Domain Transfer SVM for Video Concept Detection

Lixin Duan Ivor W. Tsang Dong Xu
School of Computer Engineering
Nanyang Technological University
{S080003, IvorTsang, DongXu}@ntu.edu.sg

Stephen J. Maybank
School of Computer Science and Information Systems
Birkbeck College, University of London
sjmaybank@dcs.bbk.ac.uk

Abstract

Cross-domain learning methods have shown promising results by leveraging labeled patterns from auxiliary domains to learn a robust classifier for target domain, which has a limited number of labeled samples. To cope with the tremendous change of feature distribution between different domains in video concept detection, we propose a new cross-domain kernel learning method. Our method, referred to as Domain Transfer SVM (DTSVM), simultaneously learns a kernel function and a robust SVM classifier by minimizing both the structural risk functional of SVM and the distribution mismatch of labeled and unlabeled samples between the auxiliary and target domains. Comprehensive experiments on the challenging TRECVID corpus demonstrate that DTSVM outperforms existing cross-domain learning and multiple kernel learning methods.

1. Introduction

Video concept detection is a primitive task in many computer vision applications such as content-based video search and indexing, human-computer interaction and so on. There is a growing interest in the challenging task of concept detection for videos from various sources including broadcast news videos [13, 18], consumer videos [4] and web videos [21], etc. When both the training and the test data come from the same domain (*e.g.*, web videos) and sufficient labeled training samples are provided, prior methods such as [4, 13, 18, 21] have demonstrated promising results.

However, the collection of labeled training data requires expensive and time-consuming human labor. Classifiers trained with only a limited number of labeled patterns are usually not robust for video concept detection. To this end, cross-domain learning (or domain adaptation) methodswere recently proposed [1, 5, 17] to learn robust classifiers with only a limited number of labeled patterns from the target domain by leveraging a large amount of labeled training data from other domains (referred to as auxiliary/source domains). In practice, cross-domain learning methods have been successfully used in many real-world applications, such as video concept detection, sentiment classification, natural learning processing [20, 7, 1, 5].

Recall that feature distributions of training samples from different domains (*e.g.*, from broadcast news domain to web videos) change tremendously, and the training samples from multiple sources also have very different statistical properties (such as mean, intra-class and inter-class variance). Although large amounts of the training data are available in the auxiliary domains, the classifiers trained from this data or the combined data of both the auxiliary and the target domains may perform poorly on the test data of the target domain [20, 7].

To take advantage of all patterns from both the auxiliary and the target domains, Daumé III [5] proposed a Feature Replication (FR) method to augment features for cross-domain learning. The augmented features are then used to construct a kernel function for Support Vector Machine (SVM) training. Yang *et al.* [20] proposed Adaptive SVM (A-SVM) to enhance the prediction performance of video concept detection , in which the new SVM classifier $f_T(\mathbf{x})$ is adapted from an existing classifier $f_A(\mathbf{x})$ trained from the auxiliary domain. Following this work, cross-domain SVM (CD-SVM) proposed by Jiang *et al.* [7] used k-nearest neighbors from the target domain to define a weight for each auxiliary pattern, and then the SVM classifier was trained with re-weighted patterns. However, all these methods [5, 7, 17, 20] did not utilize unlabeled patterns in the target domain. Such patterns can also be used to improve the classification performance [22].

When there are only a few or even no labeled patterns in the target domain (*i.e.*, an extreme case), the classifier can be trained with the auxiliary patterns. Several cross-domain learning methods [6, 15] were also proposed to cope with inconsistency of data distribution (such as covariate shift [15] or sampling selection bias [6]). These methods re-weight the training patterns from the auxiliary domain by using unlabeled data from the target domain such that the statistics of samples from both domains are matched.

Note that the kernel function plays a crucial role in ker-

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

nel methods (*e.g.*, SVM) [12]. Typically, the kernel function needs to be chosen before learning. The associated kernel parameters (such as bandwidth parameter in the Gaussian kernel) can then be determined by optimizing generalization error bounds. Various kernel learning methods [8, 11, 14] have been proposed to directly learn the kernel function. However, these methods commonly assume that both the training data and the test data are drawn from the same domain.

In this paper, we propose a new cross-domain kernel learning method, referred to as Domain Transfer SVM (DTSVM), for the challenging video concept detection task. To deal with the tremendous change of keyframe feature distributions from different domains, DTSVM minimizes the structural risk functional of SVM and Maximum Mean Discrepancy (MMD) [2], a criterion to evaluate the distribution difference of labeled and unlabeled samples between the auxiliary and target domains. In practice, DTSVM provides a unified framework to simultaneously learn an optimal kernel function and a robust SVM classifier. To simplify kernel learning and facilitate the usage of the existing SVM software (*e.g.*, LIBSVM), we assume that the kernel function in SVM learning is from a linear combination of multiple base kernels. Moreover, we propose an efficient learning algorithm to solve the linear combination coefficients of kernels and the SVM classifier under a unified convex optimization framework. We also develop a simple predicting criterion to effectively determine how to choose cross-domain learning methods for different concepts.

The main contributions of this paper include:
1) To the best of our knowledge, DTSVM is the first semi-supervised cross-domain kernel learning method. In contrast to the prior kernel learning methods, DTSVM does not assume that the training and test data are drawn from the same domain.
2) DTSVM outperforms the state-of-the-art cross-domain learning methods in the challenging TRECVID dataset, demonstrating promising performance in real applications.
3) By learning a robust classifier with labeled patterns from both the auxiliary and the target domains, DTSVM can be successfully used for video concept detection from different video sources, especially for the target domain with only a limited number of labeled patterns, which saves a large amount of human labor.

2. Brief Review of Related Work

In this work, the transpose of vector / matrix is denoted by the superscript $'$ and the trace of a matrix $\mathbf{A}$ is represented as $\text{tr}(\mathbf{A})$. 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 $\mathbf{u} = [u_1, \ldots, u_j]' \geq \mathbf{0}$ means that $u_i \geq 0$ for $i = 1, \ldots, j$. Moreover, the element-wise product between matrices $\mathbf{A}$ and $\mathbf{B}$ is represented as $\mathbf{A} \circ \mathbf{B} = [A_{ij} B_{ij}]$. $\mathbf{A} \succ \mathbf{0}$ means that the matrix $\mathbf{A}$ is symmetric and positive definite(pd), and $\mathbf{A} \succeq \mathbf{0}$ means $\mathbf{A}$ is symmetric and positive semidefinite (psd).

Denote the data set of 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 $D^A = (\mathbf{x}_i^A, y_i^A)|_{i=1}^{n_A}$ as the data set from the auxiliary domain[1]. Let us represent the labeled training data set as $D = (\mathbf{x}_i, y_i)|_{i=1}^{n}$, which can be from the target domain (*i.e.*, $D = D_l^T$) or from the both domains (*i.e.*, $D = D_l^T \cup D^A$).

In cross-domain learning, it is crucial to reduce the difference of data distribution between the auxiliary and target domains. Many parametric criteria (e.g. Kullback-Leibler (KL) divergence) have been used to measure the distance between data distributions. However, an intermediate density estimate is usually required. To avoid such a non-trivial task, Borgwardt *et al.* [2] proposed an effective nonparametric criterion, referred to as Maximum Mean Discrepancy (MMD), to compare data distributions based on the distance between the means of samples from the two domains in the Reproducing Kernel Hilbert Space (RKHS), namely:

$$\text{dist}_k(D^A, D^T) = \left\| \frac{1}{n_A} \sum_{i=1}^{n_A} \varphi(\mathbf{x}_i^A) - \frac{1}{n_T} \sum_{i=1}^{n_T} \varphi(\mathbf{x}_i^T) \right\|^2. \quad (1)$$

To capture higher order statistics of the data (*e.g.*, higher order moments of probability distribution), the samples in (1) are transformed into a higher dimensional or even infinite dimensional space through a kernel function k induced from the nonlinear feature mapping $\varphi(\cdot)$. We therefore refer the distance in MMD as $\text{dist}_k(D^A, D^T)$. Note that the dot product of $\varphi(\mathbf{x}_i)$ and $\varphi(\mathbf{x}_j)$ equals to a kernel function k (or $k(\cdot,\cdot)$) on $\mathbf{x}_i$ and $\mathbf{x}_j$, namely, $k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)'\varphi(\mathbf{x}_j)$.

Due to the change of the data distribution from different domains, training with samples from the auxiliary domain may degrade the classification performance in the another target domain. To reduce the mismatch between the two different domains, Huang *et al.* [6] proposed a two-step approach Kernel Mean Matching (KMM). The first step is to diminish the difference of means of samples in RKHS between the two domains by re-weighting the samples $\varphi(\mathbf{x}_i)$ in the auxiliary domain as $\beta_i \varphi(\mathbf{x}_i)$, where β_i is learned by using the MMD criterion in (1). Then the second step is to learn a decision function $f(\mathbf{x}) = \mathbf{w}'\varphi(\mathbf{x}) + b$ that separates patterns of opposite classes in D using the loss function re-weighted by β_i. Recently, Pan *et al.* [10] proposed an unsupervised kernel matrix learning method by minimizing the MMD criterion in (1) as well, then apply the learned kernel matrix to train a SVM classifier for WiFi location and text categorization.

[1] Note our work can also use the unlabeled data from auxiliary domain.

3. Domain Transfer Support Vector Machine

3.1. Proposed Formulation

In previous cross-domain learning methods [6, 10], the weights or the kernel matrix of samples are learned separately using the MMD criterion in (1) without considering any label information. However, it is usually beneficial to utilize label information during kernel learning. Instead of using two-step approaches as in [6, 10], we propose a unified cross-domain learning framework DTSVM to learn the SVM decision function $f(\mathbf{x}) = \mathbf{w}'\varphi(\mathbf{x}) + b$ as well as the kernel function k simultaneously. In practice, DTSVM minimizes the distance of data distribution between the two domains, as well as the structural risk functional of SVM. The optimization problem of DTSVM is then formulated as:

$$[k, f] = \arg\min \Omega(\text{dist}_k(D^A, D^T)) + \theta\ \text{SVM}_{k,f}(D), \quad (2)$$

where $\Omega(\cdot)$ is any monotonic increasing function, and $\theta > 0$ is a tradeoff parameter to balance the difference of data distribution from two domains and the structural risk functional $\text{SVM}_{k,f}(D)$ of SVM for labeled patterns. Note, the kernel function k and the SVM decision function f can be learned at the same time.

First Criterion: The first objective in DTSVM is to minimize the mismatch of data distribution between the two domains using the MMD criterion defined in (1). We define a column vector $\mathbf{s}$ with $n_A + n_T$ entries, in which the first n_A entries are set as $1/n_A$ and the remaining entries are set as $-1/n_T$ respectively. Let $\Phi = [\varphi(\mathbf{x}_1^A), \ldots, \varphi(\mathbf{x}_{n_A}^A), \varphi(\mathbf{x}_1^T), \ldots, \varphi(\mathbf{x}_{n_T}^T)]$ be the data matrix after feature mapping, then $\frac{1}{n_A}\sum_{i=1}^{n_A} \varphi(\mathbf{x}_i^A) - \frac{1}{n_T}\sum_{i=1}^{n_T} \varphi(\mathbf{x}_i^T)$ in (1) is simplified as $\Phi\mathbf{s}$. Thus, the criterion in (1) can be rewritten as:

$$\text{dist}_k(D^A, D^T) = \|\Phi\mathbf{s}\|^2 = \text{tr}(\Phi'\Phi\mathbf{S}) = \text{tr}(\mathbf{KS}), \quad (3)$$

where $\mathbf{S} = \mathbf{ss}' \in \Re^{(n_A+n_T)\times(n_A+n_T)}$, and $\mathbf{K} = \Phi'\Phi = \begin{bmatrix} \mathbf{K}^{A,A} & \mathbf{K}^{A,T} \\ \mathbf{K}^{T,A} & \mathbf{K}^{T,T} \end{bmatrix} \in \Re^{(n_A+n_T)\times(n_A+n_T)}$, in which $\mathbf{K}^{A,A} \in \Re^{n_A\times n_A}$, $\mathbf{K}^{T,T} \in \Re^{n_T\times n_T}$ and $\mathbf{K}^{A,T} \in \Re^{n_A\times n_T}$ are the kernel matrices defined for the auxiliary domain, the target domain and the cross-domain from the auxiliary domain to the target domain respectively.

Second Criterion: The second objective in DTSVM is to minimize the structural risk functional $\text{SVM}_{k,f}(D)$ of SVM for better classification performance in the target domain. Let $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]'$ be a vector of the dual variables α_i of each labeled pattern, $\mathbf{y} = [y_1, \ldots, y_n]'$ as the label vector, $\mathbf{K}^{L,L} = [k(\mathbf{x}_i, \mathbf{x}_j)] \in \Re^{n\times n}$ is the kernel matrix of the labeled patterns, and $k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)'\varphi(\mathbf{x}_j)$. SVM is usually solved by its dual problem:

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} \boldsymbol{\alpha}'\mathbf{1} - \tfrac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\mathbf{K}^{L,L}(\boldsymbol{\alpha}\circ\mathbf{y}), \quad (4)$$

which is in form of the QP problem. Here, $\mathcal{A} = \{\boldsymbol{\alpha} \in \Re^n | C\mathbf{1} \geq \boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\alpha}'\mathbf{y} = 0\}$ is the feasible set of $\boldsymbol{\alpha}$.

Final Formulation: Substituting (3) and (4) into (2), we have the following saddle-point minimax problem:

$$\min_{\mathbf{K}\succeq\mathbf{0}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \Omega\left(\text{tr}(\mathbf{KS})\right) + \theta\left(\boldsymbol{\alpha}'\mathbf{1} - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\mathbf{K}^{L,L}(\boldsymbol{\alpha}\circ\mathbf{y})\right). \quad (5)$$

By utilizing both criteria, the samples from the auxiliary and target domain can be used to improve the classification performance of SVM classifier in the target domain. Moreover, an effective kernel function can be learned for a better representation of data in different domains.

Similar to the kernel learning method proposed in [8], the nonparametric kernel matrix $\mathbf{K}$ and the dual variables $\boldsymbol{\alpha}$ of the optimization problem in (5) can be learned by solving a semi-definite programming (SDP) problem [3] with a constraint $\mathbf{K}\succeq \mathbf{0}$. However, it is computationally prohibitive to solve a SDP problem when the data size is large.

3.2. Multiple Kernel Learning for DTSVM

Instead of learning a nonparametric kernel matrix $\mathbf{K}$, following [8, 11, 14], we can assume the kernel function k is a linear combination of a set of base kernel functions k_m, *i.e.*, $k = \sum_{m=1}^M d_m k_m$, where $d_m \geq 0$, $\sum_{m=1}^M d_m = 1$. We further assume that

$$\Omega\left(\text{tr}(\mathbf{KS})\right) = \frac{1}{2}\left(\text{tr}(\mathbf{KS})\right)^2. \quad (6)$$

This quadratic term is strictly convex, and so the second-order derivatives can be used to achieve faster convergence in kernel learning. Let us define two kernel matrices $\mathbf{K} = \sum_{m=1}^M d_m\mathbf{K}_m$, $\mathbf{K}^{L,L} = \sum_{m=1}^M d_m\mathbf{K}_m^{L,L}$, where $\mathbf{K}_m \in \Re^{(n_A+n_T)\times(n_A+n_T)}$ and $\mathbf{K}_m^{L,L} \in \Re^{n\times n}$ are the mth base kernel matrices defined for both domains and for the labeled patterns, respectively. Note the base kernel matrices $\mathbf{K}_m$ (resp. $\mathbf{K}_m^{L,L}$) are psd, so $\mathbf{K}$ (resp. $\mathbf{K}^{L,L}$) is still a psd kernel matrix. We then simplify (5) as:

$$\min_{\mathbf{d}\in\mathcal{M}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \frac{1}{2}\left(\text{tr}\left(\sum_{m=1}^M d_m\mathbf{K}_m\mathbf{S}\right)\right)^2 + \theta\left(\boldsymbol{\alpha}'\mathbf{1} - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\left(\sum_{m=1}^M d_m\mathbf{K}_m^{L,L}\right)(\boldsymbol{\alpha}\circ\mathbf{y})\right), \quad (7)$$

where $\mathbf{d} = [d_1, \ldots, d_m]'$, and a simplex $\mathcal{M} = \{\mathbf{d} \in \Re^M | \mathbf{d} \geq \mathbf{0},\ \mathbf{d}'\mathbf{1} = 1\}$ is the feasible set of $\mathbf{d}$.

However, (7) is a saddle-point minimax problem, and standard iterative update procedures may not converge. Let us define $p_m = \text{tr}\left(\mathbf{K}_m\mathbf{S}\right)$ and $\mathbf{p} = [p_1, \ldots, p_M]'$, then we have $\text{tr}\left(\sum_{m=1}^M d_m\mathbf{K}_m\mathbf{S}\right) = \mathbf{d}'\mathbf{p}$. Using the following Proposition 1, (7) can be transformed as:

$$\min_{\mathbf{d}\in\mathcal{M}} h(\mathbf{d}) = \min_{\mathbf{d}\in\mathcal{M}} \tfrac{1}{2}\mathbf{d}'\mathbf{pp}'\mathbf{d} + \theta J(\mathbf{d}), \quad (8)$$

where

$$J(\mathbf{d})=\min_{\mathbf{w},b}\frac{1}{2}\sum_{m=1}^{M}\frac{\|\mathbf{w}_m\|^2}{d_m}+C\sum_{i=1}^{n}\ell_h\left(y_i\left(\sum_{m=1}^{M}\mathbf{w}'\varphi_m(\mathbf{x}_i)+b\right)\right), \quad (9)$$

$\ell_h(f) = \max(0, 1 - f)$ is the hinge loss, and φ_m is the feature mapping function induced from the base kernel k_m.

Proposition 1. *The optimization problem in (9) is the same as the optimization problem:*

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} \boldsymbol{\alpha}'\mathbf{1}-\frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\left(\sum_{m=1}^{M}d_m\mathbf{K}_m^{L,L}\right)(\boldsymbol{\alpha}\circ\mathbf{y}). \quad (10)$$

For the derivation of this proposition, please refer to [11].

Theorem 1. *The optimization problem in (8) is jointly convex with respect to* $\mathbf{d}$*,* $\mathbf{w}_m$ *and* b*.*

Due to the space limitation, we omit the proof of Theorem 1. With Theorem 1, we can apply the alternative coordinate descent procedure proposed in [11] to update different variables ($\boldsymbol{\alpha}$ and $\mathbf{d}$) in (8) iteratively to obtain the globally optimal solution.

3.2.1 Detailed Algorithm

Update SVM parameters $\boldsymbol{\alpha}$: With a fixed $\mathbf{d}$, using Proposition 1, $J(\mathbf{d})$ in (9) can be solved by the dual of SVM using the kernel matrix $\sum_{m=1}^{M} d_m\mathbf{K}_m^{L,L}$ in (10) and standard SVM solvers, such as LIBSVM.

Update Kernel Parameters $\mathbf{d}$: When the parameters of the SVM decision function are fixed, using Proposition 1, $J(\mathbf{d})$ is linear with respect to $\mathbf{d}$ (see (10)), and (8) can be updated using second-order information and the reduced gradient method as suggested in [11]. Note that $\mathbf{pp}'$ is not full rank, to avoid numerical instability, we replace $\mathbf{pp}'$ by $\mathbf{pp}' + \epsilon\mathbf{I}$ where ϵ is set to 10^{-4} in the experiments. Then, the gradient of h in (8) is $\nabla h = (\mathbf{pp}' + \epsilon\mathbf{I})\mathbf{d} + \theta\nabla J$ where ∇J is the gradient of J in (9) (or (10)). As there is a quadratic term in (8), the hessian $\nabla^2 h = (\mathbf{pp}' + \epsilon\mathbf{I}) \succ \mathbf{0}$ is well-defined. Compared with first-order gradient methods, second-order derivative based methods usually converge faster. So we use $\mathbf{g} = (\nabla^2 h)^{-1}\nabla h = \mathbf{d} + \theta(\mathbf{pp}' + \epsilon\mathbf{I})^{-1}\nabla J$ as the update direction. To maintain $\mathbf{d} \in \mathcal{M}$, the update direction $\mathbf{g}$ is reduced as in [11], so the updated weight $\mathbf{d}$ is:

$$\mathbf{d}_{t+1} = \mathbf{d}_t - \eta_t\mathbf{g}_t \in \mathcal{M}, \quad (11)$$

where $\mathbf{d}_t$ and $\mathbf{g}_t$ are the weight vector $\mathbf{d}$ and the reduced update direction $\mathbf{g}$ at the tth iteration respectively, and η_t is the learning rate. The overall procedure of the proposed DTSVM is shown in Algorithm 1.

Algorithm 1 DTSVM Algorithm.

```
Initialize $\mathbf{d} = \frac{1}{M}\mathbf{1}$.
For $t = 1, \ldots, T_{max}$
    Solve $\boldsymbol{\alpha}$ of SVM objective in (9).
    Update $\mathbf{d}$ of multiple base kernels in (8) using (11).
End.
```

3.3. Discussions of Related Work

Our work is different from the prior cross-domain learning methods such as [5, 6, 7, 17, 20]. These methods use standard kernel functions for SVM training, in which the kernel parameters are usually determined through cross-validation. Recall that the kernel function plays a crucial role in SVM. When the labeled data from the target domain is limited, the cross-validation approach may not choose an optimal kernel. This degrades the generalization performance of SVM.

The work most closely related to DTSVM was proposed by Pan *et al.* [10], in which a two-step approach is used for cross-domain learning. The first step is to learn a kernel matrix of samples using the MMD criterion, and the second step is to apply the learned kernel matrix to train a SVM classifier. DTSVM is different from [10] in the following aspects: 1) A kernel matrix is learned in an unsupervised setting in [10] without using any label information, which is not as effective as our semi-supervised learning method DTSVM. 2) In contrast to the two-step approach in [10], DTSVM simultaneously learns a kernel function and SVM classifier. 3) The learned kernel matrix in [10] is nonparametric, thus it cannot be applied to unseen data. Instead, DTSVM can handle any new test data.

Multiple Kernel Learning (MKL) methods [8, 14, 11] also simultaneously learn the decision function and the kernel in an inductive setting. However, the default assumption of MKL is that the test data and the training data are drawn from the same domain.

4. Experiments

In this section, we compare our proposed method DTSVM with the baseline SVM, Transductive SVM (T-SVM) [22] and other cross-domain learning algorithms: FR [5], A-SVM [20], CD-SVM [7] and KMM [6]. We also report the results of the Multiple Kernel Learning (MKL) algorithm, in which the optimal kernel combination is obtained by minimizing the second term in (7) corresponding to the structural risk functional of SVM.

4.1. Description of Data Sets

The TRECVID video corpus is one of the largest annotated video benchmark data sets for research purposes. The TRECVID 2005 dataset contains 61,901 keyframes extracted from 108 hours of video programs from six broadcast sources (English, Arabic and Chinese), and the TRECVID 2007 dataset contains 21,532 keyframes extracted from 60 hours of news magazine, science news, documentaries and educational programming videos. 36

semantic concepts are chosen from the LSCOM-lite lexicon [9], a preliminary version of LSCOM, which covers 36 dominant visual concepts present in broadcast news videos, including objects, scenes, locations, people, events and programs. The 36 concepts have been manually annotated to describe the visual content of the key-frames in both TRECVID 2005 and 2007 data sets.

4.2. Experimental Setup

As shown in [7], TRECVID data sets are challenging for cross-domain learning methods because the TRECVID 2007 data set is quite different from the TRECVID 2005 data set in terms of program structure and production values. In this work, the auxiliary data set D^A is obtained by randomly sampling 100 positive samples per concept from the TRECVID 2005 data set, and 10 positive samples per concept from the TRECVID 2007 data set are randomly selected as the labeled training data set D_l^T of the target data set. The remaining samples in TRECVID 2007 data set are used as the test set.

FR, A-SVM, CD-SVM and KMM are learned from a combined training data set, which consists of the auxiliary data set D^A and the labeled training data set D_l^T from the target domain. For SVM, MKL and DTSVM, the labeled data D can be D_l^T or the whole labeled data from both domains (*i.e.*, $D = D_l^T \cup D^A$). We therefore refer to SVM (resp. MKL, DTSVM) of the above two cases as *SVM_T* (resp. *MKL_T*, *DTSVM_T*) and *SVM_AT* (resp. *MKL_AT*, *DTSVM_AT*) respectively. Considering that KMM and DTSVM can take advantage of both labeled and unlabeled data to measure the mismatch of data distribution between two domains using the MMD criterion, we use semi-supervised setting in this work. In practice, 4,000 unlabeled test samples from the target domain are randomly selected as D_u^T for KMM and DTSVM. Note that these test samples are used as unlabeled data during the learning process. Moreover, we fix the parameter θ in (7) as 1 for DTSVM. For all methods, we train one-versus-others SVM classifiers with the fixed regularization parameter $C = 1$.

Three low-level global features Grid Color Moment (225 dim.), Gabor Texture (48 dim.) and Edge Direction Histogram (73 dim.) are used to represent the diverse content of key-frames, because of their consistent, good performance reported in TRECVID [7, 20]. Then the three types of features are put together to form a 346-dimensional feature vector for each keyframe. See [19] for more details about the features.

Kernels are determined before training in all methods. As suggested in [7], in SVM_T, SVM_AT, FR, A-SVM, CD-SVM and KMM, we use Gaussian kernel (*i.e.*, $k(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$) as the default kernel, where γ is set as $\frac{1}{d}$ (d is the feature dimension). For DTSVM and MKL, we additionally use 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.* $k(x_i, x_j) = \frac{1}{\sqrt{\gamma}\|x_i - x_j\| + 1}$). We also use four kernel parameters $1.2^\delta\gamma$, where δ is set as $\{0, 0.5, 1, 1.5\}$ for Gaussian kernel and δ is set as $\{3, 3.5, 4, 4.5\}$ for other three types of kernels. In total, we have 16 base kernels.

4.3. Performance Comparisons

For performance evaluation, we use non-interpolated Average Precision (AP) [13, 16], 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.

Similarly as in [7], we group 36 concepts into three categories according to the frequency of positively labeled samples in the TRECVID 2007 data set. The first group consists of 12 concepts with high positive frequency (more than 0.05), the second group consists of 11 concepts with moderate positive frequency (0.01 $\leq$ positive frequency $\leq$ 0.05), and the third group consists of the remaining 13 concepts with low positive frequency (less than 0.01). In Fig. 1, we use three rows to show the per-concept AP for the three groups. Table 1 gives the Mean Average Precision (MAP) of the concepts of three groups and all 36 concepts, referred to as MAP_Group-1, MAP_Group-2, MAP_Group-3 and MAP_ALL respectively.

From Fig. 1 and Table 1, we have the following observations. Firstly, FR, A-SVM, CD-SVM and KMM outperform SVM_AT and SVM_T in terms of MAP over 36 concepts, which demonstrates that the information from both domains can be effectively used to improve classification performance in the target domain by cross-domain learning methods. We also observe that the overall MAP improvements from CD-SVM and KMM are relatively small, when compared with SVM_AT. Both CD-SVM and KMM match the distributions of patterns from two domains by reweighting the samples of auxiliary domains. In CD-SVM, k-nearest neighbors from the target domain are used to define the weights for the auxiliary patterns. When the total number of training samples in target domain is limited (for example, 10 samples per concept in this work), the weights of the auxiliary patterns are not reliable, which may degrade the performance of CD-SVM. Similarly, KMM learns the weights in an unsupervised setting without using any label information, which may not be as effective as other cross-domain learning methods (*e.g.*, FR and A-SVM).

Secondly, using only training samples from the target domain, MKL_T outperforms SVM_T and SVM_AT in terms of MAP_ALL, which demonstrates the effectiveness of the MKL method. We also observe that the overall MAP over 36 concepts of MKL_AT is worse than that of SVM_T and

	SVM_T	SVM_AT	FR	A-SVM	CD-SVM	KMM	MKL_T	MKL_AT	DTSVM_T	DTSVM_AT	DTSVM_Predict
MAP_Group-1	37.1%	40.0%	40.0%	40.6%	40.1%	40.2%	37.9%	40.3%	39.3%	**44.6%**	**44.6%**
MAP_Group-2	12.2%	12.6%	12.9%	12.7%	12.4%	12.7%	12.5%	12.5%	12.9%	**13.5%**	**13.5%**
MAP_Group-3	15.6%	12.7%	15.4%	15.1%	13.1%	13.0%	16.3%	11.8%	**16.5%**	12.8%	**16.5%**
MAP_ALL	21.7%	21.8%	22.8%	22.8%	21.9%	22.0%	22.4%	21.5%	23.0%	**23.6%**	**24.9%**

Table 1. Performance comparison of DTSVM with other methods. Mean Average Precision (MAP) are from concepts of three groups and all 36 concepts.

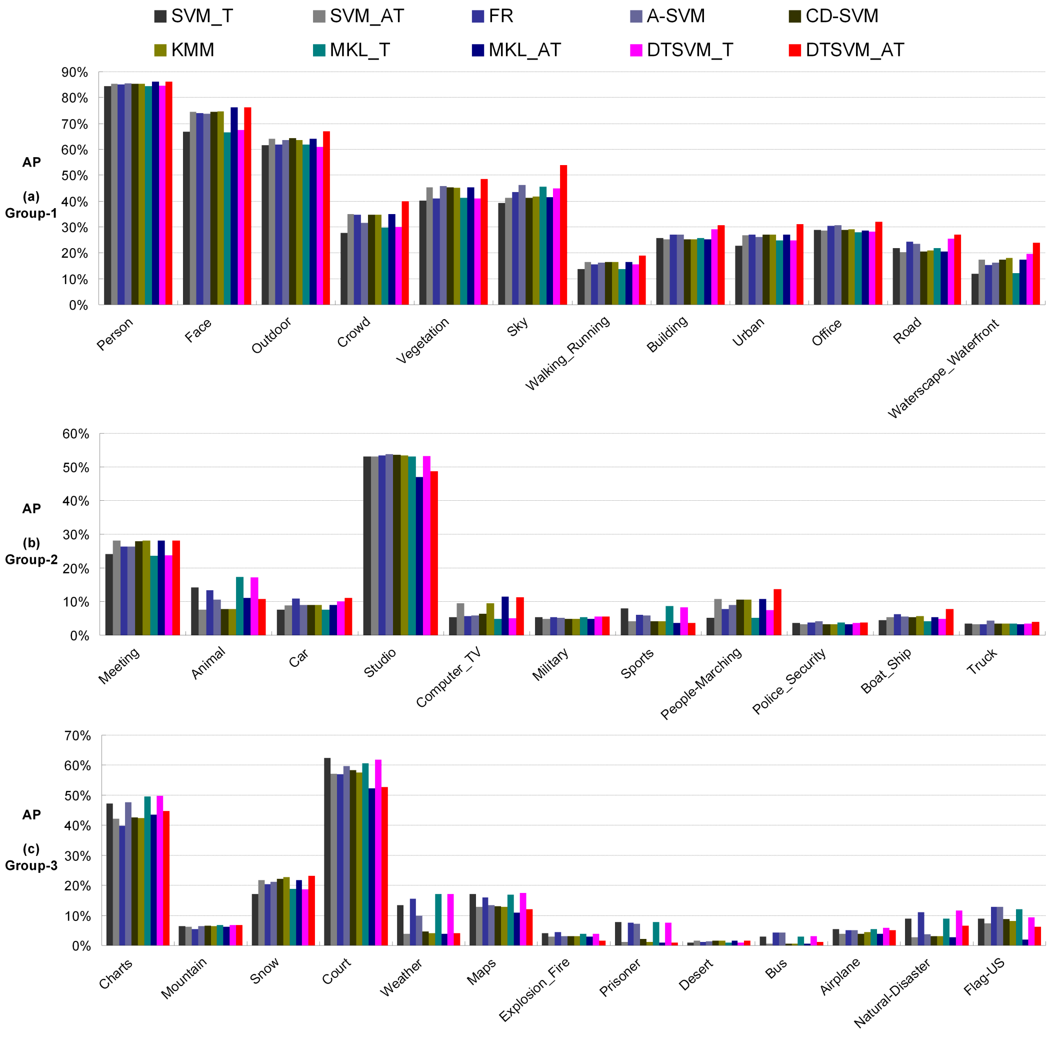

Figure 1. Performance comparison of DTSVM with other methods on all 36 concepts. The concepts are grouped into three categories according to the positive frequency.

SVM_AT. It is possibly because MKL algorithms assume that the training and test data are drawn from the same domain. The feature distribution from different domains may change tremendously. Using MKL method on both the auxiliary domain and the target domain may not produce the optimal kernel for the classification.

Finally, our proposed methods, DTSVM_AT and DTSVM_T, outperform all other algorithms in terms of the MAP_ALL, which demonstrates that DTSVM_AT and DTSVM_T successfully match the data distribution of two domains as well as minimize the structural risk of SVM through effective combination of multiple kernels. DTSVM_AT or DTSVM_T achieve the best results in 24 out of 36 concepts. Some concepts enjoy large performance gains e.g., the AP for the concept “Waterscape_Waterfront” significantly increases from 18.0% (KMM) to 23.9%

(DTSVM_AT), equivalent to a 32.8% relative improvement. When compared with the SVM_AT and MKL_T, the relative MAP_ALL improvement of DTSVM_AT is 8.3% and 5.4% respectively.

In addition, we also compare DTSVM with T-SVM. Again, the labeled data can be from D_l^T or $D_l^T \cup D^A$. We therefore refer to T-SVM in the above two cases as *T-SVM_T* and *T-SVM_AT* respectively. Note that T-SVM_T and T-SVM_AT can also utilize the unlabeled test data, under the assumption that the labeled and unlabeled data are from the same distribution. The MAP of T-SVM_T and T-SVM_AT over all 36 concepts are 20.3% and 20.7% respectively, which are worse than DTSVM_T and DTSVM_AT. We also observe that T-SVM_T (resp. T-SVM_AT) is even worse than SVM_T (resp. SVM_AT) in terms of MAP_ALL, possibly because of the sample selection bias of labeled data from the target domain.

4.4. Predicting Method

As shown from the Table 1, SVM_AT is better than SVM_T in terms of MAP for the concepts in the first group, but it is worse than SVM_T in terms of MAP for the concepts in the third group. The similar phenomenon can be observed for the pair MKL_AT and MKL_T as well as the pair DTSVM_AT and DTSVM_T. As shown in [7], the concepts in the first group generally have high positive frequency and the concepts in the third group generally have low positive frequency in both the auxiliary and the target domains. Intuitively, when sufficient positive samples exist in both domains, they will distribute densely in feature space. The data from the auxiliary domain may be helpful for concept detection in the target domain because the distributions of samples from two domains may overlap [7]. On the other hand, positive samples from both domains will distribute sparsely in feature space, if the patterns from both domains are limited. Therefore, it is more likely that the data from the auxiliary domain may degrade concept detection [7].

Based on the above analysis, we also develop a criterion for method predicting, in which DTSVM_AT is used for the concepts in the first two groups and DTSVM_T is used for the concepts in the third group. Using this simple criterion, our method referred to as *DTSVM_Predict* achieves 24.9% MAP over all 36 concepts. Compared with SVM_AT, KML_T and other cross-domain learning method FR (or A-SVM), the relative MAP_ALL improvements of DTSVM_Predict are 14.2%, 11.2% and 9.2% respectively.

5. Conclusion

We have proposed a unified cross-domain learning framework DTSVM to simultaneously learn a kernel function as well as a SVM classifier by minimizing the structural risk functional of SVM as well as the distribution mismatch of samples between the auxiliary and target domains. For efficiency, we assume that the kernel function in SVM learning is a linear combination of multiple base kernels, which can be efficiently solved by a proposed learning algorithm under a unified convex optimization framework. Moreover, we propose a simple but effective criterion to determine which cross-domain learning method to use for each concept. Experimental results show that DTSVM outperforms existing cross-domain learning and multiple kernel learning methods in the challenging TRECVID data set.

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

[1] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, 2007.

[2] 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. In *ISMB*, 2006.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] S.-F. Chang et al. Large-scale multimodal semantic concept detection for consumer video. In *ACM Workshop on MIR*.

[5] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[6] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, 2006.

[7] W. Jiang et al. Cross-domain learning methods for high-level visual concept classification. In *ICIP*, 2008.

[8] G. Lanckriet et al. Learning the kernel matrix with semidefinite programming. *JMLR*, 27–72, 2004.

[9] M.R Naphade et al. Large-scale concept ontology for multimedia. *IEEE Multimedia Magazine*, 86–91, 2006.

[10] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *AAAI* , 2008.

[11] A. Rakotomamonjy et al. SimpleMKL. *JMLR*, 2491-2521, 2008.

[12] B. Schölkopf and A. Smola. *Learning with kernels*. MIT Press, Cambridge, MA, 2002.

[13] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *ACM Workshop on MIR*, 2006.

[14] S. Sonnenburg et al. Large scale multiple kernel learning. *JMLR*, 1531–1565, 2006.

[15] A. Storkey and M. Sugiyama. Mixture regression for covariate shift. In *NIPS*, 2006.

[16] TRECVID. *http://www-nlpir.nist.gov/projects/trecvid*.

[17] P. Wu and T. G. Dietterich. Improving SVM accuracy by training on auxiliary data sources. In *ICML*, 2004.

[18] D. Xu and S.-F. Chang. Video event recognition using kernel methods with multilevel temporal alignment. *T-PAMI*, 1985–1997, 2008.

[19] A. Yanagawa, W. Hsu, and S.-F. Chang. Columbia University's baseline detectors for 374 LSCOM semantic visual concepts. Technical report, Columbia University, 2007.

[20] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM Multimedia*, 2007.

[21] S. Zanetti, L. Zelnik-Manor, and P. Perona. A walk through the web's video clips. In *International Workshop on Internet Vision, CVPR*, 2008.

[22] X. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison, 2007.