Management*, 103–112.

Mansour, Y., Mohri, M., & Rostamizadeh, A. (2009). Domain adaptation with multiple sources. *Advances in Neural Information Processing Systems 21*, 1041–1048.

Naphade, M. R., Kennedy, L., Kender, J. R., Chang, S.-F., Smith, J. R., Over, P., & Hauptmann, A. (2005). *A light scale concept ontology for multimedia understanding for TRECVID 2005* (Technical Report). IBM Research Technical Report.

Rosenstein, M. T., Marx, Z., & Kaelbling, L. P. (2005). To transfer or not to transfer. *NIPS 2005 Workshop on Inductive Transfer: 10 Years Later*.

Schweikert, G., Widmer, C., Schölkopf, B., & Rätsch, G. (2009). An empirical analysis of domain adaptation algorithm for genomic sequence analysis. *Advances in Neural Information Processing Systems 21*, 1433–1440.

Sindhwani, V., Niyogi, P., & Belkin, M. (2005). Beyond the point cloud: from transductive to semi-supervised learning. *Proceedings of the Twenty-second International Conference on Machine Learning*, 824–831.

Smeaton, A. F., Over, P., & Kraaij, W. (2006). Evaluation campaigns and trecvid. *ACM International Workshop on Multimedia Information Retrieval*, 321–330.

Storkey, A., & Sugiyama, M. (2007). Mixture regression for covariate shift. *Advances in Neural Information Processing Systems 19*, 1337–1344.

Van Gestel, T., Suykens, J., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., de Moor, B., & Vandewalle, J. (2004). Benchmarking least squares support vector machine classifiers. *Machine Learning*, *54*, 5–32.

Wu, P., & Dietterich, T. G. (2004). Improving SVM accuracy by training on auxiliary data sources. *Proceedings of the Twenty-first International Conference on Machine Learning*, 871–878.

Yang, J., Yan, R., & Hauptmann, A. G. (2007). Cross-domain video concept detection using adaptive SVMs. *Proceedings of the 15th International Conference on Multimedia*, 188–197.