Transfer learning for visual recognition and text categorization

Duan, Lixin

2012

Duan, L. (2012). Transfer learning for visual recognition and text categorization. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/48658>

<https://doi.org/10.32657/10356/48658>

TRANSFER LEARNING FOR VISUAL RECOGNITION AND TEXT CATEGORIZATION

DUAN LIXIN

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2012

Abstract

In recent decades, transfer learning has attracted intensive attention from researchers and become a hot research direction in the field of machine learning. Different from traditional machine learning, transfer learning allows that the training and testing data can be from different domains (*i.e.*, different data distributions and/or different feature spaces). This characteristic helps us to learn good classifiers for the domain of interest, where there have only a limited or even no labeled training data, by utilizing many existing data from other related sources. Because of this, transfer learning techniques have already been widely used in many areas such as machine learning, data mining, computer vision, etc.

In this thesis, we propose several transfer learning frameworks, based on which a number of transfer learning are developed and applied for different real-world applications such as visual recognition and text categorization. Specifically, first we propose a domain transfer framework based on multiple kernel learning to minimize the data distribution mismatch between domains. Two methods are further developed under this framework to simultaneously learn a kernel function modeled by multiple kernel learning as well as a robust target classifier. We demonstrate the effectiveness of our proposed methods in the video concept detection and text classification tasks. Second, we present a visual event recognition framework for consumer videos by leveraging a large number of web videos, in which a pyramid matching method and a transfer learning method have been proposed to measure the distances between videos and cope with the data distribution mismatch between the consumer and web video domains, respectively. In the proposed transfer learning method, we define the target decision function as a linear combination of pre-learned classifiers and a perturbation function modeled by multiple kernel learning, such that we can better fuse the knowledge learned from multiple levels of a video and

also different types of features, which helps learn a robust classifier. Third, we propose a domain-dependent regularization framework to handle the transfer learning problems where there exist multiple source domains. In this framework, a domain-dependent regularizer is defined based on a set of pre-learned classifiers by enforcing the smoothness that the target classifier shares similar decision values with the pre-learned classifiers on the target unlabeled samples. Furthermore, two methods are presented by incorporating least-squares SVM into the proposed framework. One of them employs a sparsity regularizer based on the ϵ -insensitive loss. And the other one additionally makes use of the Universum regularizer defined on data from the source domains. Experimental results show the good performances of our proposed methods in the video concept detection and information retrieval tasks with multiple source domain settings.

Acknowledgments

The completion of my Ph.D. journey would not become a reality without good fortune and the aggregate support of many people and several institutions.

First of all, I would like to thank my advisor, Xu Dong. His valuable advice and great guidance have supported and encouraged me over the past four years. Under his supervision, I have learned not only the cutting-edge research on computer vision but also his inspiring and down-to-earth working attitude. From him, I have also learned the way to become an independent researcher and the way to write qualified scientific papers. All of those learned from him will be great wealth in my future career.

I would also like to thank my main collaborator, Tsang Wai-Hung. Besides the advice from my supervisor, I have been truly lucky to work with Ivor, since my first year of my Ph.D. life. His deep insights into machine learning, especially statistical learning, always inspire and guide me to work out difficult problems I ever encountered. Also, his continuous encouragement makes me believe that there will be a bright future for me by hard working.

I am appreciated of the support from my thesis committee members, Ngo Chong-Wah from City University of Hong Kong, Zhang Lei from Microsoft Research Asia and Deepu Rajan. Their valuable comments offer me great help to make this thesis more complete. Deepu Rajan also serves as a panel member for my oral defense. Besides him, I am also thankful to other panel members, Cai Jianfei, Chia Liang Tien and Wu Jianxin. My special thanks go to Cai Jianfei for his concern and encouragement upon me during my Ph.D. study.

It is my great pleasure to live the four years with the excellent colleagues in our research group, Chen Lin, Hoang Anh, Huang Yi, Kan Meina, Li Jia, Li Wen, Liu Huiying, Liu Jianyi, Liu Yiming, Nie Feiping, Wei Shikui, Wu Xinxiao, Xiao Shijie,

Xiong Zhiwei, Xu Xinxing, Xu Yanwu, Yang Yi, Zeng Zinan, Yan Shengye, and others in CeMNet and CAIS, Cheng Xiangang, Duan Qi, Gao Shenghua, Hu Yiqun, Li Gang, Li Guangxia, Lu Haifeng, Teng Xiao, Tu Trung Hieu, Wang Danqi, Wang Zhengxiang, Yang Ming, Yang Shengbo, Zhang Juyong, Zhang Yu, Zhao Peilin, Zhong Feng, Zhuang Jinfeng. I also want to give my special thanks to some of my best friends, Hou Li, Li Ruoying, Ni Tian. Thanks to all of them, I can survive the long Ph.D. journey and be able to move towards the next in my life.

Last but the most importantly, I sincerely thank my parents for their love. ^_^

Dedicated to my parents.
谨以此文献给我的父母。

Contents

Abstract	i
Acknowledgments	iii
List of Figures	xi
List of Tables	xv
Notations and Nomenclatures	xvii
 1 Introduction	 1
1.1 Thesis Contribution	4
1.2 Thesis Structure	5
 2 Literature Review on Transfer Learning	 9
2.1 Transfer Learning Categorization	10
2.1.1 Inductive Transfer	11
2.1.2 Transductive Transfer	13
2.1.3 Unsupervised Transfer	15
2.2 Bounds in Transfer Learning	15
2.3 Transfer Learning with Other Machine Learning Approaches	16
2.3.1 Active Learning	16
2.3.2 Metric Learning	17
2.3.3 Online Learning	17
2.4 Real-World Applications	18
 3 Domain Transfer Multiple Kernel Learning	 19
3.1 Introduction	19
3.2 Brief Review of Related Work	22

3.2.1	Domain Transfer via Reducing Mismatch between Data Distributions of Two Domains	23
3.2.2	Domain Transfer via Existing Classifiers	24
3.3	Domain Transfer Multiple Kernel Learning Framework	25
3.3.1	Minimizing data distribution mismatch	25
3.3.2	Minimizing structural risk functional	26
3.3.3	Multiple base kernels	26
3.3.4	Learning algorithm	27
3.3.5	DTMKL using Hinge Loss	28
3.3.6	DTMKL using Existing Base Classifiers	30
3.3.7	Computational Complexity of DTMKL	33
3.3.8	Discussions with Related Work	34
3.4	Experiments	36
3.4.1	Descriptions of Data Sets and Features	36
3.4.2	Experimental Setup	40
3.4.3	Results of Video Concept Detection	42
3.4.4	Results of Text Categorization	47
3.4.5	Convergence	54
3.5	Summary	54
4	Adaptive Multiple Kernel Learning	57
4.1	Introduction	57
4.2	Related Work on Event Recognition	61
4.3	Aligned Space-Time Pyramid Matching	62
4.4	Adaptive Multiple Kernel Learning	66
4.4.1	Brief Review of Related Work	66
4.4.2	Formulation of A-MKL	67
4.4.3	Learning Algorithm of A-MKL	70
4.4.4	Discussions with Related Work	71
4.5	Experiments	73
4.5.1	Data Set Description and Features	73

4.5.2	Aligned Space-Time Pyramid Matching vs. Unaligned Space-Time Pyramid Matching	75
4.5.3	Performance Comparisons of Transfer Learning Methods	76
4.5.4	Analysis on the Combination Coefficients of the Pre-learned Classifiers	81
4.5.5	Convergence of A-MKL Learning Algorithm	82
4.5.6	Utilization of Additional Pre-learned Classifiers from Other Event Classes	83
4.5.7	Performance Variations of A-MKL using Different Proportions of Labeled Consumer Videos	85
4.5.8	Running Time and Memory Usage	85
4.6	Summary	86
5	Domain Adaptation Machine	89
5.1	Introduction	89
5.2	Brief Review of Related Work	92
5.2.1	Multiple Domain Transfer via Existing Classifiers	93
5.2.2	Regularizations	94
5.3	Domain Adaptation Machine Framework	95
5.3.1	Smoothness Assumption for Domain Adaptation	95
5.3.2	Proposed Framework	97
5.3.3	Domain Adaptation Machine with Fast Prediction	98
5.3.4	Domain Adaptation Machine with Universum	101
5.4	Discussions	105
5.4.1	Connection between FastDAM and UniverDAM	105
5.4.2	Connection to Support Vector Regression	106
5.4.3	Discussions with Related Work	107
5.5	Experiments	107
5.5.1	Descriptions of Data Sets	109
5.5.2	Experimental Setup	112
5.5.3	Performance Comparisons	115
5.5.4	Parameter Analysis for Different Methods	121
5.6	Summary	124

6	Conclusion and Future Work	127
6.1	Conclusion	127
6.2	Future Work	131
6.2.1	Avoiding Negative Transfer	131
6.2.2	Speed-Up in the Large-Scale Setting	132
6.2.3	Transfer Learning in Computer Vision Applications	133
	References	135
A	Appendix	163
A.1	Derivation of the dual problem of $J(\mathbf{d})$ in DTMKL- f in (3.15)	163
A.2	Proof of Theorem 5.3	164
A.3	Proof of Theorem 5.4	165
A.4	Transductive Inference with the Universum	166
	Publication	169

List of Figures

1.1	A comparison between traditional machine learning and transfer learning.	2
1.2	The structure of this thesis.	6
3.1	Illustration of virtual labels. The base classifier $f^{B,m}$ is learned with the base kernel function k_m and the labeled training data from $\mathcal{D}$, where $m = 1, \dots, M$. For each of the unlabeled target pattern $\mathbf{x}$ from $\mathcal{D}_u^T$, we can obtain its decision value $f^{B,m}(\mathbf{x})$ from each base classifier. Then the virtual label $\tilde{y}$ of $\mathbf{x}$ is defined as the linear combination of its decision values $f^{B,m}(\mathbf{x})$'s weighted by the coefficients d_m 's, <i>i.e.</i> , $\tilde{y} = \sum_{m=1}^M d_m f^{B,m}(\mathbf{x})$. . .	32
3.2	Per-concept APs of all the 36 concepts using different methods. The concepts are divided into three groups according to the positive frequency. Our methods achieve the best performances for the circled concepts. . . .	44
3.3	Performance comparisons of DTMKL- f with other methods in terms of the means and standard deviations of classification accuracies on the 20 News-groups data set by using different tradeoff parameter $C = 0.1, 0.2, 0.5, 1, 2, 5, 10, 20$ and 50. We set $m = 5$ (top) and $m = 10$ (bottom).	53
3.4	Performance (<i>i.e.</i> , the means of classification accuracies) variation of DTMKL- f with respect to the balance parameter $\lambda \in [0.1, 10]$ on the 20 News-groups data set. We set the tradeoff parameter $C = 2$ and $C = 5$	53
3.5	Illustration of the convergence of DTMKL-AT.	54

4.1	Four sample frames from consumer videos and YouTube videos. Our work aims to recognize the events in consumer videos by using a limited number of labeled consumer videos and a large number of YouTube videos. These two events and their examples (<i>i.e.</i> , “picnic” and “sports”) illustrate the considerable appearance differences between consumer videos and YouTube videos, which poses great challenges to conventional learning schemes but can be effectively handled by our transfer learning method, called Adaptive Multiple Kernel Learning (A-MKL).	59
4.2	The flowchart of the proposed visual event recognition framework. It consists of an <i>pyramid matching</i> method ASTPM that effectively measures the distances between two video clips and a <i>transfer learning</i> method A-MKL that effectively copes with the considerable variation in feature distributions between web videos and consumer videos.	59
4.3	Illustration of the proposed Aligned Space-Time Pyramid Matching (ASTPM) method at level-1: (a) Each video is divided into 8 space-time volumes along the width, height and temporal dimensions; (b) The matching results are obtained by using our ASTPM method. Each pair of matched volumes from two videos is highlighted in the same color. For better visualization, please see the colored PDF file.	64
4.4	Means and standard deviations of per-event APs of six events for all methods.	79
4.5	Illustration of the combination coefficients β_p 's of the pre-learned classifiers for all events.	81
4.6	Illustration of the convergence of A-MKL learning algorithm for all events.	82
4.7	Means and standard deviations of MAPs over six events for SVM-T, DTSVM, A-MKL ₄ and A-MKL ₂₄ when using different proportions (<i>i.e.</i> , r) of labeled training consumer videos.	84
5.1	Base classifiers learned by using the labeled training samples from the source domains (and the target domain as well). For each unlabeled instance $\mathbf{x}$ in $\mathcal{D}_u^T$, we define its <i>virtual label</i> $\tilde{y} = \sum_{s=1}^P \tilde{\gamma}_s f^s(\mathbf{x})$ as a weighted summation of the decision values $f^s(\mathbf{x})$'s from the base classifiers f^s 's on $\mathbf{x}$, where $\tilde{\gamma}_s = \frac{\gamma_s}{\sum_{s=1}^P \gamma_s}$	97

5.2	Per-concept APs of all the 36 concepts using different methods. The concepts are divided into three groups according to the positive frequency. .	116
5.3	Means and Standard Deviations (%) of APs of all the methods on the 20 Newsgroups data set with different tradeoff parameter $C = 0.01, 0.1, 1, 10$ and 100.	121
5.4	Means and Standard Deviations (%) of APs of FastDAM and UniverDAM on the 20 Newsgroups data set with different tradeoff parameters $\lambda_D = \lambda_{D_1} = \lambda_{D_2} = 0.01, 0.1, 1, 10, 100$ and 1000.	122
5.5	Means and Standard Deviations (%) of APs of FastDAM and UniverDAM on the 20 Newsgroups data set with different tradeoff parameter $\lambda = 0.1, 1, 10, 100$ and 1000.	123
5.6	Means and Standard Deviations (%) of APs of FastDAM and UniverDAM on the 20 Newsgroups data set with different bandwidth parameter $\beta = 0, 0.01, 1, 100$ and 10000 in (5.32).	124

List of Tables

3.1	Description of the 20 Newsgroups data set.	39
3.2	Mean average precisions (MAPs) (%) of all the methods on the TRECVID data set. MAPs are from concepts of three individual groups and all 36 concepts.	43
3.3	Average training (TR) and testing (TE) time (in second) comparisons of all the methods on the TRECVID data set. For A-SVM and DTMKL- f , the two numbers represent the average training time for the learning of the pre-learned classifiers and the learning of the target classifier.	48
3.4	Means and standard deviations (%) of classification accuracies (ACC) of all the methods with different number of positive and negative training samples (<i>i.e.</i> , m) from the target domain on the 20 Newsgroups data set. Each result in the table is the best among all the results obtained by using different tradeoff parameter $C = 0.1, 0.2, 0.5, 1, 2, 5, 10, 20$ and 50. The results shown in boldface are significantly better than the others, judged by the t-test with a significance level of 0.1.	49
3.5	Means and standard deviations (%) of classification accuracies (ACC) of all the methods with five positive and five negative training samples from the target domain on the email spam data set. Each result in the table is the best among all the results obtained by using different tradeoff parameter $C = 0.1, 0.2, 0.5, 1, 2, 5, 10, 20$ and 50. The results shown in boldface are significantly better than the others, judged by the t-test with a significance level of 0.1.	50
4.1	Numbers of videos for six events in our data set.	74

4.2	Means and standard deviations (%) of MAPs over six events at different levels using SVM with the default kernel parameter for SIFT features. . .	75
4.3	Means and standard deviations (%) of MAPs over six events at different levels using SVM with the default kernel parameter for ST features. . . .	76
4.4	Means and standard deviations (%) of MAPs over six events for all methods in three cases.	79
4.5	Means and standard deviations (%) of MAPs of A-MKL (referred to as A-MKL_4) using the pre-learned average classifiers from the same event class and A-MKL (referred to as A-MKL_24) using the pre-learned average classifiers from all six event classes. Different sets of kernel parameters (<i>i.e.</i> , $\mathcal{H}$) are employed to obtain the pre-learned average classifiers.	83
5.1	Summary of the comparisons between our two methods (<i>i.e.</i> , FastDAM and UniverDAM) and other domain adaptation methods.	108
5.2	Description of the TRECVID 2005 data set.	110
5.3	Description of the 20 Newsgroups data set.	110
5.4	Description of the email spam data set.	110
5.5	MAPs (%) of all the methods over 36 concepts on the TRECVID 2005 data set.	117
5.6	Means and Standard Deviations (%) of APs of all the methods with m positive and m negative training samples from the target domain on the 20 Newsgroups data set. The results shown in boldface are significantly better than the others, judged by the t-test with a significance level at 0.01.	118
5.7	Means and Standard Deviations (%) of APs of all the methods with 10 positive and 10 negative training samples from the target domain on the email spam data set. The results shown in boldface are significantly better than the others, judged by the t-test with a significance level at 0.01. . .	120
5.8	Average training and testing time (second) over ten rounds of experiments for all methods on the first setting (<i>i.e.</i> , rec vs. sci) of the 20 Newsgroups data set. Note that the training time of both FastDAM and UniverDAM consists of two parts (<i>i.e.</i> , the calculation of the virtual labels and the learning of the classifier).	120

List of Notation and Nomenclatures

In this thesis, we use a lowercase character to represent a scalar (*i.e.*, b). And a bold lowercase character is to denote a vector (*i.e.*, $\mathbf{x}$). Moreover, A matrix is denoted by a bold uppercase character (*i.e.*, $\mathbf{A}$). We list main notations and operators with their nomenclatures used in this thesis as follows:

$\mathcal{H}$	Reproducing kernel Hilbert space (RKHS)
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	n -dimensional real value space
$\mathbf{x}_i$	Feature vector of the i -th sample
$\mathbf{x}_i^T$	Feature vector of the i -th sample from the target domain
$\mathbf{x}_i^S$	Feature vector of the i -th sample from the source domain
$\mathbf{x}_i^s$	Feature vector of the i -th sample from the s -th source domain
y_i	Label of the i -th sample
y_i^T	Label of the i -th sample form the target domain
y_i^S	Label of the i -th sample form the source domain
y_i^s	Label of the i -th sample form the s -th source domain
n_T	Number of samples from the target domain
n_l	Number of labeled samples from the target domain
n_u	Number of unlabeled samples from the target domain
n_S	Number of samples from the source domain
n_s	Number of samples from the s -th source domain
$'$	Transpose of a vector or matrix
b	Bias term
$\mathbf{w}$	Weight vector in the feature space
$\mathbf{0}_n$	$n \times 1$ vector with every element as zero
$\mathbf{1}_n$	$n \times 1$ vector with every element as one
$\mathbf{u} \circ \mathbf{v}$	Element-wise multiplication of two vectors $\mathbf{u}$ and $\mathbf{v}$
$\mathbf{K}$	Kernel matrix
$\mathbf{I}_n$	$n \times n$ identity matrix

$\text{diag}(\mathbf{u})$	Diagonal matrix with its diagonal elements as $\mathbf{u}$
$\text{tr}(\mathbf{A})$	Trace of the matrix $\mathbf{A}$
$\phi(\cdot)$	Nonlinear feature mapping function
$k(\cdot, \cdot)$	Kernel function induced by $\phi(\cdot)$
$\ \cdot\ _{\mathcal{H}}$	Norm in the RKHS $\mathcal{H}$
$E_{\mathbf{x} \sim \mathcal{U}}[\cdot]$	Expectation operator under the data distribution $\mathcal{U}$

Chapter 1

Introduction

Supervised machine learning has been studied in depth and applied to different areas (such as data mining [1], pattern recognition [2], etc.) for many years. In most traditional supervised learning methods, however, there is a common assumption that the training and test data are drawn from the same domain, implying that the data are under the same distribution and also their feature representations share the same feature space (*i.e.*, the same type of features). More than that, the good performances of such traditional algorithms cannot be achieved, unless sufficient labeled training data are accessible in order to train robust models or classifiers for the sake of better predictions on the newly coming data [3–5]. In many real-world applications, there are only a limited number of labeled training data which cannot help train robust classifiers. And it is usually very expensive and time-consuming to obtain labeled training data under the same data distribution. To overcome these knotty issues, researchers have proposed Semi-Supervised Learning (SSL) [6, 7] and Active Learning [8–10] in the last a few decades. SSL tries to make use of a large number of unlabeled training data and very few labeled training data to learn good classifiers. However, similar to most traditional supervised learning methods, traditional SSL methods also work under the constrained environment that training data (both labeled and unlabeled) and the test data should be from the same distribution. As a result, if unlabeled data from another distribution are considered as the training data, the outcomes of data prediction could be uncertain for those traditional SSL methods. On the other hand, active learning is proposed to alleviate the effort of training data annotation by repeatedly asking an oracle (*e.g.*, usually a human annotator) to label the unlabeled samples selected from active learning systems. However, active

Figure 1.1: A comparison between traditional machine learning and transfer learning.

learning is still under the restrictive data distribution assumption. Moreover, the amount of effort for annotating training data, though it has been reduced in active learning, is still considerable to learn good classifiers.

To free the same-data-distribution constraint in traditional machine learning methods, transfer learning has been proposed by allowing the training and testing data to be from different yet related domains. The modern definition for transfer learning is to *retain and apply previous knowledge learned from one or multiple existing domains/tasks to improve learning in the new domains/tasks of interest (a.k.a., the target domains/tasks)*. Nowadays, transfer learning serves as a general machine learning term that covers a variety of approaches such as multi-task learning¹, domain adaptation, sample selection bias, covariate shift, etc [12, 13]. We present Figure 1.1 to provide an intuitional difference between traditional machine learning and transfer learning.

In the recent decade, transfer learning has been observed and successfully applied in many real-world applications. One good example related to visual recognition is the video concept classification task of TRECVID [14] sponsored by the National Institute of Standards and Technology (NIST)² every year since 2003. This classification task requires developing new visual concept classification techniques to classify the images or videos into high-level semantic concepts (*e.g.*, “person”, “animal”, “building” and so

¹Researchers usually refer to the transfer learning problem where there are multiple domains/tasks of interest as Multi-Task Learning (MTL) [11]. And all tasks are learned simultaneously in MTL.

²<http://trecvid.nist.gov>

on). Let us take the TRECVID 2005 and 2007 data sets (referred to as TRECVID 2005 and TRECVID 2007, respectively, for short) for example. The TRECVID 2005 data set collected from video programs of six broadcast sources in three different languages (*i.e.*, English, Arabic and Chinese) is fully annotated, but TRECVID data sets collected in the subsequent years have very limited annotations and are collected from different sources (*e.g.*, the videos from TRECVID 2007 are all about news magazine, science news, documentaries and educational programming). As a result, even for the same semantic concept, the data distribution of TRECVID 2007 is different from that of TRECVID 2005. Therefore, on one hand, one cannot learn robust classifiers by using the limited labeled training data on TRECVID 2007. On the other hand, while it is possible to use the traditional machine learning methods to train classifiers based on labeled training data from TRECVID 2005 or the combined data set from both TRECVID 2005 and 2007, the learned classifiers may still perform poorly on TRECVID 2007. In such a case, it is expected to transfer the knowledge from the broadcast sources on TRECVID 2005 into the new domain on TRECVID 2007.

Another example for transfer learning is to index and retrieve consumer data (*e.g.*, personal photo collections [15, 16] or videos taken by consumers [17]) by utilizing a large number of web data. As consumers may be reluctant to do the tedious annotation work, there might be very few or even no labeled data samples that can be used for training. Besides this, as consumer data are generally captured by amateurs during personal activities, they may contain a certain amount of occlusion, cluttered background and large intra-class variations within the same type of class. In contrast, the web data can be taken by professionals, compressed, or edited before they are uploaded onto the Internet. Thus, the feature distribution of consumer data is intuitively different from that of the web data. To meet with the demand of better managing consumer data, transfer learning is no doubt the first choice to adapt the web data to learn good models for data from the consumer domain.

The need for transfer learning also exists in a couple of text categorization applications [18–20]. For instance, in the email spam filtering application, email service providers need to develop robust email spam filters in order to detect and block spams before they reach users. To achieve this goal, these providers can employ some users and ask each of

them to manually label a small set of their individual emails which are used as a limited number of labeled training data to train robust filters³. Moreover, as the distributions of the emails received by different users can be different, one has to cope with the discrepancy between the distributions to learn the filter for emails from the target user if using the labeled emails from other users as training data. Such discrepancy issues can be handled by using transfer learning techniques.

Besides the several aforementioned real-world applications, there are also many other applications involving the issue of transfer learning, such as image classification [21], image clustering [22], natural language processing [23], wireless sensor networks [24, 25], sentiment classification [19], and so on and so forth (see the literature review in Chapter 2 for a thorough study on transfer learning applications).

1.1 Thesis Contribution

This thesis includes conceptual and algorithmic contributions to transfer learning. We itemize the contributions as follows:

- We propose a novel transfer learning framework, referred to as Domain Transfer Multiple Kernel Learning (DTMKL), to cope with the considerable change between feature distributions of different domains. The DTMKL framework simultaneously learns a kernel function (modeled by using multiple kernel learning techniques) and a robust target classifier by minimizing both the structural risk functional of any kernel method (such as support vector machine, support vector regression, etc.) and the distribution mismatch between the data samples from the source and target domains. Based on the proposed framework, we further develop two novel methods, DTMKL-AT and DTMKL- f , on the basis of SVM and pre-learned classifiers, respectively.
- We propose a new visual event recognition framework for consumer videos by leveraging a large number of web videos. In this framework, we first develop a new

³It is reasonable to assume that the number of emails annotated by each user is limited, since the cost of such an annotation process is usually expensive.

pyramid matching method called Aligned Space-Time Pyramid Matching (ASTP-M) to measure the distances between two videos. We then propose a new transfer learning method called Adaptive Multiple Kernel Learning (A-MKL). Similar to DTMKL, we also cope with the feature distribution mismatch between videos from the consumer and web video domains. Moreover, in A-MKL, we learn a pre-learned classifier by using training data from one type of feature and a pyramid level in a video. We fuse the knowledge information from multiple pyramid levels and features by modeling an adapted target classifier of A-MKL as a linear combination of pre-learned classifiers and a so-called perturbation function based on multiple kernel learning.

- For the transfer learning problem with multiple source domains and one single target domain, we propose a domain-dependent regularization framework called Domain Adaptation Machine (DAM). By using a set of pre-learned base classifiers, we propose a new domain-dependent regularizer based on smoothness assumption in DAM, which enforces that the target classifier shares similar decision values with the relevant base classifiers on the unlabeled samples from the target domain. Based on our framework, we develop two new domain adaptation methods referred to as FastDAM and UniverDAM, respectively. In FastDAM, a sparsity regularizer based on the ϵ -insensitive loss is employed to enforce the sparsity of the target classifier with the support vectors only from the target domain such that the prediction on any future test sample is very fast. In UniverDAM, besides the sparsity regularizer, we make use of the source data as Universum [26] to further enhance the generalization ability of the target classifier.

1.2 Thesis Structure

There will be four chapters for the rest of this thesis. The structure of this thesis is presented in Figure 1.2. We summarize the main contents of chapters as follows:

Chapter 2. Literature Review on Transfer Learning In this chapter, we will give a thoroughly study on transfer learning.

Figure 1.2: The structure of this thesis.

Chapter 3. Domain Transfer Multiple Kernel Learning In this chapter, we will propose a domain transfer framework based on multiple kernel learning for the transfer learning problems with a single source domain and a single target domain. Our DTMKL framework aims to minimize the mismatch between data distributions of the source and target domains in order to learn a kernel function as well as a robust target classifier. Under this framework, we develop two methods DTMKL_{AT} and DTMKL_f. And we demonstrate the effectiveness of these two methods for the video concept detection and text classification tasks.

Chapter 4. Adaptive Multiple Kernel Learning In this chapter, we will propose a transfer learning method A-MKL to recognize consumer videos by leveraging a large number of loosely labeled web videos. Similar to DTMKL, A-MKL also minimizes the data distribution mismatch between the source and target domains. In order to fuse the

information from multiple levels of videos and also different types of features, in A-MKL, we propose to learn a target classifier composed of a linear combination of pre-learned classifiers and a perturbation function modeled by multiple kernel learning. Besides the proposed A-MKL, we also develop a new pyramid matching method ASTPM to effectively measure the distance between two videos. Comprehensive experiments demonstrate the effectiveness of both ASTPM and A-MKL for the event recognition of consumer videos.

Chapter 5. Domain Adaptation Machine In this chapter, we will propose a domain-dependent regularization framework for the transfer learning problems with multiple source domains and one single target domain. In our DAM framework, we propose a new domain-dependent regularizer by applying smoothness assumption for a set of pre-learned base classifiers. Based on our framework, we develop two domain adaptation methods called FastDAM and UniverDAM. By introducing a sparsity regularizer into FastDAM, the prediction of its classifier is very fast. By considering the source data as a unlabeled data collection, we additionally employ the Universum regularizer [26] in UniverDAM to enhance the generalization ability of the target classifier. Experiments for video concept detection and information retrieval clearly demonstrate the effectiveness of the proposed domain-dependent regularizer as well as the utilization of source data as Universum.

Chapter 6. Conclusions and Future Work In the final chapter, we will summarize our thesis. In the future work on transfer learning, we will discuss the issue of negative transfer and show some possible ways to avoid negative transfer. We also talk about the speed-up issues for the transfer learning applications with thousands of millions of data. In the end, we present several computer vision applications related to transfer learning which deserve future efforts.

Chapter 2

Literature Review on Transfer Learning

From a psychological perspective, the goal of *transfer learning* is to explore how human beings would transfer the knowledge learned in previous context to a new context that shares similar characteristics with faster or better solutions [27,28]. And in the machine learning community, the fundamentals of transfer learning were first discussed in the NIPS 1995 workshop “Learning to Learn: Knowledge Consolidation and Transfer in Inductive Systems”¹, where researchers tried to determine the unifying principles of transfer learning and focused on lifelong machine learning methods that can benefit the current task by transfer knowledge from other sources. Ten years later, in NIPS 2005 workshop “Inductive Transfer”², researchers examined the progress of transfer learning over the last ten years and found that some fundamental questions (*e.g.*, “When to do transfer learning?”, “How to avoid negative transfer?”, etc.) still remained unsolved.

Despite of those open problems in transfer learning, many transfer learning methods have been proposed to date and published in top venues, such as *Journal of Machine Learning Research*, *Machine Learning*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *Transactions on Neural Networks*, *Annual Conference on Neural Information Processing Systems*, *International Conference on Machine Learning*, *European Conference on Machine Learning*, *AAAI Conference on Artificial Intelligence*, *International Joint Conference on Artificial Intelligence*, *Annual Meeting of the Association for*

¹http://socrates.acadiau.ca/courses/comp/dsilver/NIPS95_LTL/transfer.workshop.1995.html

²<http://iitrl.acadiau.ca/itws05/index.htm>

Computational Linguistics, International World Wide Web Conference, Conference on Empirical Methods in Natural Language Processing, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, IEEE International Conference on Computer Vision and Pattern Recognition, IEEE International Conference on Computer Vision, European Conference on Computer Vision, and so on and so forth.

2.1 Transfer Learning Categorization

Transfer learning can be categorized into smaller learning approaches by using different ways [12]. Based on different definitions of the knowledge to be transferred, transfer learning can be summarized into four approaches: *instance transfer*, *feature-representation transfer*, *parameter transfer*, *relational-knowledge transfer*. Instead, based on different situations between the source and target domains/tasks, transfer learning is categorized into three settings: *inductive transfer*, *transductive transfer* and *unsupervised transfer*. In this thesis, we study existing transfer learning techniques according to the latter categorization. Before we look into the three settings, let us first present some formal definitions for transfer learning.

Definition 2.1 *A domain $\mathcal{D}$ is a set composed of a feature space $\mathcal{X}$ with a marginal distribution $P(\mathbf{x})$, namely, $\mathcal{D} = \{\mathcal{X}, P(\mathbf{x})\}$, where $\mathbf{x} \in \mathcal{X}$.*

Definition 2.2 *Given a specific domain $\mathcal{D}$, a task $\mathcal{T}$ is a set composed of a label space $\mathcal{Y}$ and a decision function $f(\cdot)$, namely, $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$. The decision function $f(\cdot)$ can be learned by using a set of training data $(\mathbf{x}_i, y_i)$ 's³, where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, and is used to predict the label of any test sample $\mathbf{x}$ as $f(\mathbf{x})$. From a statistical point of view, $f(\mathbf{x})$ can be written as $P(y|\mathbf{x})$.*

In order to clearly present the three settings of transfer learning, we only consider the case with one source domain $\mathcal{D}^S = \{\mathcal{X}^S, P(\mathbf{x}^S)\}$ with task $\mathcal{T}^S = \{\mathcal{Y}^S, f^S(\cdot)\}$ and one target domain $\mathcal{D}^T = \{\mathcal{X}^T, P(\mathbf{x}^T)\}$ with task $\mathcal{T}^T = \{\mathcal{Y}^T, f^T(\cdot)\}$ for now. Based on the above definitions, we formally define transfer learning as follows:

³For a binary classification task, $y_i \in \{1, -1\}$.

Definition 2.3 *Transfer learning is a machine learning problem, the goal of which is to help learn a target decision function $f^T(\cdot)$ by using the knowledge transferred from $\mathcal{D}^S$ and $\mathcal{T}^S$, where $\mathcal{D}^T \neq \mathcal{D}^S$ or $\mathcal{T}^T \neq \mathcal{T}^S$.*

In the above definition, $\mathcal{D}^T \neq \mathcal{D}^S$ implies either $\mathcal{X}^T \neq \mathcal{X}^S$ or $P(\mathbf{x}^T) \neq P(\mathbf{x}^S)$. And $\mathcal{T}^T \neq \mathcal{T}^S$ implies either $\mathcal{Y}^T \neq \mathcal{Y}^S$ or $f^T(\cdot) \neq f^S(\cdot)$ (i.e., statistically $P(y^T|\mathbf{x}^T) \neq P(y^S|\mathbf{x}^S)$). We can also observe that if $\mathcal{D}^T = \mathcal{D}^S$ and $\mathcal{T}^T = \mathcal{T}^S$, the learning problem becomes the traditional machine learning problem.

2.1.1 Inductive Transfer

In inductive transfer, the target and source tasks are different yet related, while the target and source domains could be either the same or not the same. And There are required to be a few labeled data from the target domain available to train the target decision function. We formally define the inductive transfer setting as follows:

Definition 2.4 *Inductive transfer is a setting of transfer learning, the goal of which is to help learn a target decision function $f^T(\cdot)$ by using the knowledge transferred from $\mathcal{D}^S$ and $\mathcal{T}^S$, where $\mathcal{T}^T \neq \mathcal{T}^S$ and there exist a set of labeled target data available for training.*

As in this setting labeled training data from the target domain must be present, the existence of either data from the source domain or unlabeled data from the target domain is optional. As a result, three cases can be obtained under the setting of inductive transfer:

- Data from the source domain are available, while unlabeled data from the target domain are unavailable.
- Data from the source domain are available, and unlabeled data from the target domain are available as well.
- Data from the source domain are unavailable, while unlabeled data from the target domain are available.

Most transfer learning methods under the inductive setting fall into the first two cases described above. To utilize labeled training samples from both the source and target domains, Daumé III [29] 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.* [30] proposed 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 samples from a source domain. Similar to A-SVM, Schweikert *et al.* [31] proposed to first learn classifiers from the source and target domains by using supervised learning methods (*e.g.*, SVM), and then these classifiers are combined with some pre-defined parameters to obtain the final classifier for the prediction in the target domain. Dai *et al.* [20] proposed a boosting method called TrAdaBoost by extending AdaBoost [32], which iteratively reweights the source data in order to automatically select and adapt part of the source data to learn a good target classifier. Wu and Dietterich [33] proposed to learn a target classifier by directly integrating the source data into an SVM framework. Cross-domain SVM (CD-SVM) proposed by Jiang *et al.* [34] used k -nearest neighbors from the target domain to define a weight for each source sample, and then an SVM-like classifier was trained with the re-weighted source samples and the labeled target samples. Jiang *et al.* [35] proposed to mine the relationship among different visual concepts for video concept detection. They first built a semantic graph and the graph can be then adapted in an online fashion to fit the new knowledge mined from the test data.

When there are no labeled data from the source domain (only unlabeled source data are available), the source domain information can be transferred to the target domain and task. Raina *et al.* [21] called such a case as self-taught learning. In this case, they proposed an approach that used sparse coding to construct higher-level features by using the unlabeled source data [21].

However, numerous unlabeled samples in the target domain are not exploited in the above transfer learning methods [20, 21, 29–31, 33–35]. As shown in [17, 36], such unlabeled samples can also be employed to improve the generalization performance. Duan *et al.* [17, 36] proposed to utilize the unlabeled target data to more precisely measure the data distribution mismatch between the source and target domains based on the maximum mean discrepancy [37].

There are also cases where feature spaces of the source and target domains are different (*e.g.*, different data feature vectors with different lengths). Dai *et al.* [38] proposed a language model by using a Markov chain and risk minimization to link the heterogenous data from different feature spaces. Saenko *et al.* [16] and Kulis *et al.* [39] proposed to explore the relationship between the two entire different feature spaces by learning a distance metric based on the information-theoretic metric learning method [40].

2.1.2 Transductive Transfer

Under the transductive transfer setting, the source and target tasks are assumed to be the same, while the source and target domains are different⁴. Moreover, there are no labeled data available in the target domain, while many source data are there. We formally define the transductive transfer setting as follows:

Definition 2.5 *Transductive transfer is a setting of transfer learning, the goal of which is to help learn a target decision function $f^T(\cdot)$ by using the knowledge transferred from $\mathcal{D}^S$ and $\mathcal{T}^S$, where $\mathcal{D}^T \neq \mathcal{D}^S$ and $\mathcal{T}^T = \mathcal{T}^S$, and the target training data are all unlabeled.*

Note that under the transductive setting, we have either of the following two cases:

- Feature spaces of the source and target domains are different (*i.e.*, $\mathcal{X}^T \neq \mathcal{X}^S$).
- Marginal distributions of the data from two domains are different (*i.e.*, $P(\mathbf{x}^T) \neq P(\mathbf{x}^S)$).

The second case is similar to the setting in *covariate shift* [42–45] in which the single task is concerned and the distributions of the training and testing data are different. Moreover, covariate shift falls in the concept of *sample selection bias* [46–48]. If the training and test tasks are the same (*i.e.*, $\mathcal{T}_{train} = \mathcal{T}_{test}$), sample selection bias becomes covariate shift [47]. The second case of transductive transfer is also known as *domain adaptation* in the field of natural language processing [49].

⁴There are different meanings of “transductive” in machine learning. The term “transductive” used in transfer learning means that there must be some unlabeled target data available for training, and the source and target tasks are required to be the same [12, 41]. In traditional machine learning, “transductive” learning usually means to learn the specific outputs of test data, which are used as unlabeled training data, rather than learn a general model which can infer the output of any newly coming data [5].

Many methods have been proposed for the second case of the transductive setting. Due to the change of data distributions from different domains, training with samples only from the source domain may degrade the classification performance in the target domain. To reduce the mismatch between two different domains, Huang *et al.* [47] proposed a two-step approach called Kernel Mean Matching (KMM). The first step is to diminish the mismatch between means of samples in RKHS from the two domains by re-weighting the samples $\phi(\mathbf{x}_i)$ in the source domain as $\beta_i\phi(\mathbf{x}_i)$, where β_i is learned by using the square of the Maximum Mean Discrepancy (MMD) criterion [37]. Then the second step is to learn a decision function $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ that separates patterns from two opposite classes in D using the loss function re-weighted by β_i . Schweikert *et al.* [31] proposed multiple KMM (Multi-KMM) by extending KMM for the problem where there exist multiple source domains. In Multi-KMM, the samples from each source domain are shifted towards the mean of the target samples by a certain amount of the distance without considering the label information. Dai *et al.* [50] proposed to use EM-based Naive Bayes classifiers [51] for transductive transfer learning. Wang *et al.* [52] proposed Transferred Discriminative Analysis (TDA) for dimensionality reduction in transfer learning. In TDA [53], they first clustered the unlabeled target data to generate their pseudo class labels and then used dimensionality reduction by using both the unlabeled target data and labeled source data. These two steps are iteratively processed to find a better discriminative subspace for the target data. Blitzer *et al.* [23] 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. In [54], Duan *et al.* proposed a method called Domain Adaptation Machine (DAM) with a data-dependent regularizer based on the smoothness assumption that the target classifier shares similar decision values with the pre-learned classifiers on the target unlabeled samples. Although they also used labeled training data in the formulation of DAM, it is easy to derive the solution of DAM for the transductive transfer setting. Bruzzone and Marconcini [13] proposed Domain Adaptation Support Vector Machine (DASVM) by learning the target classifier step by step. At each step, DASVM labels the unlabeled training samples from the target domain and simultaneously removes some source labeled samples that are unlikely to help learn a target classifier.

2.1.3 Unsupervised Transfer

In unsupervised transfer, we assume that the target task is different from the source task but they are related, which is similar to the assumption in inductive transfer. The difference between unsupervised and inductive transfer is that none of labeled data are available in either the source domain or the target domain under the unsupervised setting. Let us formally define the unsupervised transfer setting as follows:

Definition 2.6 *Unsupervised transfer is a setting of transfer learning, the goal of which is to help learn a target decision function $f^T(\cdot)$ by using the knowledge transferred from $\mathcal{D}^S$ and $\mathcal{T}^S$, where $\mathcal{T}^T \neq \mathcal{T}^S$ and there exist no labeled training data from either the source domain or the target domain.*

Straightforwardly, methods under this setting can be proposed for clustering [55, 56] and dimensionality reduction [57, 58] which are the classical unsupervised learning tasks in machine learning. However, little work has been done so far for unsupervised transfer learning problems. To cluster data under a transfer learning setting, Dai *et al.* [59] proposed self-taught clustering based on co-clustering [60]. For dimensionality reduction in unsupervised transfer, Pan *et al.* [61] proposed Maximum Mean Discrepancy Embedding (MMDE) to learn a kernel matrix for dimensionality reduction by minimizing the square of the MMD criterion [37]. They applied the learned kernel matrix, which induces a latent low-dimensional space, on the data to reduce the distribution mismatch between the source and target domains. However, when there are a large number of training data, MMDE [61] will suffer from heavy computational burden. To release such burden, Pan *et al.* [62] further proposed an efficient algorithm called Transfer Component Analysis (TCA).

2.2 Bounds in Transfer Learning

Some researchers have theoretically studied target error bounds for the transfer learning problems [63–69]. By assuming the distribution of the target domain be a weighted combination of the source distributions, Mansour *et al.* [68] proved the loss of the target classifier has an upper bound. Crammer *et al.* [67] 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.* [63] 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 samples from the source and target domains, which is measured by a so-called d_A distance as introduced in [70]. Following [63], Mansour *et al.* [69] extended the d_A distance [63, 70] 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.* [66] proposed uniform convergence bounds by additionally considering a limited number of labeled training data from the target domain. In the recent work [64], Ben-David *et al.* further analyzed three types of assumptions which are widely used in the existing domain adaptation methods, and they also mathematically discussed that under which assumptions a domain adaptation method can work.

2.3 Transfer Learning with Other Machine Learning Approaches

In recent years, transfer learning has also attracted intensive attention from other machine learning approaches, such as active learning, online learning, distance metric learning and so on.

2.3.1 Active Learning

Active learning techniques are proposed to alleviate the effort of data annotation [8–10]. After employing transfer learning to deal with the data distribution mismatch between the source and target domains, researchers are able to select representative samples from the target domain and label them to effectively train classifiers. Till now, some researchers have proposed methods which combine both transfer learning and active learning [71–75]. Rajan *et al.* [74] proposed an active learning approach for knowledge transfer to efficiently update existing classifiers by using as few labeled data points from a new

hyperspectral image [76] as possible. Liao *et al.* [73] proposed to actively select and label the unlabeled samples in a target domain by using data from source domains. By using the transfer learning method TrAdaBoost [20] and the standard SVM, Shi *et al.* [75] developed an active learning method to select important target samples.

2.3.2 Metric Learning

Some researchers solve transfer learning problems by learning metrics to bridge the relatedness between the source and target domains [16, 39, 77–79]. Zha *et al.* [78] learned a new distance metric for the target domain by using existing distance metrics which are pre-learned from the source domains. Saenko *et al.* [16] proposed to directly learn a distance metric across different domains based on the information-theoretic metric learning method [40]. Kulis *et al.* [39] extended Saenko *et al.*'s work [16] by incorporating non-linear kernels. It is worth mentioning that the work from Saenko *et al.* [16] and Kulis *et al.* [39] can be easily extended to solve transfer learning problems with heterogenous domains. Qi *et al.* [77] proposed to learn metrics to effectively mine the information shared between the training data from two image categories, and these metrics are later combined by their developed cross-category ensemble in order to learn target classifiers. Zhang and Yeung [79] proposed a multi-task metric learning method to learn the task relationships between all source tasks and the target task.

2.3.3 Online Learning

Online learning is known for its ability to continuously update the current classifier with the labeled training data coming in a sequence (*e.g.*, stock market prediction) [80, 81]. Thus far, few methods have addressed online learning under a transfer learning setting. One piece of pioneer work along this direction was done by Zhao and Hoi [82]. They proposed an online framework which handles transfer learning problems with either homogeneous domains or heterogenous domains. Dekel *et al.* [83] also proposed an online learning framework, which aims to learn multiple tasks in parallel under a global loss function.

2.4 Real-World Applications

In this section, we summarize quite a few pieces of existing transfer learning work according to a number of real-world transfer learning applications or fields as follows:

- Natural Language Processing: [18, 23, 29, 41, 84–89]
- Sentiment Classification: [19, 90–96]
- Text Classification: [20, 21, 50, 62, 97–106]
- Text Clustering: [59, 107, 108]
- Image Classification/Retrieval: [15, 16, 33, 38, 39, 109–116]
- Image Clustering: [22]
- Video Classification/Retrieval: [17, 22, 36, 54, 117–119]
- Collaborative Filtering: [120–124]
- Wireless Sensor Networks: [24, 61, 62, 125–130]
- Learning to Rank: [131–137]
- Bioinformatics: [31, 138–141]

Chapter 3

Domain Transfer Multiple Kernel Learning

In this chapter, we propose a novel transfer learning framework, referred to as Domain Transfer Multiple Kernel Learning (DTMKL), to cope with the considerable change between feature distributions of different domains. Our framework simultaneously learns a kernel function and a robust classifier by minimizing both the structural risk functional and the distribution mismatch between the labeled and unlabeled samples from the source and target domains. Under the DTMKL framework, we also propose two novel methods by using SVM and pre-learned classifiers, respectively. We apply these two methods for two transfer learning related applications (*i.e.*, video concept detection and text classification).

3.1 Introduction

The conventional machine learning methods usually assume that the training and test data are drawn from the same data distribution. In many applications, it is expensive and time-consuming to collect the labeled training samples. Meanwhile, classifiers trained with only a limited number of labeled samples are usually not robust for pattern recognition tasks. Recall that feature distributions of training samples from different domains change tremendously, and the training samples from multiple sources also have very different statistical properties (such as mean, intra-class and inter-class variance). Though a large number of training data are available in the source domain, the classifiers

trained from those data or the combined data from both the source and target domains may perform poorly on the test data from the target domain [30, 34].

A number of transfer learning methods [29, 30, 33–35] have been proposed to take advantage of all labeled samples from the both source and target domains. However, all these methods did not utilize the unlabeled samples in the target domain. Such unlabeled samples can also be used to improve the classification performance [7, 142].

When there are only a few or even no labeled samples available in the target domain, the source samples or the unlabeled target samples can be used to train the target classifier. Several transfer learning methods [44, 47] were also proposed to cope with the inconsistency of data distributions (such as covariate shift [44] or sampling selection bias [47]). These methods re-weighted the training samples from the source domain by using the unlabeled data from the target domain such that the statistics of samples from both domains are matched. Very recently, Bruzzone and Marconcini [13] proposed Domain Adaptation Support Vector Machine (DASVM), which extended Transductive SVM (T-SVM) to label unlabeled target samples progressively and simultaneously remove some source labeled samples. Interested readers may refer to [12] for the more complete survey of transfer learning methods.

The common observation is that most of these transfer learning methods are either variants of SVM or in tandem with SVM or other kernel methods. The prediction performances of these kernel methods heavily depend on the choice of the kernel. To obtain the optimal kernel, Lanckriet *et al.* [143] proposed to directly learn a nonparametric kernel matrix by solving an expensive semi-definite programming (SDP) problem [144]. However, the time complexity is $O(n^{6.5})$, which is computationally prohibitive for many real-world applications. Instead of learning the kernel matrix, many efficient Multiple Kernel Learning (MKL) methods [143, 145–147] have been proposed to directly learn the kernel function, in which the kernel function is assumed to be a linear combination of multiple predefined kernel functions (referred to as base kernel functions). And these methods simultaneously learn the decision function as well as the kernel. In practice, MKL has been successfully employed in many computer vision applications, such as action recognition [148, 149], object detection [150] and so on. However, these methods commonly assume that both training data and test data are drawn from the same domain. As a result, MKL methods cannot learn the optimal kernel with the combined

data from the source and target domains. Therefore, the training data from the source domain may degrade the performance of MKL algorithms in the target domain.

In this chapter, we propose a unified transfer kernel learning framework, referred to as Domain Transfer Multiple Kernel Learning (DTMKL), for several challenging transfer learning tasks. The main contributions of our work include:

- To deal with the considerable change between feature distributions of different domains, DTMKL minimizes the structural risk functional and Maximum Mean Discrepancy (MMD) [37], a criterion to evaluate the distribution mismatch between the source and target domains. In practice, DTMKL provides a unified framework to simultaneously learn an optimal kernel function as well as a robust classifier.
- Many existing kernel methods including SVM, Support Vector Regression (SVR), Kernel Regularized Least-Squares (KRLS) and so on, can be incorporated into the framework of DTMKL to tackle transfer learning problems. Moreover, we propose a reduced gradient descent procedure to efficiently and effectively learn the linear combination coefficients of multiple base kernels as well as the target classifier.
- Under the DTMKL framework, we propose two methods on the basis of SVM and pre-learned classifiers, respectively. The first method *DTMKL-AT* directly utilizes the training data from the source and target domain. The second method *DTMKL-f* makes use of the labeled target training data as well as the decision values from the existing base classifiers on the unlabeled data from the target domain. And these base classifiers can be pre-learned by using any method (*e.g.*, SVM and SVR).
- To the best of our knowledge, DTMKL is the first semi-supervised transfer kernel learning framework for the single source domain problem, which can incorporate many existing kernel methods. In contrast to the traditional kernel learning methods, DTMKL does not assume that the training and test data are drawn from the same domain.
- Comprehensive experiments on TRECVID, 20 Newsgroups, and email spam data sets demonstrate the effectiveness of the DTMKL framework in real-world applications.

The rest of this chapter is organized as follows: We briefly review the related work in Section 3.2. We then introduce our framework Domain Transfer Multiple Kernel Learning in Section 3.3. In particular, we present two methods DTMKL-AT and DTMKL- f to tackle the single source domain problem by using SVM and pre-learned classifiers, respectively. We experimentally compare the two proposed methods with other SVM-based transfer learning methods on the TRECVID data set for video concept detection, as well as on the 20 Newsgroups and email spam data sets for text categorization in Section 3.4. Finally, conclusive remarks are presented in Section 3.5.

3.2 Brief Review of Related Work

Let us denote the data set of labeled and unlabeled samples from the target domain as $\mathcal{D}_l^T = (\mathbf{x}_i^T, y_i^T)_{i=1}^{n_l}$ and $\mathcal{D}_u^T = \mathbf{x}_i^T_{i=n_l+1}^{n_l+n_u}$, respectively, where y_i^T is the label of $\mathbf{x}_i^T$. We also define $\mathcal{D}^T = \mathcal{D}_l^T \cup \mathcal{D}_u^T$ as the data set from the target domain with the size $n_T = n_l + n_u$ under the marginal data distribution $\mathcal{P}$, and $\mathcal{D}^S = (\mathbf{x}_i^S, y_i^S)_{i=1}^{n_S}$ as the data set from the source domain under the marginal data distribution $\mathcal{Q}$. Let us also represent the labeled training data set as $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1}^n$, where n is the total number of labeled samples. The labeled training data can be from the target domain (*i.e.*, $\mathcal{D} = \mathcal{D}_l^T$) or from the both domains (*i.e.*, $\mathcal{D} = \mathcal{D}_l^T \cup \mathcal{D}^S$).

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 $\mathbf{I}_n$ as the n -by- n identity matrix. $\mathbf{0}_n$ and $\mathbf{1}_n$ are n -by-1 vectors all zeros and ones, respectively. The inequality $\mathbf{u} = [u_1, \dots, u_j]' \geq \mathbf{0}_j$ means that $u_i \geq 0$ for $i = 1, \dots, j$. And the element-wise product between vectors $\mathbf{u}$ and $\mathbf{v}$ is represented as $\mathbf{u} \circ \mathbf{v} = [u_1 v_1, \dots, u_j v_j]'$. $\mathbf{A} \succ 0$ means that the matrix $\mathbf{A}$ is symmetric and positive definite (pd).

In the following subsections, we will briefly review two major paradigms of transfer learning. The first is to directly learn the decision function in the target domain (also known as *target classifier*) based on the labeled data in the source domain and/or the target domain by minimizing the mismatch of data distribution between two domains. The second is to make use of the existing *source classifiers* trained based on the source domain samples for transfer learning.

3.2.1 Domain Transfer via Reducing Mismatch between Data Distributions of Two Domains

In transfer learning, it is crucial to reduce the difference between the data distributions of the source 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 process is usually required. To avoid such a non-trivial task, Borgwardt *et al.* [37] 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 two domains in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ spanned by a kernel function k , namely:

$$\begin{aligned} \text{DIST}_k(\mathcal{D}^S, \mathcal{D}^T) &= \sup_{\|f\|_{\mathcal{H}} \leq 1} (\mathbb{E}_{\mathbf{x}^S \sim \mathcal{Q}}[f(\mathbf{x}^S)] - \mathbb{E}_{\mathbf{x}^T \sim \mathcal{P}}[f(\mathbf{x}^T)]) \\ &= \sup_{\|f\|_{\mathcal{H}} \leq 1} \langle f, (\mathbb{E}_{\mathbf{x}^S \sim \mathcal{Q}}[\phi(\mathbf{x}^S)] - \mathbb{E}_{\mathbf{x}^T \sim \mathcal{P}}[\phi(\mathbf{x}^T)]) \rangle_{\mathcal{H}} \\ &= \|\mathbb{E}_{\mathbf{x}^S \sim \mathcal{Q}}[\phi(\mathbf{x}^S)] - \mathbb{E}_{\mathbf{x}^T \sim \mathcal{P}}[\phi(\mathbf{x}^T)]\|_{\mathcal{H}}, \end{aligned} \quad (3.1)$$

where $\mathbb{E}_{\mathbf{x} \sim \mathcal{U}}[\cdot]$ denotes the expectation operator under the data distribution $\mathcal{U}$, and $f(\mathbf{x})$ is any function in $\mathcal{H}$. The second equality holds as $f(\mathbf{x}) = \langle f, \phi(\mathbf{x}) \rangle_{\mathcal{H}}$ by the property of RKHS [151], where $\phi(\cdot)$ is the nonlinear feature mapping function that induces the kernel k . Note that the inner product of $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ equals to the kernel function k (or $k(\cdot, \cdot)$) on $\mathbf{x}_i$ and $\mathbf{x}_j$, namely, $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j)$. Asymptotically, the empirical measure of MMD in (3.1) can be well-estimated by:

$$\text{DIST}_k(\mathcal{D}^S, \mathcal{D}^T) = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(\mathbf{x}_i^S) - \frac{1}{n_T} \sum_{i=1}^{n_T} \phi(\mathbf{x}_i^T) \right\|_{\mathcal{H}}. \quad (3.2)$$

To capture higher order statistics of the data (*e.g.*, higher order moments of probability distribution), the samples in (3.2) are transformed into a higher dimensional or even infinite dimensional space through the nonlinear feature mapping $\phi(\cdot)$. When $\text{DIST}_k(\mathcal{D}^S, \mathcal{D}^T)$ is close to zero, the higher order moments of the data from the two domains become matched, and so their data distributions are also close to each other [37]. The MMD criterion was successfully used to integrate biological data from multiple sources in [37].

Due to the change between data distributions of different domains, training with samples only from the source domain may degrade the classification performance in the target domain. To reduce the mismatch between two different domains, Huang *et al.* [47] proposed a two-step approach called Kernel Mean Matching (KMM). The first step is to diminish the mismatch between means of samples in RKHS from the two domains by re-weighting the samples $\phi(\mathbf{x}_i)$ in the source domain as $\beta_i \phi(\mathbf{x}_i)$, where β_i is learned by using the square of the MMD criterion in (3.2). Then the second step is to learn a decision function $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ that separates samples from two opposite classes in $\mathcal{D}$ using the loss function re-weighted by β_i .

Recently, Pan *et al.* [61] proposed an unsupervised kernel matrix learning method, referred to as Maximum Mean Discrepancy Embedding (MMDE), by minimizing the square of the MMD criterion in (3.2) as well, and then applied the learned kernel matrix to train an SVM classifier for WiFi localization and text categorization.

3.2.2 Domain Transfer via Existing Classifiers

Instead of learning the target classifier directly from the labeled data in both source and target domains, some researchers make use of the source classifiers trained from the source domain to learn the target classifier. Yang *et al.* [30] proposed Adaptive SVM (A-SVM), in which a new SVM classifier $f^T(\mathbf{x})$ is adapted from an existing source classifier $f^S(\mathbf{x})$ trained with the samples from the source domain¹. Specifically, the new decision function is formulated as $f^T(\mathbf{x}) = f^S(\mathbf{x}) + \Delta f(\mathbf{x})$, where the perturbation function $\Delta f(\mathbf{x})$ is learned by using the labeled data $\mathcal{D}_l^T$ from the target domain. As shown in [30], $f^S(\mathbf{x})$ can be deemed as a pattern-dependent bias, and then the perturbation function $\Delta f(\mathbf{x})$ can be easily learned.

Besides A-SVM, Schweikert *et al.* [31] proposed to use the linear combination of the decision values from the source SVM classifier and the target SVM classifier for the prediction in the target domain. It is noteworthy that both this method and A-SVM do not utilize the abundant and useful unlabeled data $\mathcal{D}_u^T$ in the target domain for transfer learning.

¹Yang *et al.* [30] also proposed a formulation to solve the multiple source domain problem. This paper mainly focuses on single source domain setting. We therefore briefly introduce their work under this setting.

3.3 Domain Transfer Multiple Kernel Learning Framework

In this section, we introduce our proposed unified transfer learning framework, referred to as Domain Transfer Multiple Kernel Learning (DTMKL). And we also present a unified learning algorithm for DTMKL. Based on the proposed framework, we further propose two methods using SVM and the existing classifiers, respectively.

In previous transfer learning methods [47, 61], the weights or the kernel matrix of samples are learned separately using the MMD criterion in (3.2) without considering any label information. However, it is usually beneficial to utilize label information during kernel learning. Instead of using the two-step approaches as in [47, 61], we propose a unified transfer learning framework, DTMKL, to learn the following decision function for the target domain:

$$f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b = \sum_{i=1}^n \beta_i k(\mathbf{x}_i, \mathbf{x}) + b, \quad (3.3)$$

as well as the kernel function k simultaneously, where $\mathbf{w}$ is the weight vector in the feature space and b is the bias term. Notice that β_i 's are the coefficients of the kernel expansion for the decision function $f(\mathbf{x})$ using the Representer Theorem [151]. In practice, DTMKL minimizes the mismatch between the data distributions of the source and target domains, as well as any structural risk functional of kernel methods. The learning framework of DTMKL is then formulated as:

$$[k, f] = \arg \min_{k, f} \Omega(\text{DIST}_k^2(\mathcal{D}^S, \mathcal{D}^T)) + \theta R(k, f, \mathcal{D}), \quad (3.4)$$

where $\Omega(\cdot)$ is any monotonic increasing function, and $\theta > 0$ is a tradeoff parameter to balance the mismatch between data distributions of two domains and the structural risk functional $R(k, f, \mathcal{D})$ defined on labeled samples in $\mathcal{D}$.

3.3.1 Minimizing data distribution mismatch

This first objective in DTMKL is to minimize the mismatch between data distributions of two domains using the MMD criterion defined in (3.2). We define a column vector $\mathbf{s}$ with $n_S + n_T$ entries, in which the first n_S entries are set as $1/n_S$ and the remaining

entries are set as $-1/n_T$, respectively. Let $\Phi = [\phi(\mathbf{x}_1^S), \dots, \phi(\mathbf{x}_{n_S}^S), \phi(\mathbf{x}_1^T), \dots, \phi(\mathbf{x}_{n_T}^T)]$ be the kernel matrix after feature mapping, and then $\frac{1}{n_S} \sum_{i=1}^{n_S} \phi(\mathbf{x}_i^S) - \frac{1}{n_T} \sum_{i=1}^{n_T} \phi(\mathbf{x}_i^T)$ in (3.2) is simplified as $\Phi \mathbf{s}$. Thus, the criterion in (3.2) can be rewritten as:

$$\text{DIST}_k^2(\mathcal{D}^S, \mathcal{D}^T) = \|\Phi \mathbf{s}\|^2 = \text{tr}(\Phi' \Phi \mathbf{S}) = \text{tr}(\mathbf{K} \mathbf{S}), \quad (3.5)$$

where $\mathbf{S} = \mathbf{s} \mathbf{s}' \in \mathbb{R}^{(n_S+n_T) \times (n_S+n_T)}$, $\mathbf{K} = \Phi' \Phi = \begin{bmatrix} \mathbf{K}^{A,A} & \mathbf{K}^{A,T} \\ \mathbf{K}^{T,A} & \mathbf{K}^{T,T} \end{bmatrix} \in \mathbb{R}^{(n_S+n_T) \times (n_S+n_T)}$, $\mathbf{K}^{A,A} \in \mathbb{R}^{n_S \times n_S}$, $\mathbf{K}^{T,T} \in \mathbb{R}^{n_T \times n_T}$ and $\mathbf{K}^{A,T} \in \mathbb{R}^{n_S \times n_T}$ are the kernel matrices defined for the source domain, the target domain and the transfer from the source domain to the target domain, respectively.

3.3.2 Minimizing structural risk functional

The second objective in DTMKL is to minimize the structural risk functional $R(k, f, \mathcal{D})$ defined on the labeled samples in $\mathcal{D}$. Note that the structural risk functional of many existing kernel methods, including SVM, SVR, KRLS and so on, can be used here. Without using the first term in (3.4), the resultant optimization problem becomes a standard kernel learning problem [143] to learn the kernel k and the decision function f for the corresponding kernel method.

3.3.3 Multiple base kernels

Instead of learning a nonparametric kernel matrix $\mathbf{K}$ in (3.4) for transfer learning as in [61], following [143, 146, 147], we assume the kernel k is a linear combination of a set of base kernels k_m 's, namely,

$$k = \sum_{m=1}^M d_m k_m,$$

where $d_m \geq 0$, $\sum_{m=1}^M d_m = 1$. We further assume the first objective $\Omega(\text{tr}(\mathbf{K} \mathbf{S}))$ in (3.4) is as follows²:

$$\Omega(\text{tr}(\mathbf{K} \mathbf{S})) = \frac{1}{2} (\text{tr}(\mathbf{K} \mathbf{S}))^2 = \frac{1}{2} \left(\text{tr} \left(\sum_{m=1}^M d_m \mathbf{K}_m \mathbf{S} \right) \right)^2 = \frac{1}{2} \mathbf{d}' \mathbf{h} \mathbf{h}' \mathbf{d},$$

²Note that any monotonic increasing function can be used to formulate $\Omega(\cdot)$. Because of the good convexity and the simplicity of quadratic functions, in this work we assume that $\Omega(\cdot)$ is a quadratic function in the form of $\Omega(z) = \frac{1}{2} z^2$.

where $\mathbf{h} = [h_1, \dots, h_M]'$, $h_m = \text{tr}(\mathbf{K}_m \mathbf{S})$, $\mathbf{K}_m = [k_m(\mathbf{x}_i, \mathbf{x}_j)] \in \mathbb{R}^{(n_S+n_T) \times (n_S+n_T)}$ and $\mathbf{d} = [d_1, \dots, d_M]'$. Moreover, from (3.3), we have $f(\mathbf{x}) = \sum_{m=1}^M d_m \mathbf{w}_m' \phi_m(\mathbf{x}) + b$, where $\mathbf{w}_m = \sum_{i=1}^n \beta_i \phi_m(\mathbf{x}_i)$.

Thus, the optimization problem in (3.4) can be rewritten as:

$$\min_{\mathbf{d} \in \mathcal{M}} \min_f \frac{1}{2} \mathbf{d}' \mathbf{h} \mathbf{h}' \mathbf{d} + \theta R(\mathbf{d}, f, \mathcal{D}), \quad (3.6)$$

where $\mathcal{M} = \{\mathbf{d} | \mathbf{d} \geq \mathbf{0}_M, \mathbf{1}_M' \mathbf{d} = 1\}$ is the feasible set of $\mathbf{d}$ and f is the target decision function. Note that we have only M variables in $\mathbf{d}$, which is much smaller than the total number of variables $(n_S + n_T)^2$ in $\mathbf{K}$. Thus, the resultant optimization problem is much simpler than that of the non-parametric kernel matrix learning in MMDE [61].

3.3.4 Learning algorithm

Let us define

$$J(\mathbf{d}) = \min_f R(\mathbf{d}, f, \mathcal{D}). \quad (3.7)$$

Then, the optimization problem (3.6) can be rewritten as:

$$\min_{\mathbf{d} \in \mathcal{M}} h(\mathbf{d}) = \min_{\mathbf{d} \in \mathcal{M}} \frac{1}{2} \mathbf{d}' \mathbf{h} \mathbf{h}' \mathbf{d} + \theta J(\mathbf{d}). \quad (3.8)$$

It is worth mentioning that the traditional MKL methods suffer from the non-smooth problem on the linear kernel combination coefficient $\mathbf{d}$, and thus the simple coordinate descent algorithms such as SMO may not lead to the global solution [145]. As shown in the literature, the global optimum of MKL can be achieved by using the reduced gradient descent method [147] or semi-infinite linear programming [146, 152]. Following [147], we develop an efficient and effective reduced gradient descent procedure to iteratively update different variables (*e.g.*, $\mathbf{d}$ and f) in (3.6) to obtain the optimal solution. The algorithm is detailed as follows:

Updating the decision function f : With the fixed $\mathbf{d}$, only the structural risk functional $R(\mathbf{d}, f, \mathcal{D})$ in (3.6) depends on f . We can solve for the decision function f by minimizing $R(\mathbf{d}, f, \mathcal{D})$.

Updating kernel coefficients $\mathbf{d}$: When the decision function f is fixed, (3.8) can be updated using the reduced gradient descent method as suggested in [147]. Specifically, the gradient of h in (3.8) is

$$\nabla h = \mathbf{h} \mathbf{h}' \mathbf{d} + \theta \nabla J,$$

where ∇J is the gradient of J in (3.7). Furthermore, the hessian matrix can be derived as

$$\nabla^2 h = \mathbf{h}\mathbf{h}' + \theta \nabla^2 J.$$

Note that $\mathbf{h}\mathbf{h}' + \theta \nabla^2 J$ may not be full rank. Thus, to avoid numerical instability, we replace $\mathbf{h}\mathbf{h}'$ by $\mathbf{h}\mathbf{h}' + \varepsilon \mathbf{I}$ to make sure $\nabla^2 h = \mathbf{h}\mathbf{h}' + \varepsilon \mathbf{I} + \theta \nabla^2 J \succ 0$, where ε is set to 10^{-2} in the experiments. Compared with first-order gradient based methods, second-order derivative based methods usually converge faster. So we use $\mathbf{g} = (\nabla^2 h)^{-1} \nabla h$ as the updating direction. To maintain $\mathbf{d} \in \mathcal{M}$, the updating direction $\mathbf{g}$ is reduced as in [147], so the updated weight of multiple base kernels is as follows:

$$\mathbf{d}_{t+1} = \mathbf{d}_t - \eta_t \mathbf{g}_t \in \mathcal{M}, \quad (3.9)$$

where $\mathbf{d}_t$ and $\mathbf{g}_t$ are the linear combination coefficient vector $\mathbf{d}$ and the reduced updating direction $\mathbf{g}$ at the t th iteration respectively, and η_t is the learning rate. The overall procedure of the proposed DTMKL is summarized in Algorithm 1.

Algorithm 1 Optimization procedures of DTMKL

- 1: Initialize $\mathbf{d} = \frac{1}{M} \mathbf{1}_M$.
 - 2: For $t = 1, \dots, T_{\max}$
 - 3: Solve the target classifier f in the objective function in (3.7).
 - 4: Update the linear combination coefficient vector $\mathbf{d}$ of multiple base kernels using (3.9).
 - 5: End.
-

As aforementioned, one can employ any structural risk functional of kernel methods in the learning framework of DTMKL. In this chapter, we first propose to model the structural risk functional $R(k, f, \mathcal{D})$ by using SVM. Second, inspired by the utilization of source classifiers for transfer learning as in A-SVM [30], we also propose another formulation which considers the decision values from the base classifiers on the unlabeled samples in the target domain.

3.3.5 DTMKL using Hinge Loss

We use the structural risk functional of SVM with the hinge loss function (*i.e.*, $\ell_h(z) = \max(0, 1 - z)$) to model the second objective $R(\mathbf{d}, f, \mathcal{D})$ in (3.6). Here, instead of using

$\frac{1}{2}\|\mathbf{w}\|^2$ in the standard SVM, we use the MKL regularizer $\frac{1}{2}\sum_{m=1}^M d_m\|\mathbf{w}_m\|^2$ introduced in [152]. Then, the corresponding constrained optimization problem can be rewritten as:

$$\min_{\mathbf{d} \in \mathcal{M}} \min_{\mathbf{w}_m, b, \xi_i} \quad \frac{1}{2}\mathbf{d}'\mathbf{h}\mathbf{h}'\mathbf{d} + \theta \left(\frac{1}{2} \sum_{m=1}^M d_m \|\mathbf{w}_m\|^2 + C \sum_{i=1}^n \xi_i \right), \quad (3.10)$$

$$\text{s.t.} \quad y_i \left(\sum_{m=1}^M d_m \mathbf{w}_m' \phi_m(\mathbf{x}_i) + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n, \quad (3.11)$$

where $C > 0$ is the tradeoff parameter and ξ_i 's are the slack variables for the corresponding constraints. However, (3.10) in general is non-convex due to the product of d_m and $\mathbf{w}_m$ in the inequality constraints of (3.10). Following [152], we introduce a transformation $\mathbf{v}_m = d_m \mathbf{w}_m$, and (3.10) can be then rewritten as:

$$\min_{\mathbf{d} \in \mathcal{M}} \min_{\mathbf{v}_m, b, \xi_i} \quad \frac{1}{2}\mathbf{d}'\mathbf{h}\mathbf{h}'\mathbf{d} + \underbrace{\theta \left(\frac{1}{2} \sum_{m=1}^M \frac{\|\mathbf{v}_m\|^2}{d_m} + C \sum_{i=1}^n \xi_i \right)}_{J(\mathbf{d})}, \quad (3.12)$$

$$\text{s.t.} \quad y_i \left(\sum_{m=1}^M \mathbf{v}_m' \phi_m(\mathbf{x}_i) + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n. \quad (3.13)$$

In the following theorem, we prove that the optimization problem (3.12) is convex.

Theorem 3.1 *The optimization problem (3.12) is jointly convex with respect to $\mathbf{d}$, $\mathbf{v}_m$, b and ξ_i .*

Proof: The first term $\frac{1}{2}\mathbf{d}'\mathbf{h}\mathbf{h}'\mathbf{d}$ in the objective function (3.12) is a convex quadratic term. Other terms in the objective function and constraints are linear except the term $\frac{1}{2}\sum_{m=1}^M \frac{\|\mathbf{v}_m\|^2}{d_m}$ in (3.12). As shown in [147], this term is also jointly convex with respect to $\mathbf{d}$ and $\mathbf{v}_m$. Therefore, the optimization problem in (3.12) is jointly convex with respect to $\mathbf{d}$, $\mathbf{v}_m$, b and ξ_i .

Therefore, (3.12) can converge to the global minimum using the reduced gradient descent procedure described in Algorithm 1. Note that when one of the linear combination coefficients (say, d_m) is zero, the corresponding $\mathbf{v}_m$ at the optimality must be zero as well [147]. In other cases (*i.e.*, the corresponding $\mathbf{v}_m$ is nonzero), the corresponding descent direction is nonzero, and so d_m will be updated again by using the reduced descent direction in the subsequent iteration until the objective function in (3.12) cannot be decreased.

Recall that the constrained optimization problem of SVM is usually solved by its dual problem, which is in the form of a quadratic programming (QP) problem:

$$\max_{\boldsymbol{\alpha} \in \mathcal{A}} \mathbf{1}'_n \boldsymbol{\alpha} - \frac{1}{2} (\boldsymbol{\alpha} \circ \mathbf{y})' \mathbf{K} (\boldsymbol{\alpha} \circ \mathbf{y}).$$

Similarly, one can show that $J(\mathbf{d})$ in (3.12) can be written as follows [152]:

$$J(\mathbf{d}) = \max_{\boldsymbol{\alpha} \in \mathcal{A}} \mathbf{1}'_n \boldsymbol{\alpha} - \frac{1}{2} (\boldsymbol{\alpha} \circ \mathbf{y})' \left(\sum_{m=1}^M d_m \mathbf{K}_m \right) (\boldsymbol{\alpha} \circ \mathbf{y}), \quad (3.14)$$

where $J(\mathbf{d})$ is linear in $\mathbf{d} \in \mathcal{M}$, $\mathcal{A} = \{\boldsymbol{\alpha} | \boldsymbol{\alpha}' \mathbf{y} = 0, \mathbf{0}_n \leq \boldsymbol{\alpha} \leq C \mathbf{1}_n\}$ is the feasible set of the dual variables $\boldsymbol{\alpha}$, $\mathbf{y} = [y_1, \dots, y_n]'$ is the label vector and $\mathbf{K}_m = [k_m(\mathbf{x}_i, \mathbf{x}_j)] = [\phi_m(\mathbf{x}_i)' \phi_m(\mathbf{x}_j)] \in \mathbb{R}^{n \times n}$ is the m -th base kernel matrix of the labeled samples.

With the optimal $\mathbf{d}$ and the dual variables $\boldsymbol{\alpha}$, the prediction of any test data $\mathbf{x}$ using the target decision function can be obtained as follows:

$$f^T(\mathbf{x}) = \sum_{m=1}^M d_m \mathbf{w}'_m \phi_m(\mathbf{x}) + b = \sum_{i: \alpha_i \neq 0} \alpha_i y_i \sum_{m=1}^M d_m k_m(\mathbf{x}_i, \mathbf{x}) + b.$$

In this method, the labeled samples from the **A**uxiliary domain (*i.e.*, source domain) and the **T**arget domain can be directly used to improve the classification performance of the classifier in the target domain. In this case, we term this method as *DTMKL-AT*. It is worth mentioning that the unlabeled target data $\mathcal{D}_u^T$ can be used for the calculation of the MMD values in (3.2) which does not require label information.

3.3.6 DTMKL using Existing Base Classifiers

In this subsection, we extend our proposed DTMKL by defining the structural risk functional of SVR on both labeled and unlabeled data in the target domain. There are no input labels for the unlabeled target samples. Inspired by the use of base classifiers, we introduce a regularization term (*i.e.*, the last term in (3.15)) to enforce that the decision values from the target classifier and the existing base classifiers are similar on the unlabeled target samples. Moreover, we further introduce another penalty term (*i.e.*, the fourth term in (3.15)) for the labeled target samples to ensure that the decision values from the target classifier are close to the true labels. Note that the labeled training

data can be from the target domain (*i.e.*, $\mathcal{D} = \mathcal{D}_l^T$) or from the both domains (*i.e.*, $\mathcal{D} = \mathcal{D}_l^T \cup \mathcal{D}^S$). Let us denote $f^{T,m}$ and $f^{B,m}$ as the target classifier and the base classifier with the m th base kernel, respectively. For simplicity, we define $f_i^{T,m}$ and $f_i^{B,m}$ as the decision values on any data $\mathbf{x}_i$, respectively. Similar to (3.10), we also assume that the regularizer is $\frac{1}{2} \sum_{m=1}^M d_m \|\mathbf{w}_m\|^2$. Then, we present another formulation of DTMKL as follows:

$$\begin{aligned}
 \min_{\substack{\mathbf{d} \in \mathcal{M}, \mathbf{w}_m, b, \xi_i, \\ \xi_i^*, \mathbf{f}_l^{T,m}, \mathbf{f}_u^{T,m}}} \quad & \frac{1}{2} \mathbf{d}' \mathbf{h} \mathbf{h}' \mathbf{d} + \theta \left\{ \frac{1}{2} \sum_{m=1}^M d_m \|\mathbf{w}_m\|^2 + C \sum_{i=1}^{n+n_u} (\xi_i + \xi_i^*) \right. \\
 & \left. + \frac{\zeta}{2} \left(\sum_{m=1}^M \left\| \mathbf{f}_l^{T,m} - \mathbf{y} \right\|^2 + \lambda \sum_{m=1}^M \left\| \mathbf{f}_u^{T,m} - \mathbf{f}_u^{B,m} \right\|^2 \right) \right\}, \quad (3.15) \\
 \text{s.t.} \quad & \sum_{m=1}^M d_m \mathbf{w}_m' \phi_m(\mathbf{x}_i) + b - \sum_{m=1}^M d_m f_i^{T,m} \leq \epsilon + \xi_i, \xi_i \geq 0, \\
 & \sum_{m=1}^M d_m f_i^{T,m} - \sum_{m=1}^M d_m \mathbf{w}_m' \phi_m(\mathbf{x}_i) - b \leq \epsilon + \xi_i^*, \xi_i^* \geq 0, \\
 & i = 1, \dots, n + n_u.
 \end{aligned}$$

where $\lambda > 0$ is the balance parameter, $C, \zeta > 0$ are the tradeoff parameters, $\mathbf{y} = [y_1, \dots, y_n]'$ is the label vector of the labeled training data from $\mathcal{D}$, ξ_i 's and ξ_i^* 's are slack variables for the ϵ -insensitive loss, $\mathbf{f}_l^{T,m} = [f_1^{T,m}, \dots, f_n^{T,m}]'$ is the decision value vector of the labeled training data $\mathcal{D}$ from the target classifier, and $\mathbf{f}_u^{T,m} = [f_{n+1}^{T,m}, \dots, f_{n+n_u}^{T,m}]'$ and $\mathbf{f}_u^{B,m} = [f_{n+1}^{B,m}, \dots, f_{n+n_u}^{B,m}]'$ are the decision value vectors of the unlabeled target data $\mathcal{D}_u^T$ from the target classifier $f^{T,m}$ and the base classifier $f^{B,m}$, respectively. While the objective function in (3.15) is not jointly convex with respect to the variables d_m and $\mathbf{w}_m$, our iterative approach listed in Algorithm 1 can still reach the local minimum.

We denote the objective inside $\{\}$ of (3.15) as $J(\mathbf{d})$. The dual of $J(\mathbf{d})$ (see Appendix A.1 for the detailed derivation) can be derived by introducing the Lagrangian multipliers $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$:

$$J(\mathbf{d}) = \max_{(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) \in \mathcal{A}} -\frac{1}{2} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)' \tilde{\mathbf{K}} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) - \tilde{\mathbf{y}}' (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) - \epsilon \mathbf{1}_{n+n_u}' (\boldsymbol{\alpha} + \boldsymbol{\alpha}^*), \quad (3.16)$$

Figure 3.1: Illustration of virtual labels. The base classifier $f^{B,m}$ is learned with the base kernel function k_m and the labeled training data from $\mathcal{D}$, where $m = 1, \dots, M$. For each of the unlabeled target pattern $\mathbf{x}$ from $\mathcal{D}_u^T$, we can obtain its decision value $f^{B,m}(\mathbf{x})$ from each base classifier. Then the virtual label $\tilde{y}$ of $\mathbf{x}$ is defined as the linear combination of its decision values $f^{B,m}(\mathbf{x})$'s weighted by the coefficients d_m 's, *i.e.*, $\tilde{y} = \sum_{m=1}^M d_m f^{B,m}(\mathbf{x})$.

where

$$\tilde{\mathbf{K}} = \sum_{m=1}^M d_m \tilde{\mathbf{K}}_m = \sum_{m=1}^M d_m \mathbf{K}_m + \frac{1}{\zeta} \sum_{m=1}^M d_m^2 \begin{bmatrix} \mathbf{I}_n & \\ & \frac{1}{\lambda} \mathbf{I}_{n_u} \end{bmatrix}, \quad (3.17)$$

$$\tilde{\mathbf{y}} = \sum_{m=1}^M d_m \tilde{\mathbf{y}}_m = \begin{bmatrix} \mathbf{y} \\ \sum_{m=1}^M d_m \mathbf{f}_u^{B,m} \end{bmatrix}, \quad (3.18)$$

$\mathcal{A} = \{(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) | \boldsymbol{\alpha}' \mathbf{1}_{n+n_u} = \boldsymbol{\alpha}^{*'} \mathbf{1}_{n+n_u}, \mathbf{0}_{n+n_u} \leq \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \leq C \mathbf{1}_{n+n_u}\}$ is the feasible set of the dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$, and $\mathbf{K}_m = [k_m(\mathbf{x}_i, \mathbf{x}_j)] \in \mathbb{R}^{(n+n_u) \times (n+n_u)}$ is the kernel matrix of both the labeled samples from $\mathcal{D}$ and unlabeled samples from $\mathcal{D}_u^T$.

Recall that the dual form of the standard ϵ -SVR is as follows:

$$\max_{(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) \in \mathcal{A}} -\frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)' \mathbf{K} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) - \mathbf{y}'(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) - \epsilon \mathbf{1}'_{n+n_u} (\boldsymbol{\alpha} + \boldsymbol{\alpha}^*). \quad (3.19)$$

Surprisingly, (3.16) is very similar to (3.19) except for some minor changes, that is the kernel matrix $\mathbf{K}$ and $\mathbf{y}$ are replaced by $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{y}}$, respectively. Therefore, (3.16) can be efficiently solved by using the state-of-the-art SVM solver (*e.g.*, LIBSVM [153]). The kernel matrix $\tilde{\mathbf{K}}$ is similar to Automatic Relevance Determination (ARD) kernel used

in Gaussian Process, and the second term in (3.17) is to control the noise of output. Interestingly, each of the last n_u entries of $\tilde{\mathbf{y}}$ in (3.18) can be considered as a so-called virtual label $\tilde{y} = \sum_{m=1}^M d_m f^{B,m}(\mathbf{x})$ composed by the linear combination of the decision values from the base classifiers $f^{B,m}$'s on the unlabeled target pattern $\mathbf{x}$ (see Figure 3.1 for illustration).

With the optimal $\mathbf{d}$ and the dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$, the target decision function can be found as:

$$f(\mathbf{x}) = \sum_{m=1}^M d_m \mathbf{w}'_m \phi_m(\mathbf{x}) + b = \sum_{i: \alpha_i - \alpha_i^* \neq 0} (\alpha_i - \alpha_i^*) \sum_{m=1}^M d_m k_m(\mathbf{x}_i, \mathbf{x}) + b.$$

Because of the use of the existing base classification functions, we then refer to this method as *DTMKL_{-f}*.

3.3.7 Computational Complexity of DTMKL

Recall that DTMKL adopts the reduced gradient descent scheme as in [147] to iteratively update the coefficients of base kernels and learn the target classifier. For DTMKL_{-AT}, the overall optimization procedure is dominated by a series of the kernel classifier training³. For example, at each iteration of DTMKL_{-AT}, the cost is essentially the same as the SVM training. Empirically, the SVM training complexity is $O(n^{2.3})$ [154]. And so the training cost for our proposed DTMKL_{-AT} is $O(T_{\max} \times n^{2.3})$, where $T_{\max}$ is the number of iterations in DTMKL. As shown in Section 3.4.5, our DTMKL_{-AT} generally converges after less than five iterations. For DTMKL_{-f}, we use multiple base classifiers. For example, the base classifiers SVM_{-AT} can be pre-learned and adapted from the existing classifier SVM_{-A} at very little computational cost by using warm start strategy or using A-SVM. Thus, the cost of the calculation of the virtual labels for DTMKL_{-f} is not significant. Recall that DTMKL_{-f} incorporates both labeled and unlabeled samples in the training stage. Therefore, the training complexity of DTMKL_{-f} is $O(T_{\max} \times (n + n_u)^{2.3})$.

The testing complexity of DTMKL_{-AT} and DTMKL_{-f} depends on the number of support vectors learned from the training stage. And we show in Table 3.3 that our

³Here, we suppose multiple base kernels can be precomputed and loaded into memory before the DTMKL training. Then the computational cost for the calculation of the learned kernel $\mathbf{K} = \sum_{m=1}^M d_m \mathbf{K}_m$ which takes $O(Mn^2)$ time can be ignored.

methods DTMKL-AT and DTMKL- f take less than one minute to finish the whole prediction process for about 21,213 test samples from each of 36 concepts on the TRECVID data set, which are as fast as the MKL algorithm.

3.3.8 Discussions with Related Work

Our work is different from the prior transfer learning methods such as [13, 29, 30, 33, 34, 47]. 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 are limited, the cross-validation approach may not choose the optimal kernel, which significantly degrades the generalization performance of SVM. Moreover, most existing transfer learning algorithms [29, 30, 33, 34] do not explicitly consider any specific criterion to measure the distribution mismatch of samples between different domains. As demonstrated in the previous work [15, 30, 31, 54], the source classifiers (*i.e.*, the base classifiers trained with the data from one or multiple source domains) can be used to learn a robust target classifier. Again, there is no specific criterion used to minimize the distribution mismatch between the source and target domains in these methods. In addition, the work in [54] focuses on the setting with *multiple source domains* and the Domain Adaptation Machine (DAM) algorithm was specifically proposed for multiple source domain adaptation problem. The algorithm Cross-Domain Regularized Regression (CDRR) and its Incremental version Incremental CDRR (ICDRR) in [15] were specifically designed for large scale image retrieval applications. In order to achieve the real-time retrieval performance on the large image data set with about 270,000 images, a linear regression function is used as the target function in [15]. Also, in the previous work [15, 30, 31, 54], only one kernel is used in the target decision function. In contrast to these methods [15, 29–31, 33, 34, 54], DTMKL is a unified transfer kernel learning framework, in which the optimal kernel is learned by explicitly minimizing the distribution mismatch between the source and target domains by using both labeled and unlabeled samples. Most importantly, many kernel learning methods (*e.g.*, SVM, SVR, KRLS and etc) can be readily embedded into our DTMKL framework to solve transfer learning problems.

The work most closely related to DTMKL was proposed by Pan *et al.* [61], in which a two-step approach is used for transfer 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 an SVM classifier. DTMKL is different from [61] in the following aspects:

- 1) A kernel matrix is learned in unsupervised setting in [61] without using any label information, which is not as effective as our semi-supervised learning method DTMKL.
- 2) In contrast to the two-step approach in [61], DTMKL simultaneously learns a kernel function and SVM classifier.
- 3) The learned kernel matrix in [61] is nonparametric, thus it cannot be applied to unseen data. Instead, DTMKL can handle any new test data.
- 4) The optimization problem in [61] is in the form of expensive semi-definite programming (SDP) [155], the time complexity of which is $O(n^{6.5})$. As a result, it can only handle several hundred samples. Therefore, it cannot be applied to medium or large scale applications such as video concept detection.

Another related work is Adaptive Multiple Kernel Learning (A-MKL) [17], in which the target classifier is constrained as the linear combination of a set of pre-learned classifiers and the perturbation function learned by multiple kernel learning. A-MKL can be considered as an extension of DTMKL-AT. In A-MKL, the unlabeled target samples are only used to measure the distribution mismatch between the two domains in the Maximum Mean Discrepancy (MMD) criterion, which is similar as in DTMKL-AT and DTMKL- f . In contrast, in DTMKL- f , the decision values from the pre-learned base classifiers on the unlabeled target samples are used as virtual labels in a new regularizer (*i.e.*, the last term in (3.15)) in order to enforce that the decision values from the target classifier and the existing base classifiers are similar on the unlabeled target samples. Moreover, A-MKL classifier can also be used as one base classifier in DTMKL- f .

Multiple Kernel Learning (MKL) methods [143, 146, 147] also simultaneously learn the decision function and the kernel in an inductive setting. However, the default assumption of MKL is that the training data and the test data are drawn from the same

domain. When the training data and the test data come from different distributions, MKL methods cannot learn the optimal kernel with the combined training data from the source and target domains. Therefore, the training data from the source domain may degrade the classification performances of MKL algorithms in the target domain. In contrast, DTMKL can utilize the samples from both domains for better classification performances.

3.4 Experiments

In this section, we evaluate our methods DTMKL-AT and DTMKL- f for two transfer learning related applications: 1) video concept detection on the challenging TRECVID video corpus and 2) text categorization on the 20 Newsgroups data set and the email spam data set.

3.4.1 Descriptions of Data Sets and Features

3.4.1.1 TRECVID Data Set

The TRECVID video corpus⁴ is one of the largest annotated video benchmark data sets for research purposes. The TRECVID 2005 data set contains 61,901 keyframes extracted from 108 hours of video programs from six broadcast channels (in English, Arabic and Chinese), and the TRECVID 2007 data set contains 21,532 keyframes extracted from 60 hours of news magazine, science news, documentaries and educational programming videos. As shown in [34], TRECVID data sets are challenging for transfer learning methods due to the large difference between TRECVID 2007 data set and TRECVID 2005 data set in terms of program structure and production values. 36 semantic concepts are chosen from the LSCOM-lite lexicon [156], 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 keyframes in both TRECVID 2005 and 2007 data sets.

⁴<http://www-nlpir.nist.gov/projects/trecvid>

In this work, we focus on the single source domain and single target domain setting. To evaluate the performances of all the methods, we choose one Chinese channel *CCTV4* from TRECVID 2005 data set as the source domain, and use the TRECVID 2007 data set as the target domain. The source data set $\mathcal{D}^S$ consists of all the labeled samples from the source domain (*i.e.*, 10,896 keyframes in *CCTV4* channel). We randomly select 10 positive samples per concept from the TRECVID 2007 data set (containing 21,532 keyframes in total) as the labeled target training data set $\mathcal{D}_l^T$, which leads to a total of 355 training samples from the target domain⁵. Considering that it is computationally prohibitive to compare all the methods over multiple random training and testing splits, we report results from one split. And the remaining samples from the target domain are used for testing. For each of the 36 concepts, we have 21,213 test samples on average⁶.

Three low-level global features Grid Color Moment (225 dim.), Gabor Texture (48 dim.) and Edge Direction Histogram (73 dim.) are extracted to represent the diverse content of keyframes, because of their consistent good performances reported in TRECVID [30, 34]. Moreover, the three type of global features can be efficiently extracted, and the previous work [30, 34] also shows that the transfer issue exists when using these global features. Yanagawa *et al.* have made the three types of features extracted from the keyframes of TRECVID data sets publicly available (see [157] for more details). We further concatenate the three types of features to form a 346-dimensional feature vector for each keyframe.

3.4.1.2 20 Newsgroups Data Set

The 20 Newsgroups data set⁷ is a collection of 18,774 news documents. This data set is organized in a hierarchical structure which consists of six main categories and 20 subcategories. Some of the subcategories (from the same category) are related to each

⁵A large portion of keyframes in TRECVID 2007 data set have multiple labels. We therefore only have 355 unique labeled target training samples. For each concept, we make sure that there are only 10 positive samples from the target domain when training one-versus-all classifiers. It is worth noting that for some concepts (*e.g.*, “Person”), we have fewer than 345 negative samples for model learning after excluding some training samples that are selected from other non-“Person” concepts but also positively labeled as “Person”.

⁶Because of the multi-label setting of the TRECVID data set (*i.e.*, one sample may be positive in multiple concepts), we have at least 21,172 test examples for each of the 36 concepts. As a matter of fact, on the average, each concept has 21,213 test samples.

⁷<http://people.csail.mit.edu/jrennie/20Newsgroups>

other while others (from different categories) are not related, making this data set suitable to evaluate transfer learning algorithms.

In the experiments, four largest main categories (*i.e.*, “comp”, “rec”, “sci” and “talk”) are chosen for evaluation. Specifically, for each main category, the largest subcategory is selected as the target domain, while the second largest subcategory is chosen as the source domain. Moreover, we consider the largest category “comp” as the positive class and one of the three other categories as the negative class for each setting. Table 3.1 provides the detailed information of all three settings. To construct the training data set, we use all labeled samples from the source domain, as well as randomly choose m positive and m negative samples from the target domain. And the remaining samples in the target domain are considered as the test data which are also used as the unlabeled data for training. In the experiments, m is set as 0, 1, 3, 5, 7 and 10. For any given m , we randomly sample the training data from the target domain for 5 times, and report the means and the standard deviations of all methods. Moreover, the word-frequency feature is used to represent each document.

3.4.1.3 Email Spam Data Set

There are three email subsets (denoted by User1, User2 and User3, respectively) annotated by three different users in the email spam data set⁸. The task is to classify spam and non-spam emails. Since the spam and non-spam emails in the subsets have been differentiated by different users, the data distributions of the three subsets are related but different. Each subset has 2,500 emails, in which one half of the emails are *non-spam* (labeled as 1) and the other half of them are *spam* (labeled as -1).

On this data set, we consider three settings: 1) User1 (source domain) & User2 (target domain); 2) User2 (source domain) & User3 (target domain) and 3) User3 (source domain) & User1 (target domain). For each setting, the training data set contains all labeled samples from the source domain as well as the labeled samples from the target domain, in which 5 positive and 5 negative samples are randomly chosen. And the remaining samples in the target domain are used as the unlabeled training data and the test data as well. We randomly sample the training data from the target domain for

⁸<http://www.ecmlpkdd2006.org/challenge.html>

Table 3.1: Description of the 20 Newsgroups data set.

(a) Source and target domains in different settings.

Setting	Source Domain	Target Domain
comp vs. rec	comp.windows.x & rec.sport.hockey	comp.sys.ibm.pc.hardware & rec.motorcycles
comp vs. sci	comp.windows.x & sci.crypt	comp.sys.ibm.pc.hardware & sci.med
comp vs. talk	comp.windows.x & talk.politics.mideast	comp.sys.ibm.pc.hardware & talk.politics.guns

(b) Numbers of training and test samples in different settings.

Setting	# positive/negative training samples from the source domain	# positive/negative samples from the target domain	
		Training	Testing
comp vs. rec	979/993	m/m	$982 - m/997 - m$
comp vs. sci	979/987	m/m	$982 - m/989 - m$
comp vs. talk	979/909	m/m	$982 - m/940 - m$

In the experiments, m is set as 0, 1, 3, 5, 7 and 10.

5 times and report the means and the standard deviations of all methods. Again, the word-frequency feature is used to represent each document.

3.4.2 Experimental Setup

We systematically compare our proposed methods DTMKL-AT and DTMKL- f with the baseline SVM, and other transfer learning algorithms including Feature Replication (FR) [29], Adaptive SVM (A-SVM) [30], Cross-Domain SVM (CD-SVM) [34] and Kernel Mean Matching (KMM) [47]. We also report the results of the Multiple Kernel Learning (MKL) algorithm, in which the optimal kernel combination coefficients are learned by only minimizing the second part of DTMKL-AT in (3.10) corresponding to the structural risk functional of SVM. Note that we do not compare with [61] because their work cannot cope with thousands of training and test samples.

For all methods, we train one-versus-all classifiers. Note that the standard SVM can use the labeled training data set $\mathcal{D}_l^T$ from the target domain, the labeled training data set $\mathcal{D}^S$ from the source domain, or the combined training data set $\mathcal{D}^S \cup \mathcal{D}_l^T$ from both source and target domains. We then refer to SVM in the above three cases as SVM-T, SVM-A and SVM-AT, respectively. We also report the results of MKL-AT by employing the combined training data from two domains. The transfer learning methods FR, A-SVM, CD-SVM and KMM also make use of the combined training data set $\mathcal{D}^S \cup \mathcal{D}_l^T$ for model learning.

MKL-AT and our DTMKL based methods can make use of multiple base kernels. For fair comparison, we use the same kernels for other methods including SVM-T, SVM-A, SVM-AT, FR, A-SVM, CD-SVM and KMM. Specifically for each method, we train multiple classifiers using the same kernels and then equally fuse the decision values to obtain the final prediction results.

Note that we make use of the unlabeled target training data from $\mathcal{D}_u^T$ in KMM and our DTMKL based methods. For KMM and DTMKL-AT, the labeled and unlabeled training data are employed to measure the data distribution mismatch between two domains using the MMD criterion in (3.2). We additionally make use of the virtual labels for DTMKL- f , which are the linear combination of the decision values from multiple base classifiers on

the unlabeled training data from $\mathcal{D}_u^T$. In this work, we employ SVM_AT from multiple base kernels as the base classifiers in DTMKL_ f .

With our experimental setting, cross validation is not suitable to automatically tune the optimal parameters for the target classifier, because we only have a limited number of labeled samples or even no labeled samples from the target domain. For the two text data sets, we vary the tradeoff parameter C for all methods and report the best result of each method with the optimal C , where $C = 0.1, 0.2, 0.5, 1, 2, 5, 10, 20$ and 50 . We fix the tradeoff parameter C as the default value 1 in LIBSVM for the large TRECVID data set, because it is time-consuming to run the experiments multiple times using different C .

3.4.2.1 Details on the TRECVID Data Set

4,000 unlabeled samples from the target domain are randomly selected as the unlabeled training data set $\mathcal{D}_u^T$ for model learning in KMM and our DTMKL methods. Moreover, for DTMKL_ f , only the labeled and unlabeled samples $\mathcal{D}_l^T \cup \mathcal{D}_u^T$ from the target domain are used as the training data. For KMM, the parameter B is empirically set as 0.99. And for our methods, the parameter θ in DTMKL_AT and DTMKL_ f and the parameters λ, ζ in DTMKL_ f need to be determined beforehand. We empirically set $\zeta = 0.1$ and $\theta = 10^{-5}$. Recall that the parameter λ in DTMKL_ f is used to balance the costs from labeled data and unlabeled data. Considering that the total number of unlabeled target samples is roughly 10 times more than that of the labeled target samples, we fix $\lambda = 0.1$ in our experiments.

Base kernels are predetermined for all methods. Specifically, we use four types of kernels: Gaussian kernel (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$), 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.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2 + 1}$) and inverse distance kernel (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\sqrt{\gamma}\|\mathbf{x}_i - \mathbf{x}_j\| + 1}$), where the kernel parameter γ is set as the default value $\frac{1}{d} = 0.0029$ ($d = 346$ is the feature dimension) in LIBSVM. And for each type of kernels, we use 13 kernel parameters $1.2^{\delta+3}\gamma$, $\delta \in \{-3, -2.5, \dots, 2.5, 3\}$. In total, we have 52 base kernels for all methods.

Note that our framework can readily incorporate other methods such as FR. Therefore, we introduce another approach (referred to as DTMKL_AT_{FR}) by replacing SVM

with FR in DTMKL-AT, in which we employ the kernel proposed in the FR method [29] to form the base kernels for DTMKL-AT_{FR}.

For performance evaluation, we use non-interpolated Average Precision (AP) [14, 158, 159] which has been used as the official performance metric in TRECVID since 2001. AP is related to the multi-point average precision value of a precision-recall curve, and incorporates the effect of recall when AP is computed over the entire classification results.

3.4.2.2 Details on the 20 Newsgroups and Email Spam Data Sets

On two text data sets, all the test data in the target domain are also considered as the unlabeled data in the training stage. And for our proposed method DTMKL- f , the unlabeled data from the target domain as well as the labeled data from both the source and target domains are used to construct the training data set, *i.e.*, $\mathcal{D}^S \cup \mathcal{D}_l^T \cup \mathcal{D}_u^T$. For DTMKL- f , we set $\lambda = 1$ in the experiments, because the total number of unlabeled target samples is roughly the same with that of the labeled training samples from both domains.

We consider two types of base kernels: linear kernel (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i' \mathbf{x}_j$) and polynomial kernel (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i' \mathbf{x}_j + 1)^a$), where $a = 1.5, 1.6, \dots, 2$. Then, we have totally 7 base kernels for all methods. Classification accuracy is adopted as the performance evaluation metric for text categorization.

3.4.3 Results of Video Concept Detection

We compare our DTMKL methods with other algorithms on the challenging TRECVID data set for the video concept detection task. For each concept, we count the frequency (referred to as positive frequency) of positive samples in the source domain. According to the positive frequency, we partition all the 36 concepts into three groups (*i.e.*, Group_High, Group_Med and Group_Low), with 12 concepts for each group in Figure 3.2. The concepts in Group_High, Group_Med and Group_Low are with high, moderate and low positive frequencies, respectively. And the average results of all the methods are presented in Table 3.2, where Mean Average Precisions (MAPs) of the concepts in three groups and all 36 concepts are referred to as MAP_High, MAP_Med, MAP_Low and MAP_ALL, respectively. From Table 3.2, we have the following observations:

Table 3.2: Mean average precisions (MAPs) (%) of all the methods on the TRECVID data set. MAPs are from concepts of three individual groups and all 36 concepts.

	SVM_T	SVM_A	SVM_AT	MKL_AT	FR	A-SVM	CD-SVM	KMM	DTMKL_AT	DTMKL_AT _{FR}	DTMKL_ <i>f</i>
MAP_High	39.6	39.4	44.1	42.1	45.7	45.4	43.8	44.0	45.0	46.7	46.5
MAP_Med	12.0	9.2	12.7	11.7	13.1	13.1	12.1	12.7	12.9	13.7	15.1
MAP_Low	15.3	2.5	14.5	14.4	15.4	15.3	14.9	14.5	14.6	15.8	16.4
MAP_ALL	22.3	17.0	23.8	22.7	24.7	24.6	23.6	23.7	24.2	25.4	26.0

Figure 3.2: Per-concept APs of all the 36 concepts using different methods. The concepts are divided into three groups according to the positive frequency. Our methods achieve the best performances for the circled concepts.

- 1) SVM_A is much worse than SVM_T according to the MAPs over all the 36 concepts, which demonstrates that the SVM classifier learned with the training data from the source domain performs poorly on the target domain. The explanation is that the data distributions of TRECVID data sets collected in different years are quite different. It is interesting to observe that SVM_AT outperforms SVM_T and SVM_A in terms of MAP_High, but SVM_T is better than SVM_AT and SVM_A in terms of MAP_Low. The explanation is that the concepts in Group_High generally have a large number of positive samples in both source and target domains. Intuitively, when sufficient positive samples exist in both domains, the samples distribute densely in the feature space. In this case, the distributions of samples from two domains may overlap between each other [34], and thus, the data from the source domain may be helpful for video concept detection in the target domain. On the other hand, for the concepts in Group_Low, the positive samples from both domains distribute sparsely in the feature space. It is more likely that there is less overlap between the data distributions of two domains. Therefore, for the concepts in Group_Low, the data from the source domain may degrade the performance for video concept detection in the target domain.
- 2) MKL_AT is worse than SVM_AT. The assumption in MKL is the training data and the test data come from the same domain. When the data distributions of different domains change considerably in transfer learning, the optimal kernel combination coefficients may not be effectively learned by using MKL methods based on the combined data set from two domains.
- 3) FR and A-SVM outperform SVM_AT in terms of MAPs from all the three groups, which demonstrates that the information from the source domain can be effectively used in FR and A-SVM to improve the classification performance in the target domain. We also observe that KMM and CD-SVM are slightly worse than SVM_AT in terms of MAP_ALL. A possible explanation is that in CD-SVM, k-nearest neighbors from the target domain are used to define the weights for the source samples. When the total number of positive training samples in the target domain is very limited (*e.g.*, 10 positive samples per concept in this work), the learned weights

for the source samples are not reliable, which may degrade the performance of CD-SVM. Similarly, KMM learns the weights for the source samples in an unsupervised setting without using any label information, which may not be as effective as other transfer learning methods (*e.g.*, FR and A-SVM).

- 4) DTMKL_AT is better than SVM_AT and MKL_AT in terms of MAPs over all the 36 concepts. Moreover, DTMKL_AT_{FR} and DTMKL_*f* outperform all other method in terms of MAPs from all the three groups. These results clearly demonstrate that the DTMKL methods can successfully minimize the data distribution mismatch between two domains and the structural risk functional through effective combination of multiple base kernels. DTMKL_*f* is better than DTMKL_AT_{FR} in terms of MAP_ALL, because of the additional utilization of the base classifiers. DTMKL_AT_{FR} or DTMKL_*f* achieve the best results in 21 out of 36 concepts. In addition, some concepts enjoy large performance gains. For instance, the AP for the concept “Waterscape_Waterfront” significantly increases from 20.0% (A-SVM) to 24.5% (DTMKL_*f*), equivalent to a 22.5% relative improvement; and the AP for the concept “Car” is improved from 11.9% (CD-SVM) to 14.3% (DTMKL_*f*), equivalent to a 20.2% relative improvement. Compared with the best results from the existing methods, DTMKL_*f* (15.1%) enjoys a relative improvement 15.3% over FR and A-SVM (13.1%) in terms of MAP_Med, DTMKL_*f* (16.4%) enjoys a relative improvement 6.5% over FR (15.4%) in terms of MAP_Low. Moreover, compared with FR (24.7%), A-SVM (24.6%), KMM (23.7%), CD-SVM (23.6%), MKL_AT (22.7%), SVM_AT (23.8%) and SVM_T (22.3%), the relative MAP improvements of DTMKL_*f* (26.0%) over all the 36 concepts are 5.3%, 5.7%, 9.7%, 10.2%, 14.5%, 9.2% and 16.6%, respectively.
- 5) We also observe that DTMKL_AT_{FR} is slightly better than DTMKL_*f* in terms of MAP_High, possibly because the distributions of samples from two domains overlap between each other in this case. We therefore propose a simple predicting method by using DTMKL_AT_{FR} for the concepts in Group_High and DTMKL_*f* for the rest concepts in Group_Med and Group_Low. The MAP of the predicting method over all 36 concepts is 26.1%, with the relative improvements over FR, A-SVM,

KMM, CD-SVM, MKL-AT, SVM-AT and SVM-T as 5.7%, 6.1%, 10.1%, 10.6%, 15.0%, 9.7% and 17.0%, respectively.

We additionally report the average training and testing time of all the methods for each concept in Table 3.3. All the experiments are performed on an IBM workstation (2.66 GHz CPU with 32 Gbyte RAM) with LIBSVM [153]. From Table 3.3, we observe that SVM-T is quite fast, because it only utilizes the labeled training data from the target domain. We also observe that some MKL-based methods (*i.e.*, MKL-AT, DTMKL-AT and DTMKL-AT_{FR}) are much faster than the late-fusion based methods except SVM-T in the training phase. Note that before learning the target classifiers for A-SVM and DTMKL-*f*, we have to obtain the pre-learned classifiers. For A-SVM, the learning of the pre-learned classifiers takes 1576 seconds per concept on average, while it takes 1618 seconds for DTMKL-*f* to learn. For both A-SVM and DTMKL-*f*, it is very fast to learn the target classifiers. Moreover, all the MKL-based methods are also much faster than the late-fusion based methods except SVM-T in the testing phase. On average, our DTMKL methods take less than one minute to finish the whole prediction phase for about 21,213 test samples from each concept, which is still acceptable in the real-world applications.

3.4.4 Results of Text Categorization

For the text categorization task, we focus the comparisons between DTMKL-*f* and other related methods using two text data sets. For each setting, we report the results of all the methods obtained by using the training data from the source domain as well as m positive and m negative training samples randomly selected from the target domain, where we set $m = 0, 1, 3, 5, 7$ and 10 for the 20 Newsgroups data set and $m = 5$ for the email spam data set. We randomly sample the training data from the target domain for five times. In Tables 3.4 and 3.5, we report the means and standard deviations of classification accuracies for all the methods on the 20 Newsgroups and email spam data sets, respectively. It is worth noting that when there are no training samples from the target domain, DTMKL-*f* can employ the base SVM classifiers learned from the source data only. But other methods like SVM-T, SVM-AT, MKL-AT, FR, A-SVM, CD-SVM

Table 3.3: Average training (TR) and testing (TE) time (in second) comparisons of all the methods on the TRECVID data set. For A-SVM and DTMKL- f , the two numbers represent the average training time for the learning of the pre-learned classifiers and the learning of the target classifier.

	SVM_T	SVM_A	SVM_AT	MKL_AT	FR	A-SVM	CD-SVM	KMM	DTMKL_AT	DTMKL_AT _{FR}	DTMKL- f
TR	1	1576	1618	686	1636	32	2287	4218	639	688	186
TE	31	475	509	34	523	511	485	508	41	43	52

Table 3.4: Means and standard deviations (%) of classification accuracies (ACC) of all the methods with different number of positive and negative training samples (*i.e.*, m) from the target domain on the 20 Newsgroups data set. Each result in the table is the best among all the results obtained by using different tradeoff parameter $C = 0.1, 0.2, 0.5, 1, 2, 5, 10, 20$ and 50. The results shown in boldface are significantly better than the others, judged by the t-test with a significance level of 0.1.

(a) comp vs. rec									
m	SVM_T	SVM_A	SVM_AT	MKL_AT	FR	A-SVM	CD-SVM	KMM	DTMKL_f
0	–	89.0±0.0	–	–	–	–	–	89.2±0.0	91.8±0.0
1	52.9±5.9	89.0±0.0	89.3±0.2	89.4±0.4	86.6±2.2	88.2±1.6	88.8±0.3	89.6±0.3	92.3±0.3
3	64.0±5.8	89.0±0.0	90.0±0.2	90.2±0.4	85.7±3.9	88.2±1.4	89.5±0.6	90.3±0.5	92.8±0.4
5	76.8±10.9	89.0±0.0	90.6±0.4	90.9±0.1	88.9±3.1	89.5±2.0	90.6±0.7	91.4±0.7	93.3±0.5
7	80.6±9.1	89.0±0.0	91.0±0.1	91.1±0.0	89.5±2.5	90.2±1.6	90.9±0.1	91.1±0.1	93.6±0.5
10	84.8±6.2	89.0±0.0	91.7±0.1	91.7±0.1	91.3±2.2	91.1±1.5	91.5±0.1	91.8±0.1	94.2±0.4

(b) comp vs. sci									
m	SVM_T	SVM_A	SVM_AT	MKL_AT	FR	A-SVM	CD-SVM	KMM	DTMKL_f
0	–	70.7±0.0	–	–	–	–	–	70.2±0.0	72.9±0.0
1	51.6±3.7	70.7±0.0	70.8±0.1	71.1±0.1	70.5±1.6	70.3±0.5	69.5±1.0	70.3±0.1	73.1±0.1
3	57.8±8.9	70.7±0.0	72.0±0.7	71.8±0.8	69.6±1.8	70.4±0.4	72.0±0.8	72.0±0.5	74.8±0.6
5	63.8±13.0	70.7±0.0	74.1±3.1	74.1±3.2	71.3±7.7	71.3±2.8	74.1±3.1	74.0±3.0	77.0±2.9
7	73.8±4.3	70.7±0.0	75.6±2.7	75.8±2.7	76.0±4.5	73.4±3.3	75.8±2.7	75.8±2.6	78.3±2.7
10	76.6±3.9	70.7±0.0	78.1±2.7	77.9±2.7	78.4±3.4	74.8±2.4	78.1±2.7	78.1±2.8	80.5±2.8

(c) comp vs. talk									
m	SVM_T	SVM_A	SVM_AT	MKL_AT	FR	A-SVM	CD-SVM	KMM	DTMKL_f
0	–	92.9±0.0	–	–	–	–	–	92.2±0.0	94.3±0.0
1	56.7±7.1	92.9±0.0	93.1±0.2	93.2±0.3	91.4±3.9	92.3±1.5	93.1±0.3	92.3±0.1	94.6±0.2
3	72.4±8.4	92.9±0.0	93.4±0.4	93.6±0.4	91.7±3.0	93.7±0.6	93.4±0.3	92.5±0.5	94.9±0.2
5	81.9±2.1	92.9±0.0	93.6±0.3	93.7±0.4	93.1±0.6	94.0±0.6	93.7±0.3	92.8±0.5	95.0±0.3
7	83.5±2.0	92.9±0.0	93.7±0.3	93.8±0.3	93.3±1.0	93.7±0.7	93.8±0.4	93.7±0.3	95.1±0.3
10	93.5±2.0	92.9±0.0	94.0±0.4	94.1±0.4	93.6±1.0	94.0±0.4	94.0±0.3	93.9±0.8	95.2±0.4

Table 3.5: Means and standard deviations (%) of classification accuracies (ACC) of all the methods with five positive and five negative training samples from the target domain on the email spam data set. Each result in the table is the best among all the results obtained by using different tradeoff parameter $C = 0.1, 0.2, 0.5, 1, 2, 5, 10, 20$ and 50. The results shown in boldface are significantly better than the others, judged by the t-test with a significance level of 0.1.

(a) User1 (source domain) & User2 (target domain)									
	SVM_T	SVM_A	SVM_AT	MKL_AT	FR	A-SVM	CD-SVM	KMM	A-MKL
ACC	80.2±3.5	96.1±0.0	96.2±0.1	96.2±0.0	92.5±2.8	95.4±0.9	96.2±0.2	96.2±0.1	96.3±0.3
									DTMKL-f
									96.9±0.1
(b) User2 (source domain) & User3 (target domain)									
	SVM_T	SVM_A	SVM_AT	MKL_AT	FR	A-SVM	CD-SVM	KMM	A-MKL
ACC	82.0±2.6	96.9±0.0	97.0±0.1	97.0±0.1	92.1±2.4	96.0±1.3	97.0±0.1	97.0±0.1	97.3±0.0
									DTMKL-f
									97.7±0.1
(c) User3 (source domain) & User1 (target domain)									
	SVM_T	SVM_A	SVM_AT	MKL_AT	FR	A-SVM	CD-SVM	KMM	A-MKL
ACC	79.1±1.9	91.7±0.0	91.8±0.1	91.4±0.1	91.8±2.5	92.4±1.2	91.8±0.1	91.8±0.1	92.8±0.5
									DTMKL-f
									94.0±0.4

and A-MKL [17] cannot work in this case. Also note that for all the methods, each result in Tables 3.4 and 3.5 is the best among all the results obtained by using different tradeoff parameter $C = 0.1, 0.2, 0.5, 1, 2, 5, 10, 20$ and 50. From Tables 3.4 and 3.5, we have the following observations:

- 1) On both data sets, MKL_AT is comparable with SVM_AT, which shows that the source domain is relevant to the target domain. The performances of SVM_T and SVM_AT become better on the 20 Newsgroups data set, when the number of labeled positive and negative training samples (*i.e.*, m) increases. And SVM_AT outperforms SVM_T and SVM_A on both data sets, which demonstrates that it is beneficial to utilize the data from the source domain to improve the performance in the target domain.
- 2) Some transfer learning methods (*i.e.*, CD-SVM and KMM) generally achieve similar performances when compared with SVM_AT. The explanation is that the data distributions of two domains are quite related, making it difficult for the existing transfer learning methods to further improve the performances in the target domain. We also observe that A-SVM is worse than SVM_AT in most settings on the two text data sets. It seems that the limited number of labeled training samples from the target domain are not sufficient to facilitate robust adaptation for A-SVM. And it is interesting to observe that FR is generally worse than SVM_AT on the email spam data set in terms of the means of classification accuracies. A possible explanation is that the kernel of FR, which is constructed based on the augmented features, is less effective on this data set. Moreover, in most cases, A-MKL [17] outperforms other methods except DTMKL_ f in terms of the means of classification accuracies.
- 3) Our proposed method DTMKL_ f is consistently better than all the other methods in terms of the means of classification accuracies on both data sets, thanks to the explicit modeling of the data distribution mismatch as well as the successful utilization of the unlabeled data and the base classifiers. As shown in Table 3.4, when the number of labeled positive and negative training samples (*i.e.*, m) from the target domain increases, DTMKL_ f becomes better on the 20 Newsgroups data

set. Moreover, judged by the t-test with a significance level of 0.1, DTMKL- f is significantly better than other methods in all settings.

We also compare our proposed method DTMKL- f with the competitive methods including MKL-AT, A-SVM, CD-SVM and KMM by using different tradeoff parameter $C = 0.1, 0.2, 0.5, 1, 2, 5, 10, 20$ and 50. The results of all the methods are obtained by using m positive and m negative training samples from the target domain as well as the training data from the source domain, in which we set $m = 5$ and 10 for the 20 Newsgroups data set in Figure 3.3. From the figure, we observe that when C becomes larger, all the methods tend to have better performances. In addition, our method DTMKL- f consistently outperforms other methods in terms of the means of classification accuracies. Moreover, DTMKL- f is also relatively stable according to the standard deviations of classification accuracies. We have similar observations on this data set when using different m and also on the email spam data set.

Recall that the parameter λ in DTMKL- f balances the costs from the labeled and unlabeled samples (see (3.15)). In Figure 3.4, we take the 20 Newsgroups data set as an example to investigate the performance variation of DTMKL- f with respect to the parameter λ , in which we set $m = 5$ and the tradeoff parameter $C = 2$ and 5. Note the total number of labeled samples from two domains and the number of unlabeled samples from the target domain are almost the same on the 20 Newsgroups data set. From Figure 3.4, we have the following observations:

- 1) The performance of DTMKL- f changes with different λ in a large range (*i.e.*, $\lambda \in [0.1, 10]$).
- 2) When λ is quite small or quite large (*i.e.*, the cost from labeled data or unlabeled data is more important), the performances of DTMKL- f generally degrade a bit.
- 3) When we set $\lambda \in [0.5, 1.5]$, DTMKL- f achieves the best results and is not sensitive to the parameter λ as well. In this case, both the labeled data and the unlabeled data from the target domain can be effectively utilized to learn a robust classifier. We have similar observations on this data set when using different C and m , and on the email spam data set as well.

Figure 3.3: Performance comparisons of DTMKL_f with other methods in terms of the means and standard deviations of classification accuracies on the 20 Newsgroups data set by using different tradeoff parameter $C = 0.1, 0.2, 0.5, 1, 2, 5, 10, 20$ and 50 . We set $m = 5$ (top) and $m = 10$ (bottom).

Figure 3.4: Performance (*i.e.*, the means of classification accuracies) variation of DTMKL_f with respect to the balance parameter $\lambda \in [0.1, 10]$ on the 20 Newsgroups data set. We set the tradeoff parameter $C = 2$ and $C = 5$.

Figure 3.5: Illustration of the convergence of DTMKL_AT.

3.4.5 Convergence

In Theorem 3.1, we theoretically prove that DTMKL_AT is jointly convex with respect to $\mathbf{d}$, $\mathbf{v}_m$, b and ξ_i . Here we take two concepts “Person” and “Airplane” from the TRECVID data set as examples to experimentally demonstrate the convergence of DTMKL_AT. As shown in Figure 3.5, the objective values of DTMKL_AT converge after less than five iterations. We have similar observations for other concepts as well.

3.5 Summary

In this chapter, we have proposed a unified transfer learning framework Domain Transfer Multiple Kernel Learning (DTMKL) to explore the single source domain and single target domain problem. DTMKL simultaneously learns a kernel function and a target classifier by minimizing the structural risk functional as well as the distribution mismatch between the samples from the source and target domains. By assuming that the kernel function is a linear combination of multiple base kernels, we also develop a unified learning algorithm by using the second order derivatives to accelerate the convergence of the proposed framework. Most importantly, many existing kernel methods including SVM, Support Vector Regression (SVR), Kernel Regularized Least-Squares (KRLS) and so on, can be readily incorporated into the framework of DTMKL to tackle transfer learning problems.

Based on the DTMKL framework, we propose two methods DTMKL_AT and DTMKL- f by using SVM and existing classifiers, respectively. For DTMKL- f , many machine

learning methods (*e.g.*, SVM and SVR) can be used to learn the base classifiers. Specifically, in DTMKL- f , we enforce that 1) for the unlabeled target data, the target classifier produces similar decision values with those obtained from the base classifiers; 2) for the labeled target data, the decision values obtained from the target classifier are close to the true labels. Experimental results show that DTMKL- f outperforms existing transfer learning and multiple kernel learning methods on the challenging TRECVID data set for video concept detection as well as on the 20 Newsgroups and email spam data sets for text categorization.

In the experiments, we randomly select a number of unlabeled target samples as the training data for DTMKL- f . Considering that it is beneficial to establish the optimal balance between the labeled and unlabeled samples [160], we will investigate how to determine such optimal balance in the future. Moreover, we will also study how to automatically determine the optimal parameters for DTMKL-AT and DTMKL- f .

Chapter 4

Adaptive Multiple Kernel Learning

In this chapter, we propose a visual event recognition framework for consumer videos by leveraging a large amount of loosely labeled web videos (*e.g.*, from YouTube). Observing that consumer videos contain large intra-class variations within the same type of events, we first propose a new pyramid matching method called Aligned Space-Time Pyramid Matching (ASTPM) to measure the distances between two video clips. Second, we propose a new transfer learning method called Adaptive Multiple Kernel Learning (A-MKL) in order to 1) fuse the information from multiple pyramid levels and features (*i.e.*, space-time features and static SIFT features) and 2) cope with the considerable variation in feature distributions between videos from two domains (*i.e.*, web video domain and consumer video domain). We test A-MKL on consumer video recognition by using a large number of labeled web videos.

4.1 Introduction

In recent years, digital cameras and mobile phone cameras are becoming popular in our daily life. Consequently, there is an increasingly urgent demand on indexing and retrieving from a large amount of unconstrained consumer videos. In particular, visual event recognition in consumer videos has attracted growing attention. However, this is an extremely challenging computer vision task due to two main issues. First, consumer videos are generally captured by amateurs using hand-held cameras of unstaged events and thus contain considerable camera motion, occlusion, cluttered background and large intra-class variations within the same type of events, making their visual cues highly

variable and thus less discriminant. Second, these users are generally reluctant to annotate many consumer videos, posing a great challenge to the traditional video event recognition techniques that often cannot learn robust classifiers from a limited number of labeled training videos.

While a large number of video event recognition techniques have been proposed (see Section 4.2 for more details), few [161–165] focused on event recognition in the highly unconstrained consumer video domain. Loui *et al.* [165] developed a consumer video data set which was manually labeled for 25 concepts including activities, occasions, static concepts like scenes and objects, as well as sounds. Based on this data set, Chang *et al.* [161] developed a multi-modal consumer video classification system by using visual features and audio features. In the web video domain, Liu *et al.* [164] employed strategies inspired by PageRank to effectively integrate both motion features and static features for action recognition in YouTube videos. In [162], action models were first learned from loosely labeled web images and then used for identifying human actions in YouTube videos. However, the work in [162] cannot distinguish actions like “sitting_down” and “standing_up” because it did not utilize temporal information in its image-based model. Recently, Ikizler-Cinbis and Sclaroff [163] proposed to employ multiple instance learning to integrate multiple features of the people, objects and scenes for action recognition in YouTube videos.

Most event recognition methods [148, 159, 161, 164, 166–168] followed the conventional framework. First, a sufficiently large corpus of training data is collected, in which the concept labels are generally obtained through expensive human annotation. Next, robust classifiers (also called models or concept detectors) are learned from the training data. Finally, the classifiers are used to detect the presence of the concepts in any test data. When sufficient and strong labeled training samples are provided, these event recognition methods have achieved promising results. However, for visual event recognition in consumer videos, it is time-consuming and expensive for users to annotate a large number of consumer videos. It is also well-known that the learned classifiers from a limited number of labeled training samples are usually not robust and do not generalize well.

In this chapter, we propose a new event recognition framework for consumer videos by leveraging a large amount of loosely labeled YouTube videos. Our work is based on

Figure 4.1: Four sample frames from consumer videos and YouTube videos. Our work aims to recognize the events in consumer videos by using a limited number of labeled consumer videos and a large number of YouTube videos. These two events and their examples (*i.e.*, “picnic” and “sports”) illustrate the considerable appearance differences between consumer videos and YouTube videos, which poses great challenges to conventional learning schemes but can be effectively handled by our transfer learning method, called Adaptive Multiple Kernel Learning (A-MKL).

Figure 4.2: The flowchart of the proposed visual event recognition framework. It consists of an *pyramid matching* method ASTPM that effectively measures the distances between two video clips and a *transfer learning* method A-MKL that effectively copes with the considerable variation in feature distributions between web videos and consumer videos.

the observation that a large amount of loosely labeled YouTube videos can be readily obtained by using keywords (also called tags) based search. However, the quality of YouTube videos is generally lower than consumer videos because YouTube videos are often down-sampled and heavily compressed by the web server. In addition, YouTube videos may have been selected and edited to attract attention while consumer videos are in their natural captured state. In Figure 4.1, we show four frames from two events (*i.e.*, “picnic” and “sports”) as examples to illustrate the considerable appearance differences between consumer videos and YouTube videos. Clearly, the visual feature distributions of samples from the two domains (*i.e.*, web video domain and consumer video domain) can change considerably in terms of the statistical properties (such as mean, intra-class and inter-class variance).

Our proposed framework is shown in Figure 4.2 and consists of two contributions. First, we extend the recent work on pyramid matching [159, 166, 169–171] and present a new pyramid matching method called *Aligned Space-Time Pyramid Matching* (ASTPM) to effectively measure the distances between two video clips that may be from different domains. Specifically, we divide each video clip into space-time volumes over multiple levels. We calculate the pair-wise distances between any two volumes and further integrate the information from different volumes with Integer-flow Earth Mover’s Distance (EMD) to explicitly align the volumes. In contrast to the fixed volume-to-volume matching used in [166], the space-time volumes of two videos across different space-time locations may be matched using our ASTPM method, making it better at coping with the large intra-class variations within the same type of events (*e.g.*, moving objects in consumer videos can appear at different space-time locations, and the background within two different videos even captured from the same scene may be shifted due to considerable camera motions).

The second is our main contribution. In order to cope with the considerable variation in feature distributions between videos from the web video domain and consumer video domain, we propose a new transfer learning method, referred to as Adaptive Multiple Kernel Learning (A-MKL). Specifically, we first obtain one pre-learned classifier for each event class at each pyramid level and with each type of local features, in which the existing kernel methods can be readily employed. In this work, we adopt the *pre-learned*

average classifier by equally fusing a set of SVM classifiers that are pre-learned based on a combined training set from two domains by using multiple base kernels from different kernel types and parameters. For each event class, we then learn an *adapted classifier* based on multiple base kernels and the pre-learned average classifiers from this event class or all the event classes by minimizing both the structural risk functional and mismatch between data distributions of two domains. It is noteworthy that the utilization of the pre-learned average classifiers from all the event classes in A-MKL is based on the observation that some events may share common motion patterns. For example, the videos from some events (such as “birthday”, “picnic” and “wedding”) usually contain a number of people talking with each other. Therefore, it is beneficial to learn an adapted classifier for “birthday” by also leveraging the pre-learned classifiers from “picnic” and “wedding”.

The remainder of this chapter is organized as follows. Section 4.2 provides brief reviews of event recognition. The proposed Aligned Space-Time Pyramid Matching method and A-MKL will be introduced in Sections 4.3 and 4.4, respectively. Extensive experimental results will be presented in Section 4.5, followed by conclusion in Section 4.6.

4.2 Related Work on Event Recognition

Event recognition methods can be roughly categorized into model-based methods and appearance-based techniques. Model-based approaches relied on various models including HMM [172], coupled HMM [173], and Dynamic Bayesian Network [174] to model the temporal evolution. The relationships among different body parts and regions are also modeled in [172, 173], in which object tracking needs to be conducted at first before model learning.

Appearance-based approaches employed space-time features extracted from volumetric regions that can be densely sampled or the salient regions with significant local variations in both spatial and temporal dimensions [167, 168, 175]. In [176], Ke *et al.* employed boosting to learn a cascade of filters based on space-time features for efficient visual event detection. Laptev and Lindeberg [175] extended the ideas of Harris interest point operators and Dollar *et al.* [177] employed separable linear filters to detect the salient volumetric regions. Statistical learning methods including Support Vector Machine

(SVM) [168] and probabilistic Latent Semantic Analysis (pLSA) [167] were then applied to the above space-time features to obtain the final classification. Recently, Kovashka and Grauman [178] proposed a new feature formation technique by exploiting multi-level vocabularies of space-time neighborhoods. Promising results [167, 168, 178–180] have been reported on video data sets under controlled settings, such as Weizman [179] and KTH [168] data sets. Interested readers may refer to [181] for a recent survey.

Recently, researchers proposed new methods to address the more challenging event recognition task on video data sets captured under much less uncontrolled conditions, including movies [148, 166] and broadcast news videos [159]. In [166], Laptev *et al.* integrated local space-time features (*i.e.*, HoG and HoF), space-time pyramid matching and SVM for action classification in movies. In order to locate the actions from movies, a new discriminative clustering algorithm [182] was developed based on the weakly-labeled training data that can be readily obtained from movie scripts without any cost of manual annotation. Sun *et al.* [148] employed Multiple Kernel Learning (MKL) to efficiently fuse three types of features including SIFT average descriptor and two trajectory-based features. To recognize events in diverse broadcast news videos, Xu and Chang [159] proposed a multi-level temporal matching algorithm for measuring video similarity.

However, all these methods followed the conventional learning framework by assuming that the training and test samples are from the same domain and distribution. When the total number of labeled training samples is limited, the performances of these methods would suffer. In contrast, the goal of this work is to propose an effective event recognition framework for consumer videos by leveraging a large amount of loosely labeled web videos, where we must deal with the distribution mismatch between videos from two domains (*i.e.*, web video domain and consumer video domain). As a result, our algorithm can learn a robust classifier for event recognition requiring only a small number of labeled consumer videos.

4.3 Aligned Space-Time Pyramid Matching

Recently, pyramid matching algorithms were proposed for different applications, such as object recognition, scene classification, and event recognition in movies and news

videos [159, 166, 169–171]. These methods involved pyramidal binning in different domains (*e.g.*, feature, spatial, or temporal domain), and improved performances were reported by fusing the information from multiple pyramid levels. Spatial pyramid matching [170] and its space-time extension [166] used fixed block-to-block matching and fixed volume-to-volume matching (we refer to them as *unaligned space-time matching*), respectively. In contrast, our proposed *Aligned Space-Time Pyramid Matching* (ASTPM) extends the methods of Spatially Aligned Pyramid Matching (SAPM) [171] and Temporally Aligned Pyramid Matching (TAPM) [159] from either spatial domain or temporal domain to joint space-time domain, where the volumes across different space and time locations can be matched.

Similar to [166], we divide each video clip into 8^l non-overlapped space-time volumes over multiple levels, $l = 0, \dots, L - 1$, where the volume size is set as $1/2^l$ of the original video in width, height and temporal dimension. Figure 4.3 illustrates the partition for two videos V_i and V_j at level-1. Following [166], we extract the local space-time (ST) features including Histograms of Oriented Gradient (HoG) and Histograms of Optical Flow (HoF), which are further concatenated together to form lengthy feature vectors. We also sample each video clip to extract image frames and then extract static local SIFT features from them [183].

Our method consists of two matching stages. In the first matching stage, we calculate the pairwise distance D_{rc} between each two space-time volumes $V_i(r)$ and $V_j(c)$, where $r, c = 1, \dots, R$ with R being the total number of volumes in a video. The space-time features are vector-quantized into *visual words* and then each space-time volume is represented as a token-frequency feature. As suggested in [166], we use χ^2 distance to measure the distance D_{rc} . Note that each space-time volume consists of a set of image blocks. We also extract token-frequency (tf) features from each image block by vector-quantizing the corresponding SIFT features into visual words. And based on the SIFT features, as suggested in [159], the pairwise distance D_{rc} between two volumes $V_i(r)$ and $V_j(c)$ is calculated by using Earth Mover’s Distance (EMD) [184] as follows:

$$D_{rc} = \frac{\sum_{u=1}^H \sum_{v=1}^I \hat{f}_{uv} d_{uv}}{\sum_{u=1}^H \sum_{v=1}^I \hat{f}_{uv}},$$

Figure 4.3: Illustration of the proposed Aligned Space-Time Pyramid Matching (ASTPM) method at level-1: (a) Each video is divided into 8 space-time volumes along the width, height and temporal dimensions; (b) The matching results are obtained by using our ASTPM method. Each pair of matched volumes from two videos is highlighted in the same color. For better visualization, please see the colored PDF file.

where H, I are the numbers of image blocks in $V_i(r), V_j(c)$ respectively, d_{uv} is the distance between two image block (Euclidean distance is used in this work), and $\hat{f}_{uv}$ is the optimal flow that can be obtained by solving the linear programming problem as follows:

$$\begin{aligned} \hat{f}_{uv} &= \arg \min_{f_{uv} \geq 0} \sum_{u=1}^H \sum_{v=1}^I f_{uv} d_{uv}, \\ \text{s.t. } \sum_{u=1}^H \sum_{v=1}^I f_{uv} &= 1; \sum_{v=1}^I f_{uv} \leq \frac{1}{H}, \forall u; \sum_{u=1}^H f_{uv} \leq \frac{1}{I}, \forall v \end{aligned}$$

In the second stage, we further integrate the information from different volumes with Integer-flow EMD to explicitly align the volumes. We try to solve a flow matrix $\hat{F}_{rc}$ containing binary elements that represent unique matches between volumes $V_i(r)$ and $V_j(c)$. As suggested in [159, 171], such binary solution can be conveniently computed by using the standard Simplex method for linear programming. The following Theorem 4.2 is utilized:

Theorem 4.2 ([185]) *The linear programming problem*

$$\begin{aligned} \hat{F}_{rc} &= \arg \min_{F_{rc}} \sum_{r=1}^R \sum_{c=1}^R F_{rc} D_{rc}, \\ \text{s.t. } \sum_{c=1}^R F_{rc} &= 1, \forall r; \sum_{r=1}^R F_{rc} = 1, \forall c, \end{aligned}$$

will always have an integer optimum solution when solved with the Simplex method.

Figure 4.3 illustrates the matching results of two videos after using our ASTPM method, indicating the reasonable matching between similar scenes (*i.e.*, the crowds, the playground and the Jumbotron TV screens in the two videos). It is also worth mentioning that our ASTPM method can preserve the space-time proximity relations between volumes from two videos at level-1 when using the ST or SIFT features. Specifically, the ST features (*resp.*, SIFT features) in one volume can only be matched to the ST features (*resp.*, SIFT features) within another volume at level-1 in our ASTPM method rather than arbitrary ST features (*resp.*, SIFT features) within the entire video as in the classical bag-of-words model (*e.g.*, ASTPM at level-0).

Finally, the distance $D_l(V_i, V_j)$ between two video clips V_i and V_j at level- l can be directly calculated by

$$D_l(V_i, V_j) = \frac{\sum_{r=1}^R \sum_{c=1}^R \hat{F}_{rc} D_{rc}}{\sum_{r=1}^R \sum_{c=1}^R \hat{F}_{rc}}.$$

In the next section, we will propose a new transfer learning method to fuse the information from multiple pyramid levels and different types of features.

4.4 Adaptive Multiple Kernel Learning

Following the prior terminology, we refer to the web video domain as *source domain* $\mathcal{D}^A$ (*a.k.a., auxiliary domain*) and consumer video domain as *target domain* $\mathcal{D}^T = \mathcal{D}_l^T \cup \mathcal{D}_u^T$, where $\mathcal{D}_l^T$ and $\mathcal{D}_u^T$ represent the labeled and unlabeled data in the target domain, respectively. In this work, we denote $\mathbf{I}_n$ as the $n \times n$ identity matrix and $\mathbf{0}_n, \mathbf{1}_n \in \mathbb{R}^n$ as $n \times 1$ column vectors of all zeros and all ones, respectively. The inequality $\mathbf{a} = [a_1, \dots, a_n]' \geq \mathbf{0}_n$ means that $a_i \geq 0$ for $i = 1, \dots, n$. Moreover, the element-wise product between vectors $\mathbf{a}$ and $\mathbf{b}$ is defined as $\mathbf{a} \circ \mathbf{b} = [a_1 b_1, \dots, a_n b_n]'$.

4.4.1 Brief Review of Related Work

Transfer learning methods have been proposed for many applications [15, 29, 30, 36]. In Chapter 3, we have introduced two pieces of previous work for transfer learning. One is the Maximum Mean Discrepancy [37] (MMD) criterion which measures the mismatch between two data distributions based on the distance between the means of samples respectively from the source domain $\mathcal{D}^S$ and the target domain $\mathcal{D}^T$ in a RKHS spanned by a kernel function. The MMD criterion is also a key component in our DTMKL framework (see Section 3.2.1 and Section 3.3 for more details). The other one is Adaptive SVM (A-SVM) [30] in which the target classifier $f^T(\mathbf{x})$ is adapted from an existing classifier $f^S(\mathbf{x})$ (referred to as source classifier) trained based on the samples from the source domain. Specifically, the target decision function is defined as follows:

$$f^T(\mathbf{x}) = f^S(\mathbf{x}) + \Delta f(\mathbf{x}), \quad (4.1)$$

where $\Delta f(\mathbf{x})$ is called as a *perturbation function* that is learned by using the labeled data from the target domain only (*i.e.*, $\mathcal{D}_l^T$). While A-SVM can also employ multiple

source classifiers, these source classifiers are fused with equal weights to obtain $f^S(\mathbf{x})$ [30]. Moreover, the target classifier $f^T(\mathbf{x})$ is learned based on only one kernel.

4.4.2 Formulation of A-MKL

Motivated by [30, 119], we propose a new transfer learning method to learn a target classifier adapted from a set of pre-learned classifiers as well as a perturbation function that is based on multiple base kernels k_m 's. The pre-learned classifiers are used as prior for learning a robust adapted target classifier. In A-MKL, the existing machine learning methods (*e.g.*, SVM, FR and so on) using different types of features (*e.g.*, SIFT and ST features) can be readily used to obtain the pre-learned classifiers. In contrast to A-SVM [30] which uses the equal weights to combine the pre-learned source classifiers, we also learn the linear combination coefficients $\beta_p|_{p=1}^P$ of the pre-learned classifiers $f_p(\mathbf{x})|_{p=1}^P$ in this work, where P is the total number of the pre-learned classifiers. In this work, we use the average classifiers from one event class or all the event classes as the pre-learned classifiers (see Sections 4.5.3 and 4.5.6 for more details). We additionally employ multiple *predefined* base kernels to model the perturbation function in this work, because the utilization of multiple *base kernels* k_m 's instead of a single kernel can further enhance the interpretability of the decision function and improve performances [143]. We refer to our transfer learning method based on multiple base kernels as Adaptive Multiple Kernel Learning (A-MKL), because A-MKL can handle the domain transfer from the web video domain to the consumer video domain.

Following the traditional MKL assumption [143] and the DTMKL framework presented in Chapter 3, we assume the kernel function k is represented as a linear combination of multiple base kernels k_m 's, namely, $k = \sum_{m=1}^M d_m k_m$, where d_m 's are the linear combination coefficients, $d_m \geq 0$ and $\sum_{m=1}^M d_m = 1$; each base kernel function k_m is induced from the nonlinear feature mapping function $\phi_m(\cdot)$, *i.e.*, $k_m(\mathbf{x}_i, \mathbf{x}_j) = \phi_m(\mathbf{x}_i)' \phi_m(\mathbf{x}_j)$, and M is the total number of base kernels. Inspired by semiparametric SVM [186], we define the target decision function, which is adapted from multiple pre-learned classifiers, on any sample $\mathbf{x}$ as follows:

$$f^T(\mathbf{x}) = \sum_{p=1}^P \beta_p f_p(\mathbf{x}) + \underbrace{\sum_{m=1}^M d_m \mathbf{w}_m' \phi_m(\mathbf{x})}_{\Delta f(\mathbf{x})} + b, \quad (4.2)$$

where $\Delta f(\mathbf{x}) = \sum_{m=1}^M d_m \mathbf{w}'_m \phi_m(\mathbf{x}) + b$ is the perturbation function with b as the bias term. The perturbation function can adjust the adapted target classifier in order to obtain better predictions for data samples. Note that multiple base kernels are employed in $\Delta f(\mathbf{x})$.

Similar to our DTMKL framework in Chapter 3, we employ the MMD criterion to reduce the mismatch between the data distributions of two domains in this work. Let us define the linear combination coefficient vector as $\mathbf{d} = [d_1, \dots, d_M]'$ and the feasible set of $\mathbf{d}$ as $\mathcal{M} = \{\mathbf{d} \in \mathbb{R}^M | \mathbf{1}'_M \mathbf{d} = 1, \mathbf{d} \geq \mathbf{0}_M\}$. Based on the traditional MKL assumption [143], similarly as in Section 3.3.3, the MMD criterion (3.2) can be rewritten as:

$$\text{DIST}_k^2(\mathcal{D}^A, \mathcal{D}^T) = \Omega(\mathbf{d}) = \mathbf{h}'\mathbf{d}, \quad (4.3)$$

where $\mathbf{h} = [\text{tr}(\mathbf{K}_1 \mathbf{S}), \dots, \text{tr}(\mathbf{K}_M \mathbf{S})]'$, $\mathbf{K}_m = [\phi_m(\mathbf{x})' \phi_m(\mathbf{x})] \in \mathbb{R}^{N \times N}$ is the m -th base kernel matrix defined on the samples from both source and target domains. Let us denote the labeled training samples from both the source and target domains (*i.e.*, $\mathcal{D}^A \cup \mathcal{D}_l^T$) as $(\mathbf{x}_i, y_i)_{i=1}^n$, where n is the total number of labeled training samples from the two domains. The optimization problem in A-MKL is then formulated as follows:

$$\min_{\mathbf{d} \in \mathcal{M}} G(\mathbf{d}) = \frac{1}{2} \Omega^2(\mathbf{d}) + \theta J(\mathbf{d}), \quad (4.4)$$

where

$$\begin{aligned} J(\mathbf{d}) = \min_{\mathbf{w}_m, \boldsymbol{\beta}, b, \xi_i} & \quad \frac{1}{2} \left(\sum_{m=1}^M d_m \|\mathbf{w}_m\|^2 + \lambda \|\boldsymbol{\beta}\|^2 \right) + C \sum_{i=1}^n \xi_i, \\ \text{s.t.} & \quad y_i f^T(\mathbf{x}_i) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n, \end{aligned} \quad (4.5)$$

$\boldsymbol{\beta} = [\beta_1, \dots, \beta_P]'$ is the vector of β_p 's and $\lambda, C > 0$ are the tradeoff parameters. Denote $\tilde{\mathbf{w}}_m = [\mathbf{w}'_m, \sqrt{\lambda} \boldsymbol{\beta}']'$ and $\tilde{\phi}_m(\mathbf{x}_i) = [\phi_m(\mathbf{x}_i)', \frac{1}{\sqrt{\lambda}} \mathbf{f}(\mathbf{x}_i)']'$, where $\mathbf{f}(\mathbf{x}_i) = [f_1(\mathbf{x}_i), \dots, f_P(\mathbf{x}_i)]'$. The optimization problem in (4.5) can then be rewritten as follows:

$$\begin{aligned} J(\mathbf{d}) = \min_{\tilde{\mathbf{w}}_m, b, \xi_i} & \quad \frac{1}{2} \sum_{m=1}^M d_m \|\tilde{\mathbf{w}}_m\|^2 + C \sum_{i=1}^n \xi_i, \\ \text{s.t.} & \quad y_i \left(\sum_{m=1}^M d_m \tilde{\mathbf{w}}'_m \tilde{\phi}_m(\mathbf{x}_i) + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n. \end{aligned} \quad (4.6)$$

Algorithm 2 Adaptive Multiple Kernel Learning

-
- 1: **Input:** labeled training samples $(\mathbf{x}_i, y_i)_{i=1}^n$, pre-learned classifiers $f_p(\mathbf{x})|_{p=1}^P$ and pre-defined base kernel functions $k_m|_{m=1}^M$
 - 2: **Initialization:** $t \leftarrow 1$ and $\mathbf{d}_t \leftarrow \frac{1}{M}\mathbf{1}_M$
 - 3: Solve for the dual variables $\boldsymbol{\alpha}_t$ in (4.8) by using SVM with the kernel matrix $\sum_{m=1}^M d_m \tilde{\mathbf{K}}_m$.
 - 4: **While** $t < T_{\max}$ **Do**
 - 5: $\mathbf{q}_t \leftarrow [\frac{1}{2}(\boldsymbol{\alpha}_t \circ \mathbf{y})' \mathbf{K}_1(\boldsymbol{\alpha}_t \circ \mathbf{y}), \dots, \frac{1}{2}(\boldsymbol{\alpha}_t \circ \mathbf{y})' \mathbf{K}_M(\boldsymbol{\alpha}_t \circ \mathbf{y})]'$
 - 6: $\mathbf{d}_t^{\text{new}} \leftarrow \theta(\mathbf{h}\mathbf{h}' + \epsilon \mathbf{I}_M)^{-1} \mathbf{q}_t$ and project $\mathbf{d}_t^{\text{new}}$ onto the feasible set $\mathcal{M}$.
 - 7: Update the base kernel combination coefficients $\mathbf{d}_{t+1}$ by using (4.11) with standard line search.
 - 8: Solve for the dual variables $\boldsymbol{\alpha}_{t+1}$ in (4.8) by using SVM with the kernel matrix $\sum_{m=1}^M d_m \tilde{\mathbf{K}}_m$.
 - 9: **If** $|G(\mathbf{d}_{t+1}) - G(\mathbf{d}_t)| \leq \tau$ **then** break
 - 10: $t \leftarrow t + 1$
 - 11: **End While**
 - 12: **Output:** $\mathbf{d}_t$ and $\boldsymbol{\alpha}_t$
-

By defining $\tilde{\mathbf{v}}_m = d_m \tilde{\mathbf{w}}_m$, we rewrite the optimization problem in (4.6) as a quadratic programming (QP) problem [147]:

$$\begin{aligned}
 J(\mathbf{d}) = \min_{\tilde{\mathbf{v}}_m, b, \xi_i} \quad & \frac{1}{2} \sum_{m=1}^M \frac{\|\tilde{\mathbf{v}}_m\|^2}{d_m} + C \sum_{i=1}^n \xi_i, \\
 \text{s.t.} \quad & y_i \left(\sum_{m=1}^M \tilde{\mathbf{v}}_m' \tilde{\phi}_m(\mathbf{x}_i) + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n.
 \end{aligned} \tag{4.7}$$

Note that the optimization problem in (4.7) is in the same form of DTMKL-AT as proposed in Section 3.3.5. Therefore, according to Theorem 3.1, the objective in (4.4) can reach its global minimum. And the optimization procedure of A-MKL can be analogously derived. Specifically, by introducing the Lagrangian multiplier $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]'$, we solve the dual form of the optimization problem in (4.7) as follows:

$$J(\mathbf{d}) = \max_{\boldsymbol{\alpha} \in \mathcal{A}} \mathbf{1}_n' \boldsymbol{\alpha} - \frac{1}{2} (\boldsymbol{\alpha} \circ \mathbf{y})' \left(\sum_{m=1}^M d_m \tilde{\mathbf{K}}_m \right) (\boldsymbol{\alpha} \circ \mathbf{y}), \tag{4.8}$$

where $\mathbf{y} = [y_1, \dots, y_n]'$ is the label vector of the training samples, $\mathcal{A} = \{\boldsymbol{\alpha} \in \mathbb{R}^n | \boldsymbol{\alpha}' \mathbf{y} = 0, \mathbf{0}_n \leq \boldsymbol{\alpha} \leq C \mathbf{1}_n\}$ is the feasible set of the dual variable $\boldsymbol{\alpha}$, $\tilde{\mathbf{K}}_m = [\tilde{\phi}_m(\mathbf{x}_i)' \tilde{\phi}_m(\mathbf{x}_j)] \in \mathbb{R}^{n \times n}$ is defined by the labeled training data from both domains, and $\tilde{\phi}_m(\mathbf{x}_i)' \tilde{\phi}_m(\mathbf{x}_j) =$

$\phi_m(\mathbf{x}_i)' \phi_m(\mathbf{x}_j) + \frac{1}{\lambda} \mathbf{f}(\mathbf{x}_i)' \mathbf{f}(\mathbf{x}_j)$. Recall that $\mathbf{f}(\mathbf{x})$ is a vector of the predictions on $\mathbf{x}$ from the pre-learned classifiers f_p 's, which resembles the label information of $\mathbf{x}$ and can be used to construct the *idealized* kernel [187]. Thus, the new kernel matrix $\tilde{\mathbf{K}}_m$ can be viewed as the integration of both the visual information (*i.e.*, from $\mathbf{K}_m$) and the label information, which can lead to larger discriminative power. Surprisingly, the optimization problem in (4.8) is in the same form as the dual form of SVM with the kernel matrix $\sum_{m=1}^M d_m \tilde{\mathbf{K}}_m$. Thus, the optimization problem can be solved by existing SVM solvers, such as LIBSVM [153].

4.4.3 Learning Algorithm of A-MKL

Similar to DTMKL presented in Chapter 3, we employ the reduced gradient descent procedures (see Algorithm 1 in Section 3.3.4) to iteratively update the linear combination coefficient $\mathbf{d}$ and the dual variable $\boldsymbol{\alpha}$ in (4.4). We detail such procedures designed for A-MKL as below.

Updating the dual variable $\boldsymbol{\alpha}$: Given the linear combination coefficient $\mathbf{d}$, we solve the optimization problem in (4.8) to obtain the dual variable $\boldsymbol{\alpha}$ by using LIBSVM [153].

Updating the linear combination coefficient $\mathbf{d}$: Suppose the dual variable $\boldsymbol{\alpha}$ is fixed. With respect to $\mathbf{d}$, the objective function $G(\mathbf{d})$ in (4.4) becomes:

$$\begin{aligned}
 G(\mathbf{d}) &= \frac{1}{2} \mathbf{d}' \mathbf{h} \mathbf{h}' \mathbf{d} + \theta \left(\mathbf{1}'_n \boldsymbol{\alpha} - \frac{1}{2} (\boldsymbol{\alpha} \circ \mathbf{y})' \left(\sum_{m=1}^M d_m \tilde{\mathbf{K}}_m \right) (\boldsymbol{\alpha} \circ \mathbf{y}) \right) \\
 &= \frac{1}{2} \mathbf{d}' \mathbf{h} \mathbf{h}' \mathbf{d} - \theta \mathbf{q}' \mathbf{d} + \text{const},
 \end{aligned} \tag{4.9}$$

where $\mathbf{q} = [\frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})' \mathbf{K}_1(\boldsymbol{\alpha} \circ \mathbf{y}), \dots, \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})' \mathbf{K}_M(\boldsymbol{\alpha} \circ \mathbf{y})]'$ and the last term is a constant term that is irrelevant to $\mathbf{d}$, namely, $\text{const} = \theta \left(\mathbf{1}'_n \boldsymbol{\alpha} - \frac{1}{2\lambda} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{f}(\mathbf{x}_i)' \mathbf{f}(\mathbf{x}_j) \right)$.

We adopt the second-order gradient method to update linear combination coefficient $\mathbf{d}$ at iteration $t + 1$ by:

$$\mathbf{d}_{t+1} = \mathbf{d}_t - \eta_t \mathbf{g}_t, \tag{4.10}$$

where η_t is the learning rate which can be obtained by using standard line search methods [147], $\mathbf{g}_t = (\nabla_t^2 G)^{-1} \nabla_t G$ is the updating direction, and $\nabla_t G = \mathbf{h} \mathbf{h}' \mathbf{d}_t - \theta \mathbf{q}$ and $\nabla_t^2 G = \mathbf{h} \mathbf{h}'$ are the first-order and second-order derivatives of G in (4.9) with respect

to $\mathbf{d}$ at the t -th iteration, respectively. Note that $\mathbf{h}\mathbf{h}'$ is not full rank, and therefore we replace $\mathbf{h}\mathbf{h}'$ by $\mathbf{h}\mathbf{h}' + \epsilon \mathbf{I}_M$ to avoid numerical instability, where ϵ is set as 10^{-5} in the experiments. Then, the updating function (4.10) can be rewritten as follows:

$$\mathbf{d}_{t+1} = (1 - \eta_t)\mathbf{d}_t + \eta_t \mathbf{d}_t^{\text{new}}, \quad (4.11)$$

where $\mathbf{d}_t^{\text{new}} = \theta(\mathbf{h}\mathbf{h}' + \epsilon \mathbf{I}_M)^{-1} \mathbf{q}$. Note that by replacing $\mathbf{h}\mathbf{h}'$ with $\mathbf{h}\mathbf{h}' + \epsilon \mathbf{I}_M$, the solution to $\nabla_t G = \mathbf{h}\mathbf{h}'\mathbf{d}_t - \theta \mathbf{q} = \mathbf{0}_M$ becomes $\mathbf{d}_t^{\text{new}}$. Given $\mathbf{d}_t \in \mathcal{M}$, we project $\mathbf{d}_t^{\text{new}}$ onto the feasible set $\mathcal{M}$ to ensure $\mathbf{d}_{t+1} \in \mathcal{M}$ as well.

The whole optimization procedure is summarized in Algorithm 2. We terminate the iterative updating procedure, once the objective in (4.4) converges or the number of iterations reaches $T_{\max}$. We set the tolerance parameter $\tau = 10^{-5}$ and $T_{\max} = 15$ in the experiments.

Note that by setting the derivative of the Lagrangian obtained from (4.6) with respect to $\tilde{\mathbf{w}}_m$ to zero, we obtain $\tilde{\mathbf{w}}_m = \sum_{i=1}^n \alpha_i y_i \tilde{\phi}_m(\mathbf{x}_i)$. Recall that $\sqrt{\lambda} \boldsymbol{\beta}$ and $\frac{1}{\sqrt{\lambda}} \mathbf{f}(\mathbf{x}_i)$ are the last P entries of $\tilde{\mathbf{w}}_m$ and $\tilde{\phi}_m(\mathbf{x}_i)$, respectively. Therefore, the linear combination coefficient $\boldsymbol{\beta}$ of the pre-learned classifiers can be obtained as follows:

$$\boldsymbol{\beta} = \frac{1}{\lambda} \sum_{i=1}^n \alpha_i y_i \mathbf{f}(\mathbf{x}_i). \quad (4.12)$$

With the optimal dual variables $\boldsymbol{\alpha}$ and linear combination coefficients $\mathbf{d}$, the target decision function (4.2) of our method A-MKL can be rewritten as

$$f^T(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i \left(\sum_{m=1}^M d_m \mathbf{K}_m(\mathbf{x}_i, \mathbf{x}) + \frac{1}{\lambda} \mathbf{f}(\mathbf{x}_i)' \mathbf{f}(\mathbf{x}) \right) + b.$$

4.4.4 Discussions with Related Work

A-SVM [30] assumes that the target classifier $f^T(\mathbf{x})$ is adapted from existing source classifiers $f_p^A(\mathbf{x})$'s. However, our proposed method A-MKL is different from A-SVM in several aspects as follows:

- 1) In A-SVM, the source classifiers are learned by using only the training samples from the source domain. In contrast, the pre-learned classifiers used in A-MKL can be learned by using the training samples either from the source domain or from both domains.

- 2) In A-SVM, the source classifiers are equally fused in the target classifier, *i.e.*, $f^T(\mathbf{x}) = \frac{1}{P} \sum_{p=1}^P f_p^A(\mathbf{x}) + \Delta f(\mathbf{x})$. In contrast, A-MKL learns the optimal combination coefficients β_p 's in (4.2).
- 3) In A-SVM, the perturbation function $\Delta f(\mathbf{x})$ is based on one single kernel, *i.e.*, $\Delta f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$. However, in A-MKL, the perturbation function $\Delta f(\mathbf{x}) = \sum_{m=1}^M d_m \mathbf{w}'_m \phi_m(\mathbf{x}) + b$ in (4.2) is based on multiple kernels, and the optimal kernel combination is automatically determined during the learning process.
- 4) A-SVM cannot utilize the unlabeled data in the target domain. In contrast, the valuable unlabeled data in the target domain are used in the MMD criterion of A-MKL for measuring the distribution mismatch between two domains.

Moreover, our work differs from Multi-Adaptive SVM (MA-SVM) [119] which is the subsequent extension of A-SVM [30]. MA-SVM improves A-SVM from pre-defining the combination coefficients of the source classifiers to automatically learning them. However, the same as A-SVM, MA-SVM uses only one kernel in the perturbation function. In contrast, our proposed method A-MKL employ MKL by using multiple base kernels.

Our work is also different from the prior work of DTSVM [36]¹, where the target decision function $f^T(\mathbf{x}) = \sum_{m=1}^M d_m \mathbf{w}'_m \phi_m(\mathbf{x}) + b$ is only based on multiple base kernels. In contrast, in A-MKL, we use a set of pre-learned classifiers $f_p(\mathbf{x})$'s as the parametric functions, and model the perturbation function $\Delta f(\mathbf{x})$ based on multiple base kernels in order to better fit the target decision function. To fuse multiple pre-learned classifiers, we also learn the optimal linear combination coefficients β_p 's. As shown in the experiments, our A-MKL is more robust in real applications by utilizing optimally combined classifiers as the prior.

MKL methods [143, 147] utilize the training data and the test data drawn from the same domain. When they come from different distributions, MKL methods may fail to learn the optimal kernel. This would degrade the classification performance in the target domain. On the contrary, A-MKL can better make use of the data from two domains to improve the classification performance.

¹Note that DTSVM is exactly the transfer learning method DTMKL_AT as introduced in Chapter 3. In the remainder of this chapter, we will continue using the old name DTSVM unless specifically mentioned.

4.5 Experiments

In this section, we first evaluate the effectiveness of the proposed Aligned Space-Time Pyramid Matching method. We then compare our proposed method Adaptive Multiple Kernel Learning (A-MKL) with the baseline SVM, and three existing transfer learning algorithms: Feature Replication (FR) [29], Adaptive SVM (A-SVM) [30], Multi-Adaptive SVM (MA-SVM) [119] and Domain Transfer SVM (DTSVM) [36], as well as a Multiple Kernel Learning (MKL) method discussed in [36]. We also analyze the learned combination coefficients β_p 's of the pre-learned classifiers, illustrate the convergence of the learning algorithm of A-MKL and investigate the performance variations of A-MKL using different proportions of labeled consumer videos. Moreover, we show that A-MKL using the pre-learned classifiers from all event classes is better than A-MKL using the pre-learned classifiers from one event class.

For all methods, we train one-versus-all classifiers with a fixed tradeoff parameter $C = 1$. For performance evaluation, we use the same non-interpolated Average Precision (AP) as in [159, 166] which corresponds to the multi-point average precision value of a precision-recall curve and incorporates the effect of recall. Mean Average Precision (MAP) is the mean of APs over all the event classes.

4.5.1 Data Set Description and Features

In our data set, part of the consumer videos are derived (under a usage agreement) from the Kodak Consumer Video Benchmark Data Set [165], which was collected by Kodak from about 100 real users over the period of one year. There are 1358 consumer video clips in the Kodak data set. A second part of the Kodak data set contains web videos from YouTube collected using keywords based search. After removing TV commercial videos and low-quality videos, there are 1873 YouTube video clips in total. An ontology of 25 semantic concepts were defined and keyframe based annotation was performed by the students at Columbia University to assign binary labels (presence or absence) for each visual concept for both sets of videos (see [165] for more details).

In this work, six events “birthday”, “picnic”, “parade”, “show”, “sports” and “wedding” are chosen for experiments. We additionally collected new consumer video clips

Table 4.1: Numbers of videos for six events in our data set.

	birthday	parade	picnic	show	sports	wedding	Total
Consumer videos	16	14	6	57	75	27	195
Web videos	151	119	85	200	260	91	906

from real users on our own. Similarly to [165], we also downloaded new YouTube videos from the website. Moreover, we also annotate the consumer videos to determine whether a specific event occurred by asking an annotator, who is not involved in the algorithmic design, to watch each video clip rather than just look at the key frames as done in [165]. For video clips in the Kodak consumer data set [165], only the video clips receiving positive labels in their keyframe based annotation are re-examined. We do not additionally annotate the YouTube videos² collected by ourselves and Kodak because in a real scenario we can only obtain loosely labeled YouTube videos and cannot use any further manual annotation. It should be clear that our consumer video set comes from two sources – the Kodak consumer video data set and our additional collection of personal videos, and our web video set is a combined set of YouTube videos as well. We confirm that the quality of YouTube videos is much lower than that of consumer videos directly collected from real users. Therefore, our data set is quite challenging for transfer learning algorithms. The detailed information on our data set is presented in Table 5.2. Note that our data set is a single-label data set, *i.e.*, each video belongs to only one event.

In real-world applications, the labeled samples in the target domain (*i.e.*, consumer video domain) are usually *much fewer* than those in the source domain (*i.e.*, web video domain). In this work, all 906 loosely labeled YouTube videos are used as the training data in the source domain. We randomly sample three consumer videos from each event (18 videos in total) as the labeled training videos in the target domain, and the remaining videos in the target domain are used as the test data. We sample the labeled target training videos for five times and report the means and standard deviations of MAPs or per-event APs for each method.

For all the videos in the data sets, we extract two types of features. The first one is the local space-time (ST) feature [166], in which 72-dimensional Histograms of Oriented

²The annotator felt that at least 20% of YouTube videos are incorrectly labeled after checking the video clips.

Table 4.2: Means and standard deviations (%) of MAPs over six events at different levels using SVM with the default kernel parameter for SIFT features.

	Gaussian	Laplacian	ISD	ID
Level-0	41.4 ± 3.7	44.2 ± 3.8	45.0 ± 3.5	46.2 ± 4.0
Level-1 (Unaligned)	43.0 ± 2.7	47.7 ± 1.7	49.0 ± 1.6	48.2 ± 1.5
Level-1 (Aligned)	50.4 ± 3.7	53.8 ± 1.8	52.9 ± 3.6	51.0 ± 2.5

Gradient (HoG) and 90-dimensional Histograms of Optical Flow (HoF) are extracted by using the online tool³. After that, they are concatenated together to form a 162-dimensional feature vector. We also sample each video clip at a rate of 2 frames per second to extract image frames from each video clip (we have 65 frames per video on average). For each frame, we extract 128-dimensional SIFT features from salient regions, which are detected by Difference-of-Gaussian (DoG) interest point detector [183]. On the average, we have 13847 ST features and 41441 SIFT features per video. Then, we build *visual vocabularies* by using k -means to group the ST features and SIFT features into 1000 and 2500 clusters, respectively.

4.5.2 Aligned Space-Time Pyramid Matching vs. Unaligned Space-Time Pyramid Matching

We compare our proposed Aligned Space-Time Pyramid Matching method discussed in Section 4.3 with the fixed volume-to-volume matching method (referred to as unaligned space-time pyramid matching) used in [166]. In [166], the space-time volumes of one video clip are matched with the volumes of the other video at the same spatial and temporal locations at each level. In other words, the second matching stage based on Integer-flow EMD is not applied and the distance between two video clips is equal to the sum of diagonal elements of the distance matrix, *i.e.*, $\sum_{r=1}^R D_{rr}$. For computational efficiency, we set the total number of levels $L = 2$ in this work. Therefore, we have two types of partitions, in which one video clip is divided into $1 \times 1 \times 1$ and $2 \times 2 \times 2$ space-time volumes, respectively.

We use the baseline SVM classifier based on the combined training data set from two domains (*i.e.*, consumer video domain and web video domain). We test the performances

³<http://www.irisa.fr/vista/Equipe/People/Laptev/download.html>

Table 4.3: Means and standard deviations (%) of MAPs over six events at different levels using SVM with the default kernel parameter for ST features.

	Gaussian	Laplacian	ISD	ID
Level-0	22.2 ± 1.8	36.1 ± 0.8	22.0 ± 3.8	35.6 ± 0.7
Level-1 (Unaligned)	20.1 ± 1.0	33.9 ± 0.6	21.8 ± 0.7	33.4 ± 0.7
Level-1 (Aligned)	20.6 ± 0.7	35.8 ± 1.7	22.3 ± 1.1	35.9 ± 1.8

with four types of kernels: Gaussian kernel (*i.e.*, $K(i, j) = \exp(-\gamma D^2(V_i, V_j))$), Laplacian kernel (*i.e.*, $K(i, j) = \exp(-\sqrt{\gamma} D(V_i, V_j))$), inverse square distance (ISD) kernel (*i.e.*, $K(i, j) = \frac{1}{\gamma D^2(V_i, V_j) + 1}$) and inverse distance (ID) kernel (*i.e.*, $K(i, j) = \frac{1}{\sqrt{\gamma} D(V_i, V_j) + 1}$), where $D(V_i, V_j)$ represents the distance between video V_i and V_j , and γ is the kernel parameter. We use the default kernel parameter $\gamma = \gamma_0 = \frac{1}{A}$, where A is the mean value of the square distances between all training samples as suggested in [166].

Tables 4.2 and 4.3 show the MAPs of the baseline SVM over six events for SIFT and ST features at different levels according to different types of kernels with the default kernel parameter. Based on the means of MAPs, we have the following three observations:

- 1) In all cases, the results at level-1 using aligned matching are better than those at level-0 based on SIFT features, which demonstrates the effectiveness of space-time partition and it is also consistent with the findings for prior pyramid matching methods [159, 166, 170, 171].
- 2) At level-1, our proposed Aligned Space-Time Pyramid Matching method outperforms the unaligned space-time pyramid matching method used in [166], thanks to the additional alignment of space-time volumes.
- 3) The results from space-time features are not as good as those from static SIFT features. As also reported in [188], a possible explanation is that the extracted ST features may fall on cluttered backgrounds because the consumer videos are generally captured by amateurs with hand-held cameras.

4.5.3 Performance Comparisons of Transfer Learning Methods

We compare our method A-MKL with other methods including the baseline SVM, FR, A-SVM, MKL and DTSVM. For the baseline SVM, we report the results of SVM_AT

and SVM-T, in which the labeled training samples are from two domains (*i.e.*, the source domain and the target domain) and only from the target domain, respectively. Specifically, The aforementioned four types of kernels (*i.e.*, Gaussian kernel, Laplacian kernel, ISD kernel and ID kernel) are adopted. In this experiment, we test A-MKL by using a set of kernel parameters, *i.e.*, $\gamma \in \mathcal{L} = \{-3, -2, \dots, 1\}$. Note that the total number of base kernels is $16|\mathcal{L}|$ from two pyramid levels and two types of local features, four types of kernels and $|\mathcal{L}|$ kernel parameters, where $|\mathcal{L}|$ is the cardinality of $\mathcal{L}$.

All methods are compared in three cases: (a) classifiers learned based on SIFT features; (b) classifiers learned based on ST features; and (c) classifiers learned based on both SIFT and ST features. We detail the experimental settings for all the methods as follows:

- For SVM-AT and FR (*resp.* SVM-T), we train $4|\mathcal{L}|$ independent classifiers with the corresponding $4|\mathcal{L}|$ base kernels for each pyramid level and each type of local features using the training samples from two domains (*resp.*, the training samples from target domain). And we further fuse the $4|\mathcal{L}|$ independent classifiers with equal weights to obtain the *average classifier* f_l^{SIFT} or f_l^{ST} , where $l = 0$ and 1 . For SVM-T, SVM-AT and FR, the final classifier is obtained by fusing average classifiers with equal weights (*e.g.*, $\frac{1}{2} (f_0^{SIFT} + f_1^{SIFT})$ for case (a), $\frac{1}{2} (f_0^{ST} + f_1^{ST})$ for case (b) and $\frac{1}{4} (f_0^{SIFT} + f_1^{SIFT} + f_0^{ST} + f_1^{ST})$ for case (c)).
- For A-SVM, we learn $4|\mathcal{L}|$ independent source classifiers for each pyramid level and each type of local features using the training data from the source domain and the corresponding $4|\mathcal{L}|$ base kernels, and then we independently learn four adapted target classifiers from two pyramid levels and two types of features by using the labeled training data from the target domain based on the Gaussian kernel with the default kernel parameter [30]. Similar to SVM-T, SVM-AT and FR, the final A-SVM classifier is obtained by fusing two (*resp.*, four) adapted target classifiers for cases (a) and (b) (*resp.*, case (c)).
- For MA-SVM, all $16|\mathcal{L}|$ independent source classifiers are learned as the same as for A-SVM. For each pyramid level and each type of local features, we obtain one average source classifier from $4|\mathcal{L}|$ source classifiers. Similar to A-SVM, we

independently learn four adapted target classifiers from two pyramid levels and two types of features by using four average source classifiers as well as the labeled training data from the target domain based on the Gaussian kernel with the default kernel parameter. After that, the final MA-SVM classifier is obtained by fusing two (*resp.*, four) adapted target classifiers for cases (a) and (b) (*resp.*, case (c)).

- For MKL and DTSVM, we simultaneously learn the linear combination coefficients of $8|\mathcal{L}|$ base kernels (for cases (a) or (b)) or $16|\mathcal{L}|$ base kernels (for case (c)) by using the combined training samples from both domains.
- For our method A-MKL, we make use of pre-learned classifiers as well as multiple base kernels (see (4.2) in Section 4.4.2). In the experiment, we consider each average classifier as one pre-learned classifier and learn the target decision function of A-MKL based on two average classifiers $f_l^{SIFT}|_{l=0}^1$ or $f_l^{ST}|_{l=0}^1$ for cases (a) or (b) (*resp.*, all the four average classifiers for case (c)) as well as $8|\mathcal{L}|$ base kernels based on SIFT or ST features for cases (a) or (b) (*resp.*, $16|\mathcal{L}|$ base kernels based on both types of features for case (c)). For A-MKL, we empirically fix $\theta = 10^{-5}$ and set $\lambda = 20$ for all three cases.

Considering that DTSVM and A-MKL can take advantage of both labeled and unlabeled data by using the MMD criterion to measure the mismatch in data distributions between two domains, we use semi-supervised setting in this work. More specifically, all the samples (including test samples) from the target domain and source domain are used to calculate $\mathbf{h}$ in (4.3). Note that all test samples are used as unlabeled data during the learning process.

Table 4.4 report the means and standard deviations of MAPs over all six events in three cases for all methods. From Tables 4.4, we have the following observations based on the means of MAPs:

- 1) The best result of SVM_T is worse than that of SVM_AT, which demonstrates that the learned SVM classifiers based on a limited number of training samples from the target domain are not robust. We also observe that SVM_T is always better than SVM_AT for cases (b) and (c). A possible explanation is that the ST feature is not

Table 4.4: Means and standard deviations (%) of MAPs over six events for all methods in three cases.

	SVM_T	SVM_AT	FR	A-SVM	MA-SVM	MKL	DTSVM	A-MKL
MAP-(a)	42.32 $\pm$ 5.50	53.93 $\pm$ 5.58	49.98 $\pm$ 5.63	38.42 $\pm$ 7.93	45.28 $\pm$ 7.25	47.19 $\pm$ 2.59	52.36 $\pm$ 1.88	57.14 $\pm$ 2.34
MAP-(b)	32.56 $\pm$ 2.08	24.73 $\pm$ 2.22	28.44 $\pm$ 2.61	24.95 $\pm$ 1.25	30.22 $\pm$ 1.97	35.34 $\pm$ 1.55	31.07 $\pm$ 2.60	37.24 $\pm$ 1.58
MAP-(c)	42.00 $\pm$ 4.94	36.23 $\pm$ 3.37	44.11 $\pm$ 3.57	32.40 $\pm$ 4.99	43.03 $\pm$ 5.18	46.92 $\pm$ 2.53	53.78 $\pm$ 2.99	58.20 $\pm$ 1.87

Figure 4.4: Means and standard deviations of per-event APs of six events for all methods.

robust enough so that the samples from the source and target domains distribute sparsely in the ST feature space. Therefore, it is more likely that the data from the source domain may degrade the event recognition performances in the target domain for cases (b) and (c).

- 2) In this application, A-SVM achieves the worst results in terms of the mean of MAPs, possibly because the limited number (*e.g.*, three samples per event) of labeled training samples from the target domain are not sufficient for A-SVM to robustly learn an adapted target classifier which is based on only one kernel. MA-SVM performs better than A-SVM, because of the automatic learning of the combination coefficients for the source average classifiers. However, the best result of MA-SVM in different settings is worse than those of SVM_AT, MKL and FR, which again shows that it is not sufficient to learn a robust target classifier by only using a limited number of labeled training samples from the target domain.
- 3) DTSVM is generally better than MKL in terms of the mean of MAPs. This is consistent with [36].
- 4) For all methods, the MAPs based on SIFT features are better than those based on ST features. In practice, the simple ensemble method, SVM_AT, achieves good performances when only using the SIFT features in case (a). It indicates that SIFT features are more effective for event recognition in consumer videos. However, the MAPs of SVM_AT, FR and A-SVM in case (c) are much worse compared with case (a). It suggests that the simple late fusion methods using equal weights are not robust for integrating strong features and weak features. In contrast, for DTSVM and our method A-MKL, the results in case (c) are improved by learning optimal linear combination coefficients to effectively fuse two types of features.
- 5) For each of three cases, our proposed method A-MKL achieves the best performance by effectively fusing four average classifiers (from two pyramid levels and two types of local features) and multiple base kernels as well as reducing the mismatch in the data distributions between two domains. We also believe the utilization of multiple base kernels and pre-learned average classifiers can also well cope with

Figure 4.5: Illustration of the combination coefficients β_p 's of the pre-learned classifiers for all events.

YouTube videos with noisy labels. In Table 4.4, compared with the best means of MAPs of SVM-T (42.32%), SVM-AT (53.93%), FR (49.98%), A-SVM (38.42%), MKL (47.19%) and DTSVM (53.78%), the relative improvements of our best result (58.20%) are 37.52%, 7.92%, 16.54%, 51.48%, 23.33% and 8.22%, respectively.

In Figure 4.4, we plot the means and standard deviations of per-event APs of all methods. Our method achieves the best performances in 3 out of 6 events in case (c) and some concepts enjoy large performance gains according to the means of per-event APs, *e.g.*, the AP of “parade” significantly increases from 65.96% (DTSVM) to 75.21% (A-MKL).

4.5.4 Analysis on the Combination Coefficients of the Pre-learned Classifiers

Recall that we learn the linear combination coefficients β_p 's of the pre-learned classifiers f_p 's in A-MKL⁴. And the absolute value of each β_p reflects the importance of the corresponding pre-learned classifier. Specifically, the larger β_p is, the more f_p contributes in the target decision function. For better representation, let us denote the corresponding average classifiers f_0^{SIFT} , f_1^{SIFT} , f_0^{ST} and f_1^{ST} as f_1 , f_2 , f_3 and f_4 , respectively.

⁴Note that β can be computed by using (4.12) after solving the optimization problem in A-MKL.

Figure 4.6: Illustration of the convergence of A-MKL learning algorithm for all events.

Taking one round of training/test data split in the target domain for example, we draw the combination coefficients β_p 's of the four pre-learned classifiers f_p 's for all events in Figure 4.5. In this experiment, we again set $\mathcal{L} = \{-3, -2, \dots, 1\}$. We observe that the absolute values of β_1 and β_2 are always much larger than those of β_3 and β_4 , which shows that the pre-learned classifiers (*i.e.*, f_1 and f_2) based on SIFT features play dominant roles among all the pre-learned classifiers. This is not surprising, because SIFT features are much more robust than ST features as demonstrated in Section 4.5.3. From Figure 4.5, we also observe that the values of β_3 and β_4 are generally not close to zero, which demonstrates that A-MKL can further improve the event recognition performance by effectively integrating strong and weak features. Recall that A-MKL using both types of features outperforms A-MKL with only SIFT features (see Table 4.4). We have similar observations for other rounds of experiments.

4.5.5 Convergence of A-MKL Learning Algorithm

Recall that we iteratively update the dual variable α and the linear combination coefficient $\mathbf{d}$ in A-MKL (see Section 4.4.3). We take one round of training/test data split as

Table 4.5: Means and standard deviations (%) of MAPs of A-MKL (referred to as A-MKL_4) using the pre-learned average classifiers from the same event class and A-MKL (referred to as A-MKL_24) using the pre-learned average classifiers from all six event classes. Different sets of kernel parameters (*i.e.*, $\mathcal{H}$) are employed to obtain the pre-learned average classifiers.

	$\mathcal{H} = \{-3, -2, \dots, 1\}$	$\mathcal{H} = \{-4, -3, \dots, 1\}$	$\mathcal{H} = \{-5, -4, \dots, 1\}$	$\mathcal{H} = \{-6, -5, \dots, 1\}$
A-MKL_4	58.20 ± 1.87	58.33 ± 2.33	58.56 ± 2.53	57.16 ± 3.02
A-MKL_24	58.74 ± 3.30	59.04 ± 3.53	59.25 ± 3.73	59.28 ± 2.57

an example to discuss the convergence of the iterative algorithm of A-MKL, in which we also set $\mathcal{L}$ as $\{-3, -2, \dots, 1\}$, and we use both types of features. In Figure 4.6, we plot the change of the objective value of A-MKL with respect to the number of iterations. We observe that A-MKL converges after about eight iterations for all events. We have similar observations for other rounds of experiments.

4.5.6 Utilization of Additional Pre-learned Classifiers from Other Event Classes

In the previous experiments, for a specific event class, we only utilize the pre-learned classifiers (*i.e.*, average classifiers $f_t^{SIFT}|_{l=0}^1$ and $f_t^{ST}|_{l=0}^1$) from this event class. As a general learning scheme, A-MKL can readily incorporate additional pre-learned classifiers. In our event recognition application, we observe that some events may share common motion patterns. For example, the videos from some events (like “birthday”, “picnic” and “wedding”) usually contain a number of people talking with each other. Thus, it is beneficial to learn an adapted classifier for “birthday” by leveraging the pre-learned classifiers from “picnic” and “wedding”. Based on this observation, for each event, we make use of the pre-learned classifiers from all event classes for the learning of the adapted classifier in A-MKL. Therefore, the total number of the pre-learned classifiers is 24 for each event when using both types of features. For better representation, we refer to A-MKL with four pre-learned average classifiers discussed in Sections 4.5.3, 4.5.4 and 4.5.5 (*resp.*, A-MKL with all 24 pre-learned average classifiers) as A-MKL_4 (*resp.*, A-MKL_24).

In Sections 4.5.3, 4.5.4 and 4.5.5, the same kernel parameter set (*i.e.*, $\mathcal{L} = \{-3, -2, \dots, 1\}$) is used for the base kernels and also employed to obtain the pre-learned average classifiers in A-MKL. In this experiment, we also use the same set of kernel parameters (*i.e.*,

Figure 4.7: Means and standard deviations of MAPs over six events for SVM_T, DTSVM, A-MKL_4 and A-MKL_24 when using different proportions (*i.e.*, r) of labeled training consumer videos.

$\mathcal{L} = \{-3, -2, \dots, 1\}$) for the base kernels but we additionally vary the set of kernel parameters (denoted as $\mathcal{H}$ for better representation) to obtain the pre-learned average classifiers for A-MKL_4 and A-MKL_24. Specifically, for each pyramid level and each type of features, we learn $4|\mathcal{H}|$ independent SVM classifiers from the parameter set $\mathcal{H}$ and four types of kernels (*i.e.*, Gaussian kernel, Laplacian kernel, ISD kernel and ID kernel) by using the training samples from both the source and target domains, which are further averaged to obtain one pre-learned classifier (*i.e.*, $f_l^{SIFT}|_{l=0}^1$ or $f_l^{ST}|_{l=0}^1$).

In Table 4.5, we compare the results of A-MKL_4 and A-MKL_24 when using 1) $\mathcal{H} = \{-3, -2, \dots, 1\}$; 2) $\mathcal{H} = \{-4, -3, \dots, 1\}$; 3) $\mathcal{H} = \{-5, -4, \dots, 1\}$ and 4) $\mathcal{H} = \{-6, -5, \dots, 1\}$. From Table 4.5, We observe that while the performances of A-MKL_4 and A-MKL_24 change when using different $\mathcal{H}$, A-MKL_24 is consistently better than A-MKL_4 in terms of the mean of MAPs. It clearly demonstrates that A-MKL can learn a more robust target classifier by effectively leveraging the pre-learned average classifiers from all the event classes. The performance of A-MKL_24 is the best, when setting $\mathcal{H} = \{-6, -5, \dots, 1\}$. Compared with the other methods such as SVM_T, SVM_AT, FR, A-SVM, MKL and DTSVM in terms of the mean of per-event APs for case (c), A-MKL_24 achieves the best performances in 4 out of 6 events. The relative improvements of the best mean of MAPs from A-MKL_24 (59.28%) over those from SVM_AT (53.93%) and DTSVM (53.78%) in Table 4.4 are 9.92% and 10.23%, respectively.

4.5.7 Performance Variations of A-MKL using Different Proportions of Labeled Consumer Videos

We also investigate the performance variations of A-MKL when using different proportions of labeled training samples from the target domain. Specifically, we randomly choose a proportion (*i.e.*, r) of positive samples from the target domain for each event class. All the randomly chosen samples are considered as the labeled training data from the target domain, while the remainder of samples in the target domain are used as the test data. Again, we sample the labeled target training videos for five times and report the means and standard deviations of MAPs. Considering that the users are reluctant to annotate a large number of consumer videos, we set r as 5%, 10%, 20% and 30%. By using both the SIFT and ST features (*i.e.*, case (c)), we compare our methods A-MKL_4 and A-MKL_24 with the baseline method SVM.T and the existing transfer learning method DTSVM that achieves the second best results in case (c) (see Tables 4.4). For DTSVM, A-MKL_4 and A-MKL_24, we use the same setting as in Section 4.5.6 by setting the kernel parameter set $\mathcal{L}$ for the base kernels as $\{-3, -2, \dots, 1\}$. For A-MKL_24, we additionally set the kernel parameter set $\mathcal{H}$ for the pre-learned average classifiers as $\{-6, -5, \dots, 1\}$, with which A-MKL_24 achieves the best result (see Table 4.5).

From Figure 4.7, we have the following observations based on the mean of MAPs. First, the results of all methods generally increase, when using more labeled training samples from the target domain. Second, the transfer learning methods DTSVM, A-MKL_4 and A-MKL_24 consistently outperform the baseline method SVM.T. Third, our methods A-MKL_4 and A-MKL_24 consistently perform better than DTSVM, which shows the effectiveness of the utilization of pre-learned average classifiers. Finally, A-MKL_24 is consistently better than A-MKL_4, which demonstrates that the information from other event classes is helpful for improving the event recognition performance for individual classes.

4.5.8 Running Time and Memory Usage

Finally, we report the running time and memory usage of our proposed framework. All the experiments are conducted on a server machine with Intel Xeon 3.33GHz CPUs and 32GB RAM by using a single thread. The main costs in running time and memory

usage are from feature extraction and our proposed ASTPM method. Specifically, on the average it takes about 63.3 seconds (*resp.*, 246.5 seconds) to extract the SIFT features (*resp.*, ST features) from a one-minute-long video. For each video, its SIFT features (*resp.*, ST features) occupy 41.7 megabytes (*resp.*, 17.9 megabytes) on the average. In this work, each type of features are vector-quantized into visual words by using k -means. Considering the quantization process for the SIFT and ST features from training videos can be conducted in an offline manner and the quantization process for the SIFT and ST features from a test video is very fast, we do not count the running time of this process. For our ASTPM using the SIFT and ST features, it respectively takes about 20.9 milliseconds and 0.1 milliseconds (*resp.*, 1213.6 milliseconds and 0.4 milliseconds) to calculate the distance between a pair of videos at level-0 (*resp.*, level-1) on the average. For each event class, on the average it takes about 68.4 seconds to learn one A-MKL classifier, which includes 7.1 seconds for obtaining the pre-learned average classifiers. The average prediction time for each test video is only about 11 milliseconds. To accelerate our framework for a median to large scale video event recognition task, we can extract the SIFT and ST features by using multiple threads in a parallel fashion and employ the fast EMD algorithm [189] in ASTPM.

4.6 Summary

In this chapter, we propose a new event recognition framework for consumer videos by leveraging a large amount of loosely labeled YouTube videos. Specifically, we propose a new pyramid matching method called Aligned Space-Time Pyramid Matching (ASTPM) and a novel transfer learning method, referred to as Adaptive Multiple Kernel Learning (A-MKL), to better fuse the information from multiple pyramid levels and different types of local features and to cope with the mismatch between the feature distributions of consumer videos and web videos. Experiments clearly demonstrate the effectiveness of our framework. To the best of our knowledge, our work is the first to perform event recognition in consumer videos by incorporating cost-effective transfer learning.

To put it in a larger perspective, our work falls into the recent research trend of “Internet Vision”, where the massive web data including images and videos together

with rich and valuable contextual information (e.g., tags, categories and captions) are employed for various computer vision and computer graphics applications such as image annotation [52, 190], image retrieval [15], scene completion [191], and so on. By treating the “*web data as the king*”, these methods [190, 191] have achieved promising results by adopting the simplistic learning methods such as the k NN classifier. In this work, we have demonstrated that it is beneficial to *learn from web data* by developing more advanced machine learning methods (specifically, the transfer learning method A-MKL in this work) to further improve the classification performances. A possible future research direction is to develop effective methods to select more useful videos from a large number of low-quality YouTube videos to construct the source domain.

While transfer learning has been studied for years in other fields (*e.g.*, natural language processing [23, 29]), it is still an emerging research topic in computer vision [16]. In some vision applications, there is an existing domain (*i.e.*, source domain) with a large number of labeled data but we want to recognize the images or videos in another domain of interest (*i.e.*, target domain) with very few labeled samples. Besides the adaption between the web domain and consumer domain studied in this work and [15], other examples that vision researchers are recently working on include the adaptation of cross-category knowledge to a new category domain [77], the knowledge transfer by mining semantic relatedness [114], and the adaption between two domains with different feature representations [16, 39]. In the future, we will extend our A-MKL for these interesting vision applications.

Chapter 5

Domain Adaptation Machine

In this chapter, we look into the transfer learning problem where there exist multiple source domains. We propose a novel domain-dependent regularization framework, referred to as Domain Adaptation Machine (DAM), for the multiple source domain adaptation problem. In this framework, first obtain a set of pre-learned base classifiers which can be trained by existing machine learning methods such as support vector machine (SVM). With the base classifiers, we then 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 samples from the target domain. Based on our framework, we develop two new domain adaptation methods referred to as FastDAM and UniverDAM, respectively. We evaluate our proposed methods FastDAM and UniverDAM on two multiple source domain adaptation related applications.

5.1 Introduction

It is well known that the collection of labeled samples 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 [23, 29, 30, 33, 54] to learn robust classifiers with only a few or even no labeled samples 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 samples collected from other domains are

also useful for classifying the samples from the target domain in many real applications, such as sentiment classification, text categorization, WiFi localization and video concept detection.

Supervised domain adaptation methods [23, 29, 30] have been proposed by utilizing all labeled training instances from the source and target domains. However, numerous unlabeled samples in the target domain are not exploited in the above transfer learning methods [29, 30, 33]. As shown in [7, 13, 36, 54, 142, 192, 193], such unlabeled samples can also be employed to improve the generalization performance. When there are only a few or even no labeled samples available in the target domain, the classifiers can be trained with the samples from the source domains. In such an extreme case, several domain adaptation methods [44, 47, 194, 195] were proposed to cope with the inconsistency of data distribution (such as *covariate shift* [44] or *sampling selection bias* [47]). These methods re-weighted the training samples from the source domain by leveraging the unlabeled data from the target domain such that the statistics of samples from both domains are matched.

Recently, several domain adaptation methods [54, 98] were proposed to learn robust classifiers with the diverse training data from multiple source domains. Luo *et al.* [98] 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 [31], which is a well-known open problem termed as *negative transfer* [196]. Moreover, Some researchers also theoretically studied the domain adaptation problem [63, 64, 66–69] where there exist multiple source domains.

In this chapter, 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 Domain Adaptation Machine (DAM) to learn a robust decision function (referred to as *target classifier*) for label prediction of samples in the target domain by leveraging a set of pre-learned classifiers (referred to as *base classifiers*). In our framework, any classifier such as the standard SVM classifier learned with the labeled samples from the source domains or the FR classifier [29] learned with the labeled samples from the source domains and the target domain can be readily used as the base classifier.

Motivated from Manifold Regularization (MR) [192] and the graph based Multi-Task Learning (MTL) [197–199], 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 samples of the target domain with the pre-learned base classifiers from relevant source domains. This newly proposed regularizer can be readily introduced to many kernel methods such as SVM, SVR, least-squares SVM (LS-SVM) and so on, and extend these algorithms to the corresponding domain adaptation methods. To the best of knowledge, the smoothness assumption encoded in the domain-dependent regularizer is the first to be introduced into domain adaptation.

Under this framework, we develop two new methods referred to as FastDAM and UniverDAM. Motivated from Manifold 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. The recent work [200, 201] indicated that the samples which do not belong to the positive class and the negative class can be used as an additional data collection called *Universum* [26] to improve the generalization ability of SVM for the binary classification task (see Appendix A.4 for a introduction of Universum). However, how to choose/construct Universum is problem-dependent. For example, Weston *et al.* [201] experimentally demonstrated that one can make use of symbols (*e.g.*, uppercase and lowercase letters) as Universum for digit classification. Note that while the samples from the source domains are with different data distributions of those from 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 samples 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 suggested in [200, 201] which is related to the decision values of the pre-learned base classifiers on the samples from the source domains into LS-SVM for UniverDAM.

We evaluate our two DAM-based methods in two multiple domain adaptation related applications: video concept detection and document retrieval. In the video concept detection task, the experimental results on the large TRECVID 2005 data set 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 data sets. The comprehensive experiments on the two data sets also demonstrate the effectiveness of FastDAM and UniverDAM. UniverDAM achieves the best document retrieval performances on both data sets because of the successful utilization of *Universum* (*i.e.*, the samples from multiple source domains).

A preliminary version of this work appeared in [54] which focused on the FastDAM algorithm using source classifiers (*i.e.*, SVMs learned with the labeled samples from source domains) as the base classifiers. In this chapter, we additionally propose a new method called UniverDAM and discuss the connection between FastDAM and UniverDAM. We also present more details about the non-sparse solution of FastDAM in order to better motivate the utilization of the sparse regularizer based on the ϵ -insensitive loss. Moreover, we further employ the FR classifiers [29], which are learned with the labeled samples from the source domains and the target domain, as the base classifiers in our DAM framework and report more experimental results using two new data sets (*i.e.*, 20 Newsgroups and email spam) for document retrieval.

The rest of this chapter is organized as follows. We briefly review the related work in Section 5.2. We then introduce our proposed framework DAM and two methods FastDAM and UniverDAM in Section 5.3. The connections between the proposed two methods and the related works are discussed in Section 5.4. The experimental results are reported in Section 5.5. Finally, we conclude this chapter in Section 5.6.

5.2 Brief Review of Related Work

Let us represent the labeled and unlabeled samples from the target domain as $\mathcal{D}_l^T = (\mathbf{x}_i^T, y_i^T)_{i=1}^{n_l}$ and $\mathcal{D}_u^T = \mathbf{x}_i^T|_{i=n_l+1}^{n_l+n_u}$, respectively, where y_i^T is the label of $\mathbf{x}_i^T$. We also define

$\mathcal{D}^s = (\mathbf{x}_i^s, y_i^s)_{i=1}^{n_s}$ as the data set from the s -th source domain, where $s = 1, \dots, P$ and P is the total number of source domains. Moreover, we denote the data set from the target domain as $\mathcal{D}^T = \mathcal{D}_l^T \cup \mathcal{D}_u^T$ 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 $\mathbf{0}_n, \mathbf{1}_n \in \mathbb{R}^n$ as the column vectors of all zeros and all ones, respectively. The inequality $\mathbf{u} = [u_1, \dots, u_n]' \geq \mathbf{v} = [v_1, \dots, v_n]'$ means that $u_i \geq v_i$ for $i = 1, \dots, n$. And $\text{diag}(\mathbf{u}')$ represents a diagonal matrix with u_i as its i -th diagonal entry.

5.2.1 Multiple Domain Transfer via Existing Classifiers

Yang *et al.* [30] proposed adaptive support vector machine (A-SVM), in which a new SVM classifier $f^T(\mathbf{x})$ is adapted from the existing source classifiers $f^s(\mathbf{x})$'s trained with the samples from the auxiliary sources. Specifically, the new decision function is formulated as:

$$f^T(\mathbf{x}) = \sum_{s=1}^P \gamma_s f^s(\mathbf{x}) + \Delta f(\mathbf{x}), \quad (5.1)$$

where the perturbation function $\Delta f(\mathbf{x})$ is learned by using the labeled data $\mathcal{D}_l^T$ from the target domain, and $\gamma_s \in [0, 1]$ is the weight of each source classifier f^s and $\sum_{s=1}^P \gamma_s = 1$. Usually in domain adaptation problems, there are only a limited number of labeled training samples from the target domain. Therefore, the adaptation process of A-SVM (*i.e.*, learning of the target decision function) can be very fast. In [30], the equal weights are used for all source classifiers in the experiments. As shown in [30], 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 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 source 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 i -th instance from the s -th source domain. Then the decision function (5.1) becomes:

$$f^T(\mathbf{x}) = \sum_{s=1}^P \gamma_s \sum_{i=1}^{n_s} \alpha_i^s y_i^s k(\mathbf{x}_i^s, \mathbf{x}) + \sum_{i=1}^{n_l} \alpha_i^T y_i^T k(\mathbf{x}_i^T, \mathbf{x}), \quad (5.2)$$

which is the sum of a set of weighted kernel evaluations between the test instance $\mathbf{x}$ and all labeled samples $\mathbf{x}_i^T$ and $\mathbf{x}_i^s$ respectively from the target domain and all the source

domains. Thus, the prediction using (5.2) is inefficient in the large-scale applications with a large amount of test samples. In addition, it is unclear how to use the valuable unlabeled data $\mathcal{D}_u^T$ in the target domain in A-SVM.

Schweikert *et al.* [31] also made use of the pre-learned classifiers for domain adaptation. They formally presented a so-called multiple convex combination method to linearly combine source classifiers together with the target classifier. Similar to A-SVM, each source 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 samples only from the target domain. Then the final classifier $f(\mathbf{x})$ is formulated as follows:

$$f(\mathbf{x}) = \gamma f^T(\mathbf{x}) + \frac{1-\gamma}{P} \sum_{s=1}^P f^s(\mathbf{x}), \quad (5.3)$$

where γ is the weight to balance the two terms.

5.2.2 Regularizations

Belkin *et al.* [192] extended Regularized Least Squares (RLS) and SVM to Laplacian Regularized Least Squares (LapRLS) and Laplacian SVM (LapSVM) for semi-supervised learning 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 semi-supervised learning). Let us denote G as an undirected weighted graph with a vertex set and a similarity matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, in which n is the total number of labeled and unlabeled training samples and each element w_{ij} of the real symmetric matrix W represents the similarity of a pair of vertices. The proposed regularizer in [192] is as follows:

$$\Omega_{manifold}(f) = \sum_{i,j=1}^n w_{ij} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2, \quad (5.4)$$

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

A graph based regularizer is also proposed in [197–199] for Multi-Task Learning (MTL). It is based on two MTL functions f^i and f^j of the i -th task and the j -th task in the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$:

$$\Omega_{\mathcal{G}}(f^1, f^2, \dots, f^K) = \sum_{i,j: (f^i, f^j) \in \mathcal{G}} \gamma_{ij} \|f^i - f^j\|_{\mathcal{H}}^2, \quad (5.5)$$

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

Weston *et al.* [201] proposed a new data-dependent regularizer in combination with SVM for binary-class problems, in which an additional data set is employed in the learning process to improve the generalization ability of the learned classifier. This additional data set which does not belong to the positive class and the negative class is called as *Universum* as suggested by Vapnik in [26]. In [201], Weston *et al.* minimized the classification error, as well as minimized a data-dependent regularizer based on the Universum which is equivalent to maximizing the number of contradictions on the Universum¹. The Universum regularizer is formulated as follows:

$$\Omega_{\mathcal{U}}(f) = \sum_{i=1}^n |f(\mathbf{x}_i)|^2, \quad (5.6)$$

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

5.3 Domain Adaptation Machine Framework

In this section, we introduce our proposed framework referred to as Domain Adaptation Machine (DAM) as well as two methods FastDAM and UniverDAM, for multiple source domain adaptation.

5.3.1 Smoothness Assumption for Domain Adaptation

In manifold regularization [192], the decision function in (5.4) is enforced to be smooth on the data manifold, namely, the two nearby samples in a high-density region should

¹see Appendix A.4 for an introduction of Universum.

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 pre-computed base classifiers. For the i -th 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 s -th base classifier. In our DAM framework, any classifier can be readily used as the base classifier. 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 samples from the s -th source domain; and 2) the feature replication (FR) classifier trained with the labeled samples from the s -th source domain and the target domain. For the unlabeled target samples $\mathcal{D}_u^T$ in the target domain, let us define the decision values from the target classifier and the s -th base classifier as $\mathbf{f}_u^T = [f_{n_l+1}^T, \dots, f_{n_T}^T]'$ and $\mathbf{f}_u^s = [f_{n_l+1}^s, \dots, f_{n_T}^s]'$ respectively.

Let us also define γ_s as the weight for measuring the distribution relevance between the s -th source domain and the target domain (see Section 5.5.2 for more discussions on γ_s). If the s -th source domain and the target domain are relevant (*i.e.*, γ_s is large), we enforce f_i^s should be close to f_i^T on the unlabeled samples in the target domain. Note that the source domains are assumed to be independent from each other in this work.

Motivated by manifold regularization [192] and graph based multi-task learning [197–199] (see the corresponding regularizers in Section 5.2.2), we propose the following domain-dependent regularizer for the target classifier f^T in Definition 5.7.

Definition 5.7 Domain-Dependent Regularizer for Domain Adaptation:

$$\Omega_{\mathcal{A}}(f^T) = \sum_{s=1}^P \gamma_s \sum_{i=n_l+1}^{n_T} (f_i^T - f_i^s)^2 = \sum_{s=1}^P \gamma_s \|\mathbf{f}_u^T - \mathbf{f}_u^s\|^2. \quad (5.7)$$

It is worth mentioning the difference between the regularizers defined in our DAM and MTL in the following two aspects:

- 1) MTL simultaneously learns all task functions $f^1, \dots, f^K$ (see (5.5)) and each two of the task functions are compared in the same RKHS $\mathcal{H}$. In contrast, the base classifiers f^s 's in (5.7) 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 base classifiers and the target classifier in DAM.

Figure 5.1: Base classifiers learned by using the labeled training samples from the source domains (and the target domain as well). For each unlabeled instance $\mathbf{x}$ in $\mathcal{D}_u^T$, 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 $\tilde{\gamma}_s = \frac{\gamma_s}{\sum_{s=1}^P \gamma_s}$.

- 2) It is still unclear how to exploit the unlabeled samples through the regularizer (5.5) in MTL. In contrast, the unlabeled samples $\mathcal{D}_u^T$ from the target domain are used in DAM (see Figure 5.1 and (5.7)).

As shown in our experiments in Section 5.5, the proposed domain-dependent regularizer generally works well in some real-world data sets like the TRECVID, 20 Newsgroups and email spam data sets.

5.3.2 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 $\Omega_A(f^T)$ in (5.7) and the data-dependent regularizer $\Omega_u(f^T)$ in (5.6) for the Universum. The proposed framework, Domain Adaptation Machine (DAM), is then formulated as follows:

$$\min_{f^T} \Omega(f^T) + \lambda_L \Omega_L(f^T) + \lambda_D \Omega_D(f^T), \quad (5.8)$$

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 samples of the target domain and the last term $\Omega_D(f^T)$ represents different regularizers such as $\Omega_{\mathcal{A}}(f^T)$ and $\Omega_{\mathcal{U}}(f^T)$. 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 work, we model $\Omega_L(f^T)$ in (5.8) as the square error of the target classifier f^T on the labeled samples $\mathcal{D}_l^T$ in the target domain, which is analogous to the least-squares SVM (LS-SVM) [202]. Note that the experimental results in [202] 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 (5.7) to model $\Omega_D(f^T)$. In UniverDAM, we additionally incorporate the regularizer $\Omega_{\mathcal{U}}(f^T)$ in (5.6) by treating the samples from the source domains as Universum.

5.3.3 Domain Adaptation Machine with Fast Prediction

With the domain-dependent regularizer $\Omega_{\mathcal{A}}(f^T)$ in (5.7), we rewrite (5.8) as follows:

$$\min_{f^T} \Omega(f^T) + \frac{\lambda_D}{2} \sum_{s=1}^P \gamma_s \sum_{i=n_l+1}^{n_T} (f_i^T - f_i^s)^2 + \frac{\lambda_L}{2} \sum_{i=1}^{n_l} (f_i^T - y_i^T)^2. \quad (5.9)$$

5.3.3.1 Non-sparse Solution

Theorem 5.3 *Assume that the target decision function is in the form of $f^T(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x})$ and the regularizer $\Omega(f^T) = \frac{1}{2}\|\mathbf{w}\|^2$. Then, the solution f^T of the optimization problem (5.9) is*

$$f^T(\mathbf{x}) = \lambda_D \sum_{s=1}^P \gamma_s \sum_{i=n_l+1}^{n_T} f^s(\mathbf{x}_i^T) \tilde{k}(\mathbf{x}_i^T, \mathbf{x}) + \sum_{i=1}^{n_l} \alpha_i^T \tilde{k}(\mathbf{x}_i^T, \mathbf{x}), \quad (5.10)$$

where

$$\tilde{k}(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) - \mathbf{k}'_{\mathbf{x}_i} (\mathbf{I}_{n_u} + \mathbf{M}\mathbf{K}_u)^{-1} \mathbf{M}\mathbf{k}_{\mathbf{x}_j} \quad (5.11)$$

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 $\phi(\mathbf{x}_j)$, $\mathbf{k}_{\mathbf{x}} = [k(\mathbf{x}_{n_l+1}^T, \mathbf{x}), \dots, k(\mathbf{x}_{n_T}^T, \mathbf{x})]'$, $\mathbf{K}_u = [k(\mathbf{x}_i^T, \mathbf{x}_j^T)] \in$

$\mathbb{R}^{n_u \times n_u}$ 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 i -th labeled samples in the target domain.

Proof: The proof is given in Appendix A.2.

Note that similar to [197, 199], the solution of the target decision function f^T is non-sparse. All the base classifiers f^s 's need to be used for predicting the labels of the target samples, making it inefficient for large-scale applications (*e.g.*, video concept detection). Moreover, similar to the manifold kernel defined in [203], the kernel for domain adaptation (5.11) 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.

5.3.3.2 Sparse Solution

As shown in [204, 205], the use of the ϵ -insensitive loss function in Support Vector Regression (SVR) can usually lead to a sparse representation of the decision function². To obtain the sparse solution, we therefore introduce an additional term in (5.9), which regulates the approximation quality and the sparsity of the decision function. Moreover, we also assume that the regularizer $\Omega(f^T) = \frac{1}{2} \|\mathbf{w}\|^2$ for the penalty of function complexity of f^T . The optimization problem (5.9) is then rewritten as:

$$\begin{aligned} \min_{f_l^T, \mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{n_T} \ell_\epsilon(\mathbf{w}'\phi(\mathbf{x}_i) + b - f_i^T) + \frac{\lambda_L}{2} \|\mathbf{f}_l^T - \mathbf{y}_l^T\|^2 \\ & + \frac{\lambda_D}{2} \sum_{s=1}^P \gamma_s \|\mathbf{f}_u^T - \mathbf{f}_u^s\|^2, \end{aligned} \quad (5.12)$$

where C is another tradeoff parameter, $\mathbf{f}_l^T = [f_1^T, \dots, f_{n_l}^T]'$ is the vector of the target decision function on the labeled samples $\mathcal{D}_l^T$ from the target domain, $\mathbf{y}_l^T = [y_1^T, \dots, y_{n_l}^T]'$ is the label vector of the labeled training samples in the target domain, and $\ell_\epsilon(t)$ is the ϵ -insensitive loss: $\ell_\epsilon(z) = \begin{cases} |z| - \epsilon, & \text{if } |z| > \epsilon; \\ 0, & \text{otherwise.} \end{cases}$ Since ϵ -insensitive loss is non-smooth,

²For 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, please refer to [204].

(5.12) is usually transformed as a constrained optimization problem, that is:

$$\begin{aligned} \min_{f_i^T, \mathbf{w}, b, \xi_i, \xi_i^*} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{n_T} (\xi_i + \xi_i^*) + \frac{\lambda_L}{2} \|\mathbf{f}_l^T - \mathbf{y}_l^T\|^2 \\ & + \frac{\lambda_D}{2} \sum_{s=1}^P \gamma_s \|\mathbf{f}_u^T - \mathbf{f}_u^s\|^2, \end{aligned} \quad (5.13)$$

$$\text{s.t.} \quad \mathbf{w}'\phi(\mathbf{x}_i^T) + b - f_i^T \leq \epsilon + \xi_i, \quad \xi_i \geq 0, \quad (5.14)$$

$$f_i^T - \mathbf{w}'\phi(\mathbf{x}_i^T) - b \leq \epsilon + \xi_i^*, \quad \xi_i^* \geq 0, \quad (5.15)$$

$$i = 1, \dots, n_T,$$

where ξ_i 's and ξ_i^* 's are slack variables for the ϵ -insensitive loss. In the experiment, we fix $\epsilon = 0.1$ which is also the default value in LIBSVM [153].

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 in (5.14) (*resp.*, (5.15)), we obtain the following Lagrangian:

$$\begin{aligned} L = \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{n_T} (\xi_i + \xi_i^*) + \frac{\lambda_L}{2} \|\mathbf{f}_l^T - \mathbf{y}_l^T\|^2 + \frac{\lambda_D}{2} \sum_{s=1}^P \gamma_s \|\mathbf{f}_u^T - \mathbf{f}_u^s\|^2 \\ & - \sum_{i=1}^{n_T} \alpha_i (\epsilon + \xi_i + f_i^T - \mathbf{w}'\phi(\mathbf{x}_i^T) - b) - \sum_{i=1}^{n_T} \eta_i \xi_i \\ & - \sum_{i=1}^{n_T} \alpha_i^* (\epsilon + \xi_i^* - f_i^T + \mathbf{w}'\phi(\mathbf{x}_i^T) + b) - \sum_{i=1}^{n_T} \eta_i^* \xi_i^*. \end{aligned} \quad (5.16)$$

Let us represent $\mathbf{f}^T = [\mathbf{f}_l^{T'}, \mathbf{f}_u^{T'}]'$. Setting the derivatives of (5.16) w.r.t. the primal variables ($\mathbf{f}^T, \mathbf{w}, b, \xi_i$ and ξ_i^*) to zeros, respectively, we have:

$$\mathbf{f}^T = \tilde{\mathbf{y}} + \text{diag} \left(\left[\frac{1}{\lambda_L} \mathbf{1}_{n_l}', \frac{1}{p\lambda_D} \mathbf{1}_{n_u}' \right] \right) (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*), \quad (5.17)$$

and $\mathbf{w} = \boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \boldsymbol{\alpha})$, $\mathbf{1}_{n_T}'\boldsymbol{\alpha} = \mathbf{1}_{n_T}'\boldsymbol{\alpha}^*$, $\mathbf{0}_{n_T} \leq \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \leq C\mathbf{1}_{n_T}$, where $\tilde{\mathbf{y}} = \left[\sum_{s=1}^P \gamma_s \mathbf{y}_l^T, \sum_{s=1}^P \tilde{\gamma}_s \mathbf{f}_u^s \right]'$, $\tilde{\gamma}_s = \frac{\gamma_s}{\sum_{s=1}^P \gamma_s}$ is the normalized weight for the s -th base classifier, $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_{n_T}]'$ and $\boldsymbol{\alpha}^* = [\alpha_1^*, \dots, \alpha_{n_T}^*]'$ are the vectors of the dual variables, and $\boldsymbol{\Phi} = [\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_{n_T})]$.

Substituting them back into (5.16), we arrive at the following dual formulation:

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \quad & \frac{1}{2} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)' \tilde{\mathbf{K}} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \tilde{\mathbf{y}}' (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \epsilon \mathbf{1}_{n_T}' (\boldsymbol{\alpha} + \boldsymbol{\alpha}^*), \\ \text{s.t.} \quad & \mathbf{1}_{n_T}' \boldsymbol{\alpha} = \mathbf{1}_{n_T}' \boldsymbol{\alpha}^*, \quad \mathbf{0}_{n_T} \leq \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \leq C\mathbf{1}_{n_T}, \end{aligned} \quad (5.18)$$

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

Parametric Prediction: From the Karush-Kuhn-Tucker (KKT) condition in (5.17), we can obtain the vector of the target decision values $\mathbf{f}^T$. Moreover, the decision value of any unlabeled data $\mathcal{D}_u^T$ in the target domain is given as: $f^T(\mathbf{x}_i) = \sum_{s=1}^P \tilde{\gamma}_s f^s(\mathbf{x}_i) + \frac{\alpha_i - \alpha_i^*}{p\lambda_D}$, $\forall i = n_l + 1, \dots, n_T$, 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 $\Delta f(x_i) = \frac{\alpha_i - \alpha_i^*}{p\lambda_D}$. 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

$$f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b = \sum_{i: \alpha_i - \alpha_i^* \neq 0} (\alpha_i - \alpha_i^*) k(\mathbf{x}_i, \mathbf{x}) + b, \quad (5.19)$$

which is a linear combination of $k(\mathbf{x}_i, \mathbf{x})$'s only without involving any base classifiers. Here, $\mathbf{x}_i$ is the support vector from the samples in the target domain 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 samples have the value $|\mathbf{w}'\phi(\mathbf{x}_i) + b - f_i^T|$ less than ϵ , then their corresponding coefficient in (5.19) becomes zero. Therefore, with the use of the ϵ -insensitive loss, the computation for the prediction using the sparse representation in (5.19) can be greatly reduced when compared with that of A-SVM. We therefore refer to this sparse-solution version of DAM as FastDAM.

5.3.4 Domain Adaptation Machine with Universum

Since the classification hyperplane learned with only a limited number of the labeled training samples in the target domain may overfit the training data, the generalization ability of the learned FastDAM classifier may be degraded. As shown in the traditional transductive and semi-supervised learning methods [7, 193, 203, 206], 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 [26, 207] proposed to use an additional unlabeled data set called *Universum* to enhance the generalization ability of the learned classifier for the binary classification tasks (see Appendix A.4 for a introduction of Universum). In contrast to

the traditional transductive and semi-supervised learning 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.* [201] introduced a new regularizer into the SVM framework based on the Universum. The proposed method, referred to as $\mathfrak{U}$ -SVM, not only maximizes the margin between two classes, but also maximizes the number of contradictions on the Universum. Sinz *et al.* [200] discovered the connection between the Universum related algorithm in [201] and other algorithms including SVM in a projected subspace, Fisher discriminant analysis and oriented PCA.

Recall that the learned classifier by using the labeled samples from the source and target domains may not perform well because of the distribution mismatch [12, 17, 30, 36, 54]. As mentioned in [26, 201], Universum can be drawn from a distribution that is different from yet close to that of the labeled training samples. In this work, we use the samples in the source domains as the Universum because of the following two aspects:

- 1) the distributions of the samples from the source domain and the target domain are different but overlap to some extent.
- 2) The Universum regularizer $\Omega_{\mathfrak{U}}(f^T)$ in (5.6) is a data-dependent regularizer and it can be used to control the complexity of the learned classifier.

To this end, we model $\Omega_D(f^T)$ using two regularizers $\Omega_{\mathcal{A}}(f^T)$ and $\Omega_{\mathfrak{U}}(f^T)$, namely:

$$\Omega_D(f^T) = \frac{1}{2} \sum_{s=1}^P \gamma_s \sum_{i=n_l+1}^{n_T} (f^T(\mathbf{x}_i^T) - f^s(\mathbf{x}_i^T))^2 + \frac{\theta}{2} \sum_{s=1}^P \sum_{i=1}^{n_s} (f^T(\mathbf{x}_i^s))^2. \quad (5.20)$$

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

Denote $\lambda_{D_1} = \lambda_D$ and $\lambda_{D_2} = \theta \lambda_D$. Substituting (5.20) back into the DAM framework (5.8), we arrive at the following optimization problem:

$$\begin{aligned} \min_{f^T} \quad & \Omega(f^T) + \frac{\lambda_L}{2} \sum_{i=1}^{n_l} (f^T(\mathbf{x}_i^T) - y_i^T)^2 + \frac{\lambda_{D_1}}{2} \sum_{s=1}^P \gamma_s \sum_{i=n_l+1}^{n_T} (f^T(\mathbf{x}_i^T) - f^s(\mathbf{x}_i^T))^2 \\ & + \frac{\lambda_{D_2}}{2} \sum_{s=1}^P \sum_{i=1}^{n_s} (f^T(\mathbf{x}_i^s))^2. \end{aligned} \quad (5.21)$$

5.3.4.1 Detailed Solution

Note that the optimization problem (5.21) can be solved through the least-squares method. However, it is computationally infeasible as stated in Section 5.3.3.1. Again, we employ the ϵ -insensitive loss to regulate the approximation quality and the sparsity of the target decision function. Let $\mathbf{f}_T^T = [f^T(\mathbf{x}_1^T), \dots, f^T(\mathbf{x}_{n_T}^T)]' = [\mathbf{f}_l^{T'}, \mathbf{f}_u^{T'}]'$, $\boldsymbol{\xi}_T = [\xi_{T,1}, \dots, \xi_{T,n_T}]'$ and $\boldsymbol{\xi}_T^* = [\xi_{T,1}^*, \dots, \xi_{T,n_T}^*]'$ for the target domain; $\mathbf{f}_s^T = [f^T(\mathbf{x}_1^s), \dots, f^T(\mathbf{x}_{n_s}^s)]'$, $\boldsymbol{\xi}_s = [\xi_{s,1}, \dots, \xi_{s,n_s}]'$ and $\boldsymbol{\xi}_s^* = [\xi_{s,1}^*, \dots, \xi_{s,n_s}^*]'$ for the s -th source domain. With the ϵ -insensitive loss, we can rewrite (5.21) as a constrained optimization problem which is referred to as UniverDAM:

$$\begin{aligned} \min_{\substack{\mathbf{f}_T^T, \mathbf{w}, b, \\ \xi_{T,i}, \xi_{T,i}^*, \xi_{s,i}, \xi_{s,i}^*}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \left(\mathbf{1}_{n_T}' (\boldsymbol{\xi}_T + \boldsymbol{\xi}_T^*) + \sum_{s=1}^P \mathbf{1}_{n_s}' (\boldsymbol{\xi}_s + \boldsymbol{\xi}_s^*) \right) + \frac{\lambda_L}{2} \|\mathbf{f}_l^T - \mathbf{y}_l^T\|^2 \\ & + \frac{\lambda_{D_1}}{2} \sum_{s=1}^P \gamma_s \|\mathbf{f}_u^T - \mathbf{f}_u^s\|^2 + \frac{\lambda_{D_2}}{2} \sum_{s=1}^P \|\mathbf{f}_s^T\|^2, \quad (5.22) \\ \text{s.t.} \quad & \text{For the target domain:} \end{aligned}$$

$$\mathbf{w}'\phi(\mathbf{x}_i^T) + b - f^T(\mathbf{x}_i^T) \leq \epsilon_T + \xi_{T,i}, \quad \xi_{T,i} \geq 0, \quad i = 1, \dots, n_T, \quad (5.23)$$

$$f^T(\mathbf{x}_i^T) - \mathbf{w}'\phi(\mathbf{x}_i^T) - b \leq \epsilon_T + \xi_{T,i}^*, \quad \xi_{T,i}^* \geq 0, \quad i = 1, \dots, n_T; \quad (5.24)$$

For the s -th source domain, $s = 1, \dots, P$:

$$\mathbf{w}'\phi(\mathbf{x}_i^s) + b - f^T(\mathbf{x}_i^s) \leq \epsilon_s + \xi_{s,i}, \quad \xi_{s,i} \geq 0, \quad i = 1, \dots, n_s, \quad (5.25)$$

$$f^T(\mathbf{x}_i^s) - \mathbf{w}'\phi(\mathbf{x}_i^s) - b \leq \epsilon_s + \xi_{s,i}^*, \quad \xi_{s,i}^* \geq 0, \quad i = 1, \dots, n_s. \quad (5.26)$$

We introduce the Lagrangian multipliers $\boldsymbol{\alpha}_T = [\alpha_{T,1}, \dots, \alpha_{T,n_T}]'$ and $\boldsymbol{\eta}_T = [\eta_{T,1}, \dots, \eta_{T,n_T}]'$ (*resp.*, $\boldsymbol{\alpha}_T^* = [\alpha_{T,1}^*, \dots, \alpha_{T,n_T}^*]'$ and $\boldsymbol{\eta}_T^* = [\eta_{T,1}^*, \dots, \eta_{T,n_T}^*]'$) for the constraints of the target domain in (5.23) (*resp.*, (5.24)); as well as the Lagrangian multipliers $\boldsymbol{\alpha}_s = [\alpha_{s,1}, \dots, \alpha_{s,n_s}]'$ and $\boldsymbol{\eta}_s = [\eta_{s,1}, \dots, \eta_{s,n_s}]'$ (*resp.*, $\boldsymbol{\alpha}_s^* = [\alpha_{s,1}^*, \dots, \alpha_{s,n_s}^*]'$ and $\boldsymbol{\eta}_s^* = [\eta_{s,1}^*, \dots, \eta_{s,n_s}^*]'$) for the constraints of the s -th source domain in (5.25) (*resp.*, (5.26)). We then derive the

Lagrangian of (5.22) as follows:

$$\begin{aligned}
 L = & \frac{1}{2} \|\mathbf{w}\|^2 + C \left(\mathbf{1}'_{n_T} (\boldsymbol{\xi}_T + \boldsymbol{\xi}_T^*) + \sum_{s=1}^P \mathbf{1}'_{n_s} (\boldsymbol{\xi}_s + \boldsymbol{\xi}_s^*) \right) \\
 & + \frac{\lambda_L}{2} \|\mathbf{f}_l^T - \mathbf{y}_l^T\|^2 + \frac{\lambda_{D_1}}{2} \sum_{s=1}^P \gamma_s \|\mathbf{f}_u^T - \mathbf{f}_u^s\|^2 + \frac{\lambda_{D_2}}{2} \sum_{s=1}^P \|\mathbf{f}_s^T\|^2 \\
 & - \boldsymbol{\alpha}'_T (\epsilon_T \mathbf{1}_{n_T} + \boldsymbol{\xi}_T + \mathbf{f}_T^T - \boldsymbol{\Phi}'_T \mathbf{w} - b \mathbf{1}_{n_T}) - \boldsymbol{\eta}'_T \boldsymbol{\xi}_T \\
 & - \boldsymbol{\alpha}^{*'}_T (\epsilon_T \mathbf{1}_{n_T} + \boldsymbol{\xi}_T^* - \mathbf{f}_T^T + \boldsymbol{\Phi}'_T \mathbf{w} + b \mathbf{1}_{n_T}) - \boldsymbol{\eta}^{*'}_T \boldsymbol{\xi}_T^* \\
 & - \sum_{s=1}^P \left(\boldsymbol{\alpha}'_s (\epsilon_s \mathbf{1}_{n_s} + \boldsymbol{\xi}_s + \mathbf{f}_s^T - \boldsymbol{\Phi}'_s \mathbf{w} - b \mathbf{1}_{n_s}) + \boldsymbol{\eta}'_s \boldsymbol{\xi}_s \right. \\
 & \left. + \boldsymbol{\alpha}^{*'}_s (\epsilon_s \mathbf{1}_{n_s} + \boldsymbol{\xi}_s^* - \mathbf{f}_s^T + \boldsymbol{\Phi}'_s \mathbf{w} + b \mathbf{1}_{n_s}) + \boldsymbol{\eta}^{*'}_s \boldsymbol{\xi}_s^* \right), \tag{5.27}
 \end{aligned}$$

where $\boldsymbol{\Phi}_T = [\phi(\mathbf{x}_1^T), \dots, \phi(\mathbf{x}_{n_T}^T)]$ and $\boldsymbol{\Phi}_s = [\phi(\mathbf{x}_1^s), \dots, \phi(\mathbf{x}_{n_s}^s)]$.

We further denote $\mathbf{f}^T = [\mathbf{f}_T^T, \mathbf{f}_1^T, \dots, \mathbf{f}_P^T]'$, $\boldsymbol{\xi} = [\boldsymbol{\xi}_T', \boldsymbol{\xi}_1', \dots, \boldsymbol{\xi}_P']'$, $\boldsymbol{\xi}^* = [\boldsymbol{\xi}_T^{*'}, \boldsymbol{\xi}_1^{*'}, \dots, \boldsymbol{\xi}_P^{*'}]'$, $\boldsymbol{\alpha} = [\boldsymbol{\alpha}'_T, \boldsymbol{\alpha}'_1, \dots, \boldsymbol{\alpha}'_P]'$, $\boldsymbol{\alpha}^* = [\boldsymbol{\alpha}^{*'}_T, \boldsymbol{\alpha}^{*'}_1, \dots, \boldsymbol{\alpha}^{*'}_P]'$ and $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_T, \boldsymbol{\Phi}_1, \dots, \boldsymbol{\Phi}_P]$. By setting the derivatives of the Lagrangian (5.27) w.r.t. the primal variables $\mathbf{f}^T, \mathbf{w}, b, \boldsymbol{\xi}$ and $\boldsymbol{\xi}^*$ to zero, we can derive the following solutions:

$$\mathbf{f}^T = \tilde{\mathbf{y}} + \text{diag} \left(\left[\frac{1}{\lambda_L} \mathbf{1}'_{n_l}, \frac{1}{p\lambda_{D_1}} \mathbf{1}'_{n_u}, \frac{1}{\lambda_{D_2}} \mathbf{1}'_{\sum_{s=1}^P n_s} \right] \right) (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*), \tag{5.28}$$

$\mathbf{w} = \boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \boldsymbol{\alpha})$, $\mathbf{1}'_N \boldsymbol{\alpha} = \mathbf{1}'_N \boldsymbol{\alpha}^*$ and $\mathbf{0}_N \leq \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \leq C \mathbf{1}_N$, where $\tilde{\mathbf{y}} = [\mathbf{y}_l^T, \sum_{s=1}^P \tilde{\gamma}_s \mathbf{f}_u^s, \mathbf{0}'_{\sum_{s=1}^P n_s}]'$, $N = n_T + \sum_{s=1}^P n_s$ is the total number of training samples and $\tilde{\gamma}_s = \frac{\gamma_s}{\sum_{s=1}^P \gamma_s}$ is the normalized weight for the s -th base classifier. Similarly to the derivation of FastDAM as in Section 5.3.3.2, we arrive at the final dual formulation of UniverDAM as follows:

$$\begin{aligned}
 \min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \quad & \frac{1}{2} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)' \tilde{\mathbf{K}} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \tilde{\mathbf{y}}' (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \boldsymbol{\epsilon}' (\boldsymbol{\alpha} + \boldsymbol{\alpha}^*), \\
 \text{s.t.} \quad & \mathbf{1}'_N \boldsymbol{\alpha} = \mathbf{1}'_N \boldsymbol{\alpha}^*, \quad \mathbf{0}_N \leq \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \leq C \mathbf{1}_N,
 \end{aligned} \tag{5.29}$$

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

5.3.4.2 Parametric Prediction

The solution (5.28) 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 samples. To this end, similarly as in Section 5.3.3.2, we introduce a parametric form of the target decision function f^T for label prediction. With the dual variables α and α^* obtained from (5.29), the target decision function f^T on any test instance is formulated as follows:

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

5.4 Discussions

We first discuss the connections between our proposed methods FastDAM and UniverDAM, and we also introduce a theorem on the degradation of UniverDAM into FastDAM. Moreover, we show the connection between our two methods and support vector regression. Finally, we discuss the differences between our DAM framework and other related work.

5.4.1 Connection between FastDAM and UniverDAM

One would observe that when $\alpha_{s,i} = \alpha_{s,i}^* = 0$, the dual optimization problem (5.29) of UniverDAM reduces to the dual optimization problem (5.18) of FastDAM. The following theorem shows that UniverDAM can be reduced to FastDAM under some conditions.

Theorem 5.4 *The optimization problem (5.22) of UniverDAM will be reduced to the optimization problem (5.13) of FastDAM, if each ϵ_s ($s = 1, \dots, P$) satisfies the following condition:*

$$\epsilon_s > \max_{i=1, \dots, n_s} |\mathbf{w}'\phi(\mathbf{x}_i^s) + b - f^T(\mathbf{x}_i^s)|. \quad (5.30)$$

Proof: The proof is given in Appendix A.3.

Recall that Universum is initially proposed for the binary-class problem [26, 200, 201], which means that the positive samples are from one class and the negative samples are

from another class. The Universum is an additional data collection belonging to neither of the two classes, 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 multi-class setting in which the training data are from multiple classes.

5.4.2 Connection to Support Vector Regression

Under the framework of DAM, we have introduced two methods FastDAM and UniverDAM. Surprisingly, for both FastDAM and UniverDAM, the dual forms of (5.18) and (5.29) do not involve any expensive matrix operation as in [197–199] and can be reduced to a form which is quite close to the dual of ϵ -SVR:

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \quad & \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)' \mathbf{K}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \mathbf{y}'(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \epsilon \mathbf{1}'(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*), \\ \text{s.t.} \quad & \boldsymbol{\alpha}' \mathbf{1} = \boldsymbol{\alpha}^{*'} \mathbf{1}, \quad \mathbf{0} \leq \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \leq C \mathbf{1}, \end{aligned} \quad (5.31)$$

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 (5.18) of FastDAM as an example, and note that the dual (5.29) 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 (5.31) are replaced by the transformed kernel matrix $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{y}}$ in (5.18), respectively. In experiments, we normalize the sum of γ_s to 1 (*i.e.*, $p = 1$). So the transformed kernel matrix is $\mathbf{K} + \text{diag}\left(\left[\frac{1}{\lambda_L} \mathbf{1}'_{n_l}, \frac{1}{\lambda_D} \mathbf{1}'_{n_u}\right]\right)$, which is similar to Automatic Relevance Determination (ARD) kernel used in Gaussian Process, where λ_L and λ_D are the parameters to control the noise of output. Moreover, the i -th 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 $\mathcal{D}_u^T$ (also see Figure 5.1). Moreover, the objective function of FastDAM in (5.12) can be solved efficiently by using state-of-the-art SVM solvers such as LIBSVM [153]. When compared with the original formulation in (5.9), the calculation of the matrix inversion in (5.11) is avoided.

5.4.3 Discussions with Related Work

Our proposed DAM framework is different from MTL. DAM focuses on learning the target decision classifier only by leveraging the existing base classifiers, and the computational cost in learning stage is reduced, especially for FastDAM. In addition, according to the definition of our domain-dependent regularizer in (5.7), the base classifiers can be trained with different kernels and even different learning methods.

The most related work to DAM is A-SVM [30], in which the new SVM classifier is adapted from the existing source classifiers. However, DAM is different from A-SVM in the following two aspects:

- 1) A-SVM did not exploit the unlabeled data $\mathcal{D}_u^T$ in the target domain. In contrast, the unlabeled samples $\mathcal{D}_u^T$ in the target domain are employed in DAM (see the domain-dependent regularizer defined in (5.7)).
- 2) A-SVM employed source classifiers for the label prediction of the samples in the target domain. In contrast, the target classifier learned in DAM (see (5.19)) is in a sparse representation of the target samples only.

In Table 5.1, we summarize the comparisons between our two methods (*i.e.*, FastDAM and UniverDAM) and other domain adaptation methods (*i.e.*, feature replication (FR) [29], adaptive SVM (A-SVM) [30], multiple convex combination of SVM (MCC-SVM) [31] and multiple KMM (Multi-KMM) [31]) 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 assumption.

5.5 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.

Table 5.1: Summary of the comparisons between our two methods (*i.e.*, FastDAM and UniverDAM) and other domain adaptation methods.

	Smoothness assumption	Pre-learned classifiers	Source data	Labeled target data	Unlabeled target data	Fast adaptation	Fast prediction
FR [29]	X	X	✓	✓	X	X	X
A-SVM [30]	X	✓	X	✓	X	✓	X
MCC-SVM [31]	X	✓	X	✓	X	✓	X
Multi-KMM [31]	X	X	✓	Optional	✓	X	X
FastDAM	✓	✓	X	Optional	✓	✓	✓
UniverDAM	✓	✓	✓	Optional	✓	X	X

5.5.1 Descriptions of Data Sets

We conduct experiments on three data sets which are suitable for the multiple source domain adaptation applications. We use the challenging TRECVID 2005 data set for the video concept detection task and employ two text data sets (*i.e.*, 20 newsgroups and email spam) for document retrieval.

5.5.1.1 TRECVID 2005 Data Set

The TRECVID³ video corpus is one of the largest annotated video benchmark data sets for research purposes. The TRECVID 2005 data set 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 each channel is listed in Table 5.2. 36 semantic concepts are chosen from the LSCOM-lite lexicon [156], 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 TRECVID 2005 data set

As shown in [30], 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 $\mathcal{D}^T$. The training data set comprises of all the labeled samples from the source domains as well as the labeled samples (*i.e.*, $\mathcal{D}_l^T$) 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. Moreover, from the test data set we randomly select 4,000 samples as the unlabeled training data. We only sample the samples from the target domain once due to the very high computational cost in the large TRECVID 2005 data set.

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 keyframes, because of their consistent and good performance reported in

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

Table 5.2: Description of the TRECVID 2005 data set.

Domain	Source domains					Target domain
Channel	CNN_ENG	MSNBC_ENG	NBC_ENG	CCTV4_CHN	NTDTV_CHN	LBC_ARB
#keyframes	11,025	8,905	9,322	10,896	6,481	15,272

Table 5.3: Description of the 20 Newsgroups data set.

	Source domains	Target domain
rec vs. sci	rec.autos & sci.crypt rec.motorcycles & sci.electronics rec.sport.baseball & sci.med	rec.sport.hockey & sci.space
comp vs. 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 vs. 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 5.4: Description of the email spam data set.

Domain	Source domains			Target domain
Email sets	User1 (U00)	User2 (U01)	User3 (U02)	Public set
#emails	2,500	2,500	2,500	4,000

TRECVID [14, 30]. Yanagawa *et al.* [157] have made the three types of features extracted from the TRECVID 2005 data set publicly available. The three types of features are then put together to form a 346-dimensional feature to represent each keyframe.

5.5.1.2 20 Newsgroups Data Set

The 20 Newsgroups data set⁴ contains 18,774 documents, which has a hierarchical structure with six main categories and 20 subcategories. We choose the samples 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 samples from two subcategories (*i.e.*, one is from the positive class and the other one is from the negative class) to construct one domain. In the experiments, we have three source domains and one target domain (see Table 5.3 for the detailed settings). The training data set comprises of all the labeled samples from the source domains as well as the labeled samples from the target domain, in which m positive and m negative samples 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 for ten times with different randomly sampled samples from the target domain and report the means and the standard deviations. The word-frequency feature is used to represent each document.

5.5.1.3 Email Spam Data Set

Email spam data set⁵ contains a set of 4,000 publicly available labeled emails as well as three email sets (each has 2,500 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 data sets, 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 domain (see Table 5.4 for more

⁴<http://people.csail.mit.edu/jrennie/20Newsgroups>

⁵<http://www.ecmlpkdd2006.org/challenge.html>

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

5.5.2 Experimental Setup

For performance evaluation, we use non-interpolated Average Precision (AP) [158], 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.

Any base classifiers can be readily used in our DAM framework. In the experiments, we test our methods using two types of base classifiers because of their general good performances (see Tables 5.5, 5.6 and 5.7): 1) the standard SVM classifier learned by using the labeled samples from one source domain, which is the same as the source classifier in A-SVM [30]; 2) the feature replication (FR) classifier trained with the labeled samples from one source domain and the target domain. On the TRECVID 2005 and email spam data sets, we use FR classifiers as base classifiers in FastDAM and UniverDAM. And on the 20 Newsgroups data set, 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 samples from the target domain (see Section 5.4.2 for more details). However, the unlabeled samples from the target domain may bring ambiguity in the learning of the target classifier if their virtual labels are close to zeros. To alleviate such side effect of the ambiguity as well as accelerate the learning process, we discard the unlabeled samples from the target domain with the virtual labels ranging from -0.3 to 0.3 and only employ the remaining unlabeled samples. In the experiment, we empirically fix the thresholds as -0.3 and 0.3 . We will investigate how to automatically determine the threshold in the future.

5.5.2.1 Detailed Setup for Video Concept Detection

We compare our method FastDAM with the baseline SVM and other four domain adaptation methods: multiple convex combination of SVM (MCC-SVM) [31], feature replication (FR) [29], adaptive SVM (A-SVM) [30] and multiple KMM (Multi-KMM) [31]. We do not test UniverDAM for the video concept detection task because the samples from each source domain come from multiple classes which violates the basic assumption on UniverDAM stated in Section 5.4.1. It is also worth mentioning that UniverDAM can achieve comparable results to those of FastDAM, because FastDAM is a special case of UniverDAM (see Section 5.4.1 for the details about the connection between FastDAM and UniverDAM).

In this work, 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 samples from the target domain (*i.e.*, $\mathcal{D}_t^T$) for SVM learning; 2) in SVM.S, we *equally fuse*⁶ the decision values of five base classifiers independently trained with the labeled samples 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, similarly as in [31], we equally fuse the decision values of six SVM classifiers independently trained with the labeled samples 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 samples 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 per class) for Multi-KMM [31], we shift the samples from each source domain towards 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 semi-supervised setting in this work. In practice, 4,000 test samples from the target domain are randomly sampled as

⁶For 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.*, $g(t) = \frac{1}{1+\exp(-t)}$) as suggested in [208]. Finally, we average the P probability values as the final prediction of the test instance $\mathbf{x}$.

$\mathcal{D}_u^T$ for Multi-KMM and FastDAM, which are used as unlabeled data during the learning process.

For all methods, we train one-versus-others SVM classifiers with the fixed tradeoff 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 $\frac{1}{d} = 0.0029$ ($d = 346$ is the feature dimension). For A-SVM, we train 50 source classifiers by independently using five sources and ten kernel parameters for Gaussian kernel, which are set as $1.2^\delta\gamma$, where $\delta \in \{-0.5, 0, 0.5, \dots, 4\}$. 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; 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.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2 + 1}$) and inverse distance kernel (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\sqrt{\gamma}\|\mathbf{x}_i - \mathbf{x}_j\| + 1}$). Then, for FastDAM, there are in total 200 base classifiers from five sources, four types of kernels and ten 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

$$\gamma_s = e^{-\beta \text{DIST}_k^2(\mathcal{D}^s, \mathcal{D}^T)}, \quad (5.32)$$

where $\beta \geq 0$ is the bandwidth parameter to control the spread of $\text{DIST}_k(\mathcal{D}^s, \mathcal{D}^T)$ and $\text{DIST}_k(\mathcal{D}^s, \mathcal{D}^T)$ is the *Maximum Mean Discrepancy* (MMD) [37] for measuring the data distributions between the s -th the source domain and the target domain. MMD is a nonparametric distance metric for comparing data distributions in the RKHS, namely, $\text{DIST}_k(\mathcal{D}^s, \mathcal{D}^T) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(\mathbf{x}_i^s) - \frac{1}{n_T} \sum_{i=1}^{n_T} \phi(\mathbf{x}_i^T) \right\|_{\mathcal{H}}^2$. MMD is an effective nonparametric criterion to compare two data distributions based on the distance between the means of the samples from two domains in the Reproducing Kernel Hilbert Space (RKHS). It also captures the higher order statistics of the data (e.g., the higher order moments of probability distribution) by transforming the samples into a higher dimensional or even infinite dimensional space. As shown in [37], the data distributions of the two domains become matched, if MMD is close to zero. In this work, we adopt MMD owing to its effectiveness and simplicity [17, 36, 37]. In the experiment, we further normalize the sum of the weights γ_s 's as 1 and empirically set $\beta = 100$.

5.5.2.2 Detailed Setup for Document Retrieval

The 20 Newsgroups data set and the email spam data set 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. And the settings for these methods are the same as in 5.5.2.1.

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 $a = 1.1, 1.2, \dots, 1.5$ for both data sets. And the linear kernel is used as the default kernel because of its good performance on text data sets. Therefore, we have in total 18 base classifiers from 3 sources and 6 base kernels. In the default setting, the tradeoff 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 data sets. As the same as in the video concept detection task, we use (5.32) to determine the weight γ_s for the s -th base classifier, and we further normalize the sum of the weights γ_s 's as 1. We empirically set β as 10000 for the three settings of the 20 Newsgroups data set and β as 100 for the email spam data set. In Section 5.5.4, we take the 20 Newsgroups data set 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 .

5.5.3 Performance Comparisons

We compare our proposed methods with other baseline methods on the TRECVID 2005, 20 Newsgroups and email spam data sets. We also analyze the parameters used in different methods by using the 20 Newsgroups data set.

5.5.3.1 Results of Video Concept Detection

For each concept in the TRECVID 2005 data set, we count the frequency (referred to as positive frequency) of positive samples in the target domain (*i.e.*, LBC_ARB). According to the positive frequency, we partition all the 36 concepts into three groups (*i.e.*, Group_High, Group_Med and Group_Low), with 12 concepts for each group in Figure 5.2. The concepts in Group_High, Group_Med and Group_Low are with high, moderate and low positive frequencies, respectively. The Mean Average Precisions (MAPs) over all

Figure 5.2: Per-concept APs of all the 36 concepts using different methods. The concepts are divided into three groups according to the positive frequency.

Table 5.5: MAPs (%) of all the methods over 36 concepts on the TRECVID 2005 data set.

	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

36 concepts on the TRECVID 2005 data set is given in Table 5.5. From Table 5.5, 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 samples 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 conference version of this work [54], we have shown that our FastDAM.50 and FastDAM.200 using the source 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 5.5, 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%, which are better than SVM.T, SVM.S and other domain adaptation 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 pre-learned base classifiers.

We additionally report the results of SVM.S, MCC-SVM and FR in the single-source domain setting in which all the samples 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 5.5.

5.5.3.2 Results of Document Retrieval

Table 5.6 shows the means and standard deviations of APs of all the methods on the 20 Newsgroups data set. When the number of positive and negative training samples

Table 5.6: Means and Standard Deviations (%) of APs of all the methods with m positive and m negative training samples from the target domain on the 20 Newsgroups data set. 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 vs. sci							
m	SVM_T	SVM_S	MCC-SVM	FR	Multi-KMM	FastDAM	UniverDAM
0	–	95.44 $\pm$ 0.00	–	–	97.58 $\pm$ 0.00	97.77 $\pm$ 0.00	99.69 $\pm$ 0.00
2	67.58 $\pm$ 6.82	95.44 $\pm$ 0.01	95.45 $\pm$ 0.26	93.87 $\pm$ 2.86	97.69 $\pm$ 0.07	97.81 $\pm$ 0.05	99.69 $\pm$ 0.01
4	73.56 $\pm$ 7.46	95.44 $\pm$ 0.02	95.61 $\pm$ 0.41	94.87 $\pm$ 2.63	97.81 $\pm$ 0.09	97.87 $\pm$ 0.05	99.69 $\pm$ 0.01
6	75.77 $\pm$ 5.79	95.45 $\pm$ 0.03	95.66 $\pm$ 0.40	95.34 $\pm$ 1.58	97.91 $\pm$ 0.07	97.93 $\pm$ 0.07	99.70 $\pm$ 0.01
10	81.55 $\pm$ 6.87	95.45 $\pm$ 0.03	95.82 $\pm$ 0.56	95.94 $\pm$ 1.43	98.04 $\pm$ 0.08	98.01 $\pm$ 0.11	99.71 $\pm$ 0.02
15	87.58 $\pm$ 4.72	95.46 $\pm$ 0.05	96.03 $\pm$ 0.43	96.69 $\pm$ 0.95	98.18 $\pm$ 0.09	98.04 $\pm$ 0.12	99.72 $\pm$ 0.01
20	90.52 $\pm$ 3.70	95.46 $\pm$ 0.05	96.17 $\pm$ 0.51	97.31 $\pm$ 0.87	98.31 $\pm$ 0.09	98.11 $\pm$ 0.13	99.72 $\pm$ 0.01

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

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

(*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 samples 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 towards the target domain on this data set. 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 5.6) 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 5.6). The explanation is that the existing domain adaptation algorithms like Multi-KMM and FR can achieve good performances when there are a sufficient number of 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 [200, 201] is suitable for this binary-class document retrieval problem in which the samples 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 that there are no labeled samples 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 like MCC-SVM and FR cannot cope with such an extreme case.

Table 5.7 lists the results of all the methods on the email spam data set. 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 data set. In terms of the means of APs, our two methods outperforms the other methods and UniverDAM achieves the best result, which demonstrate the effectiveness of our methods again. Moreover, UniverDAM

Table 5.7: Means and Standard Deviations (%) of APs of all the methods with 10 positive and 10 negative training samples from the target domain on the email spam data set. 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 5.8: Average training and testing time (second) over ten rounds of experiments for all methods on the first setting (*i.e.*, rec vs. sci) of the 20 Newsgroups data set. 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

Figure 5.3: Means and Standard Deviations (%) of APs of all the methods on the 20 Newsgroups data set with different tradeoff parameter $C = 0.01, 0.1, 1, 10$ and 100 .

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 vs. sci) of the 20 Newsgroups data set as an example to compare the average training and test time of all the methods. All methods are performed on an IBM workstation (2.13GHz CPU with 16Gbyte RAM), and we set $m = 20$ for all methods. The performances of all the methods are shown in Table 5.8. Note that for our methods FastDAM and UniverDAM, we need to obtain the virtual labels of the unlabeled training samples 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 seconds 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 samples. While a large number of unlabeled training data from the source domain are used as UniverDAM, the testing time of UniverDAM is comparable with other methods (*i.e.*, SVM_S, MCC-SVM, FR and Multi-KMM).

5.5.4 Parameter Analysis for Different Methods

In this subsection, we evaluate the performance variations with respect to the tradeoff 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 data set in which 20 positive and 20 negative labeled samples from the target domain are used for training. In the default setting, we set the tradeoff parameter $C = 1$ for all methods,

Figure 5.4: Means and Standard Deviations (%) of APs of FastDAM and UniverDAM on the 20 Newsgroups data set with different tradeoff parameters $\lambda_D = \lambda_{D_1} = \lambda_{D_2} = 0.01, 0.1, 1, 10, 100$ and 1000.

$\lambda_L = \lambda_D = \lambda_{D_1} = \lambda_{D_2} = 1$ and $\beta = 10000$ for both FastDAM and UniverDAM. When evaluating the performance variations with respect to one parameter, we fix the other parameters as their default values.

5.5.4.1 Performance Variations w.r.t. the Regularization Parameter C

We compare all methods on the three settings of the 20 Newsgroups data set by using different C in Figure 5.3, 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 . 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.

5.5.4.2 Performance Variations w.r.t. the Tradeoff Parameters $\lambda_L, \lambda_D, \lambda_{D_1}$ and λ_{D_2}

We conduct two sets of 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 are set as default values. The performance variations with respect to different λ_D are shown in Figure 5.4. When setting $\lambda_D \leq 1$, it can be observed

Figure 5.5: Means and Standard Deviations (%) of APs of FastDAM and UniverDAM on the 20 Newsgroups data set with different tradeoff parameter $\lambda = 0.1, 1, 10, 100$ and 1000 .

from Figure 5.4 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 and make our discussion clearer, we fix $\lambda_L = \lambda_D = \lambda_{D_1} = \lambda_{D_2} = \lambda$. We report the results of FastDAM and UniverDAM by using different tradeoff parameter λ in Figure 5.5, where λ is set as $0.1, 1, 10, 100$ and 1000 . From Figure 5.5, we observe that the performances of both FastDAM and UniverDAM do not change much when $\lambda \geq 1$, and UniverDAM achieves the best performance when setting $\lambda = 1$.

5.5.4.3 Performance Variations w.r.t. the Bandwidth Parameter β

Recall that β is the bandwidth parameter for defining γ_s (see 5.32), and we normalize the sum of γ_s 's to 1. In Figure 5.6 we show the performances of FastDAM and UniverDAM by using different β , where β is set as $0, 0.01, 1, 100$ and 10000 . When setting $\beta = 0$, we have equal weights for all source domains (*i.e.*, $\gamma_s = \frac{1}{P}$, $\forall s = 1, \dots, P$). From Figure 5.6, we observe that both FastDAM and UniverDAM using $\beta = 10000$ achieve better performances when compared with $\beta = 0$, which demonstrates 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

Figure 5.6: Means and Standard Deviations (%) of APs of FastDAM and UniverDAM on the 20 Newsgroups data set with different bandwidth parameter $\beta = 0, 0.01, 1, 100$ and 10000 in (5.32).

than FastDAM, which again demonstrates the effectiveness of the UniverDAM constructed by using the source domain data for document retrieval.

5.6 Summary

We have proposed a new framework, referred to as Domain Adaptation Machine (DAM), for multiple source domain adaptation. It learns a robust target classifier for predicting labels of the test samples from the target domain by leveraging a set of pre-learned base classifiers. Any classifier such as the standard SVM classifier learned with the labeled samples from the source domains or the FR classifier [29] learned with the labeled samples from the source domains and the target domain can be readily used as the base classifier in our framework. With the base classifiers, we introduce 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 samples from the target domain. This newly proposed regularizer can be readily combined with many kernel methods such as SVM, SVR, least-squares SVM, and so on for domain adaptation. Under this framework, we also developed two methods, referred to as FastDAM and UniverDAM. In FastDAM, we incorporate the proposed domain-dependent regularizer into least-squares SVM (LS-SVM) as well as employ a sparsity regularizer based on the ϵ -insensitive loss to enforce the sparsity on the target classifier. In FastDAM, the label prediction of test samples is very fast, making it suitable for large scale applications

(*e.g.*, video concept detection) with a large amount of test samples. In order to further enhance the generalization ability of the target classifier, in UniverDAM we additionally introduce another regularizer suggested in [200, 201] into the objective function of FastDAM by treating the samples from the source domains as the Universum. We also show 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 LIB-SVM [153]. Comprehensive experiments on the video concept detection and document retrieval tasks clearly demonstrate the effectiveness of our two methods.

In the experiments, we adopt the simple but effective nonparametric criterion Maximum Mean Discrepancy (MMD) [37] to define the weight γ_s in (5.32) which measures the distribution relevance between the s -th 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.

Chapter 6

Conclusion and Future Work

Our work is based on the common assumption in transfer learning that there are a limited or even no labeled training samples in the target domain while one can access to data from a single or multiple outer sources whose data distributions are different from that of the target domain. The goal of our work is to learn robust classifiers for the prediction of newly coming samples in the target domain. In this thesis, we propose two frameworks and one method to handle transfer learning problems existing in real-world applications which are related to visual recognition and text categorization. In this chapter, we conclude the contributions of our proposed frameworks and methods and discuss future research topics of transfer learning.

6.1 Conclusion

We conclude this thesis by summarizing the contributions of our proposed frameworks and methods as follows:

- We propose a new cross-domain kernel learning framework, referred to as Domain Transfer Multiple Kernel Learning (DTMKL), to cope with the considerable change between feature distributions of different domains. Many existing kernel methods including Support Vector Machine (SVM), Support Vector Regression (SVR), Kernel Regularized Least-Squares (KRLS) and so on, can be readily incorporated into this framework to tackle transfer learning problems. The DTMKL framework simultaneously learns a kernel function and a robust classifier by minimizing both

the structural risk functional and the distribution mismatch between the labeled and unlabeled samples from the source and target domains. Based on Multiple Kernel Learning (MKL), we assume that the kernel function is a linear combination of a set of base kernel functions in DTMKL. Moreover, we develop a reduced gradient descent procedure to efficiently and effectively learn the linear combination coefficients of multiple base kernels as well as the target classifier. Under the DTMKL framework, we also develop two novel methods:

- By using the labeled training samples from both the source domain and the target domain, we propose DTMKL-AT by employing the structural risk functional of Support Vector Machine (SVM) together with the hinge loss function in DTMKL.
- Motivated by the utilization of pre-learned classifiers in A-SVM [30], we propose another method called DTMKL- f in which a new regularizer is developed to enforce that the decision values from the target classifier and the pre-learned classifiers are similar on the unlabeled target samples.

Comprehensive experiments on three transfer learning data sets (*i.e.*, TRECVID, 20 Newsgroups and email spam data sets) demonstrate that DTMKL based methods outperform existing transfer learning and multiple kernel learning methods. We also experimentally analyze the regularization parameters and the performance variations by using different number of labeled target samples for different transfer learning methods, as well as show the convergence of the objective values of DTMKL-AT.

- We propose a visual event recognition framework for consumer videos by leveraging a large amount of loosely labeled web videos (*e.g.*, from YouTube). Observing that consumer videos contain large intra-class variations within the same type of events, we first propose a new aligned space-time pyramid matching method called Aligned Space-Time Pyramid Matching (ASTPM) to measure the distances between two video clips. In contrast to the fixed volume-to-volume matching in [166], the space-time volumes of two videos across different space-time locations may be

matched by using our proposed ASTPM method, making it better at coping with the large intra-class variations within the same type of events. Second, we propose a new transfer learning method, referred to as Adaptive Multiple Kernel Learning (A-MKL), in order to 1) fuse the information from multiple pyramid levels and features (*i.e.*, space-time features and static SIFT features) and 2) cope with the considerable variation in feature distributions between videos from two domains (*i.e.*, web video domain and consumer video domain). For each pyramid level and each type of local features, we first train a set of SVM classifiers based on the combined training set from two domains by using multiple base kernels from different kernel types and parameters, which are fused with equal weights to obtain a pre-learned average classifier. In A-MKL, we learn an adapted target classifier for each event class based on multiple base kernels and the pre-learned average classifiers from this event class or all the event classes by minimizing both the structural risk functional and mismatch between data distributions of two domains. Similar to DTMKL, we employ a reduced gradient descent procedure to solve the optimization problem in A-MKL. Extensive experiments demonstrate the effectiveness of our proposed framework that requires only a small number of labeled consumer videos by leveraging web data. We also conduct in-depth investigation on various aspects of the proposed A-MKL such as the analysis on the combination coefficients on the pre-learned classifiers, the convergence of the learning algorithm, and the performance variation by using different proportions of labeled consumer videos. Moreover, we show that A-MKL using the pre-learned classifiers from all the event classes leads to better performance when compared with A-MKL using the pre-learned classifiers only from each individual event class.

- We propose a new framework called Domain Adaptation Machine (DAM) for the multiple source domain adaption problem. Under this framework, we learn a robust target classifier for label prediction of samples from the target domain by leveraging a set of base classifiers which are pre-learned classifiers using the labeled samples either from the source domains or from the source domains and the target domain. Any classifiers such as the standard SVM classifier can be readily used as the base classifiers in our framework. Motivated by manifold regularization [192] and the

graph based multi-task learning [197–199], 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 base classifiers on the unlabeled samples from the target domain. This newly proposed regularizer can be readily incorporated into many kernel methods (*e.g.*, SVM, SVR, least-squares SVM and so on), and consequently extend these algorithms to the corresponding domain adaptation methods. To the best of our knowledge, the smoothness assumption encoded in the domain-dependent regularizer is the first to be introduced into 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 least-squares 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 sample is very fast in FastDAM.
- UniverDAM is motivated by the recent work [200, 201] which indicated the samples belonging to neither the positive class nor the negative class can be used as an additional data collection called *Universe* [26] to improve the generalization ability of SVM for the binary classification task. We additionally make use of the samples from the source domains as *Universe* to further enhance the generalization ability of the target classifier. Specifically, we introduce our newly proposed data-dependent regularizer and the *Universe* regularizer into least-squares SVM for UniverDAM.

We evaluate our two DAM-based methods on the challenging TRECVID 2005 data set for the large-scale video concept detection task as well as on the 20 Newsgroups and email spam data sets for document retrieval. For video concept detection, the experimental results 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. For document retrieval, the comprehensive experiments on the two data sets also demonstrate the effectiveness of FastDAM and UniverDAM. UniverDAM achieves the best document retrieval performances on both data sets because of the successful utilization of *Universum* (*i.e.*, the samples from multiple source domains). We also analyze in depth the parameters for different methods by showing the performance variations on the 20 Newsgroups data set.

6.2 Future Work

In this section, we discuss the following research topics of transfer learning which deserve future attempts.

6.2.1 Avoiding Negative Transfer

In transfer learning, it is commonly assumed that there are only limited or even no labeled training samples in the target domain. And one can make use of the data from outer source(s) to help learning in the target domain. If the relationship between the source and target domains is strong (*e.g.*, the data distributions of the source and target domains overlap with each other to some extent), the performance in the target domain can be significantly improved through such positive transfer. Otherwise, the performance may decrease, where negative transfer occurs [196]. To avoid such negative transfer, one has to investigate how to select part of all the source data which can strongly relate to the samples in the target domain and thus produce more positive transfer. For this sake, either *choosing source samples* or *choosing entire single source(s)* can be considered in future investigation. In other words, we have to recognize and reject harmful source samples which may result in negative transfer for the target domain. By measuring the relatedness between the source and target domains, Rosenstein *et al.* [196] designed a hierarchical naive Bayes method to avoid negative transfer. Such relationship has also been analyzed in some Multi-Task Learning (MTL) methods [209–211], which also provides us guidance of rejecting irrelevant source tasks for the avoidance of negative transfer. More recently, Xiang *et al.* [212] proposed to embed all classes into a latent Euclidean space using a graph representation for transfer learning such that one no longer

needs to select task-specific source data. Inspired by Transductive SVM (T-SVM) [193], Bruzzone and Marconcini [13] proposed Domain Adaptation Support Vector Machine (DASVM) by learning the target classifier step by step. At each step, DASVM labels the unlabeled training samples from the target domain and simultaneously removes some source labeled samples that are unlikely to help learn a target classifier. One may also learn an importance weight for each source sample by employing techniques from sample selection bias [46–48] or covariate shift [43, 45]. Then we are able to select source samples by setting an acceptable threshold of the importance.

During these two decades, how to avoid negative transfer is still an open issue and also one of the fundamental questions in transfer learning, which deserves our efforts and will definitely attract more and more attention in the future.

6.2.2 Speed-Up in the Large-Scale Setting

In today’s heavily networked world, thousands of millions of data in various formats (*e.g.*, texts, images, videos, etc.) are available on the Internet. When people are dealing with the task at hand, a natural and effective way is to ask for the help from the Internet data. It is very likely that the data people have are different from the data from the Internet in terms of the feature distribution of data. In the computer vision community, several methods have been proposed to learn robust target classifiers by leveraging the web data from online sources (*e.g.*, *Amazon*, *Flickr*, *photoSIG*, *YouTube*, image/video search engines powered by *Google*, *Microsoft* and *Yahoo!*, and so on) [15–17, 22, 39, 213]. However, the number of the source samples used in most of these methods is in a relatively small scale (*i.e.*, less than ten thousand source samples) except for the work [15] in which more than one million web images downloaded from **photoSIG** are used to help learn a robust classifier to index and retrieval the consumer photos. Specifically, Liu *et al.* first employed the efficient decision stumps [214] to obtain a source classifier trained by using the web images. And then a transfer learning algorithm, which can be considered as a regularized-regression version of FastDAM [54] by using a linear kernel, is further proposed to achieve the real-time retrieval performance. Our proposed method FastDAM in Chapter 5 also enjoys fast prediction. Moreover, non-linear kernels, which are known to be usually more effective than a linear kernel for computer vision tasks, can be readily

employed in FastDAM. In addition, FastDAM can benefit from the knowledge transferred from multiple source domains, which is intuitively more effective than those transfer learning methods [15–17, 22, 39, 213] that can only handle a single source domain case. We will seek for more efficient learning algorithms (such as decision stumps [214]) and incorporate them in our proposed FastDAM method to further speed up both the training and testing phrases. More than that, we will also investigate the feasibility of introducing modern Hashing techniques (such as Locality Sensitive Hashing (LSH) [215] and Spectral Hashing [216]) into transfer learning when dealing with a very large number of data.

6.2.3 Transfer Learning in Computer Vision Applications

Although transfer learning has been studied for years in some fields (such as natural language processing [18, 23, 29, 86], sentiment classification [19], document classification [20, 21] and WiFi localization [24, 217]), it is still at a very early stage in the computer vision community. Vision researchers are now working on a couple of transfer learning problems such as the reduction of the distribution mismatch between data distributions of two domains [17, 36], the adaptation of cross-category knowledge to a new category domain [77], the attribute-based knowledge transfer by mining semantic relatedness [111, 114, 218], and the adaption between two domains with totally different feature spaces [16, 39].

For knowledge sharing¹ based on attributes, Rohrbach *et al.* [218] revealed that the current attribute-based methods [111, 114] do not outperform the simple one-vs-all SVM classifier in a large-scale setting, which is against the original expectation that the knowledge shared among different classes may benefit classification. One possible reason is that the graphical models proposed in [111, 114] do not possess the discriminative power as much as a standard SVM does. Therefore, to improve the classification performance, we need to focus more on discriminative models and also find ways to incorporate the shared knowledge into them.

¹In the knowledge sharing setting, it is assumed that training data are available for all classes. Knowledge can be shared among all classes, and thus it is expected to achieve better classification performance than the one-vs-all setting (*i.e.*, one classifier is learned based on the positive and negative training samples only from this class).

Another interesting direction related to computer vision applications using web data is introduced in [39], in which the feature representation of the source data (*i.e.*, web images) is entirely different from that of the target data (*i.e.*, consumer images)². Kulis *et al.* [39] proposed to explore the relationship between the two entire different feature spaces by learning a distance metric. However, the question why to represent consumer images by using SIFT features [183] and web images by using SURF features [219] has not been answered by the authors in [39]. Moreover, the method proposed in [39] only focuses on the learning of the common space shared by the two types of features. In such a case, we argue that the domain-specific knowledge, on the contrary to the common knowledge shared by two domains, should also be of equal importance. Therefore, finding a meaningful application using web data (*i.e.*, the reason for using a specific type of features to represent data from one domain makes sense) and utilizing the domain-specific knowledge are possible ways to continue this research direction.

²In the literature, this type of transfer learning problems is called Translated Learning, firstly addressed by Dai *et al.* [38]. In [38], the user preferences are translated between the text and image domains for image classification.

References

- [1] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., 3rd ed., 2011.
- [2] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2nd ed., 2007.
- [3] T. Hastie, R. Tibshirani, and J. Friedma, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2nd ed., 2009.
- [4] T. Mitchell, *Machine Learning*. McGraw-Hill, 1997.
- [5] V. Vapnik, *Statistical learning theory*. Wiley-Interscience, 1998.
- [6] O. Chapelle, B. Schölkopf, and A. Zien, *Pattern Recognition and Machine Learning*. MIT Press, 2006.
- [7] X. Zhu, “Semi-supervised learning literature survey,” Computer Sciences Technical Report 1530, University of Wisconsin-Madison, July 2008.
- [8] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, “Active learning with statistical models,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, March 1996.
- [9] D. D. Lewis and W. A. Gale, “A sequential algorithm for training text classifiers,” in *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (Dublin, Ireland), pp. 3–12, Springer-Verlag, July 2009.
- [10] B. Settles, “Active learning literature survey,” Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

- [11] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, pp. 41–75, July 1997.
- [12] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, October 2010.
- [13] L. Bruzzone and M. Marconcini, “Domain adaptation problems: a dasvm classification technique and a circular validation strategy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 770–787, May 2010.
- [14] 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 *Proceedings of the NIST TRECVID Workshop*, (Gaithersburg, Maryland, United States), TRECVID, November 2008.
- [15] Y. Liu, D. Xu, I. W. Tsang, and J. Luo, “Textual query of personal photos facilitated by large-scale web data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 1022–1036, May 2011.
- [16] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in *Proceedings of the European Conference on Computer Vision*, (Heraklion, Crete, Greece), pp. 213–226, Springer-Verlag, September 2010.
- [17] L. Duan, D. Xu, I. W. Tsang, and J. Luo, “Visual event recognition in videos by learning from web data,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (San Francisco, California, United States), pp. 1959–1966, IEEE Computer Society, June 2010.
- [18] N. Bel, C. Koster, and M. Villegas, “Cross-lingual text categorization,” in *Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries*, (Trondheim, Norway), pp. 126–139, Springer-Verlag, August 2003.
- [19] J. Blitzer, M. Dredze, and F. Pereira, “Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, (Prague, Czech Republic), pp. 440–447, ACL, June 2007.

- [20] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, “Boosting for transfer learning,” in *Proceedings of International Conference on Machine Learning*, (Corvalis, Oregon, United States), pp. 193–200, ACM, June 2007.
- [21] R. Raina, A. Y. Ng, and D. Koller, “Constructing informative priors using transfer learning,” in *Proceedings of International Conference on Machine Learning*, (Pittsburgh, Pennsylvania, United States), pp. 713–720, ACM, June 2006.
- [22] Q. Yang, Y. Chen, G.-R. Xue, W. Dai, and Y. Yu, “Heterogeneous transfer learning for image clustering via the social web,” in *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, (Suntec, Singapore), pp. 1–9, ACL, August 2009.
- [23] J. Blitzer, R. McDonald, and F. Pereira, “Domain adaptation with structural correspondence learning,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (Sydney, Australia), pp. 120–128, ACL, July 2006.
- [24] S. J. Pan, J. T. Kwok, Q. Yang, and J. J. Pan, “Adaptive localization in a dynamic wifi environment through multi-view learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, (Vancouver, British Columbia, Canada), pp. 1108–1113, AAAI Press, July 2007.
- [25] Q. Yang, “Activity recognition: Linking low-level sensors to high-level intelligence,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, (Pasadena, California, United States), pp. 20–25, Morgan Kaufmann Publishers Inc., July 2009.
- [26] V. Vapnik, *Semi-Supervised Learning*, ch. Transductive Inference and Semi-Supervised Learning, pp. 454–472. MIT press, 2006.
- [27] E. Thorndike and R. Woodworth, “The influence of improvement in one mental function upon the efficiency of other functions: Functions involving attention, observation and discrimination,” *Psychological Review*, vol. 8, pp. 553–564, November 1901.

- [28] H. C. Elli, *The Transfer of Learning*. The Macmillan Company, 1965.
- [29] H. Daumé III, “Frustratingly easy domain adaptation,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, (Prague, Czech Republic), pp. 256–263, ACL, June 2007.
- [30] J. Yang, R. Yan, and A. G. Hauptmann, “Cross-domain video concept detection using adaptive svms,” in *Proceedings of the International Conference on Multimedia*, (Augsburg, Germany), pp. 188–197, ACM, September 2007.
- [31] 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*, pp. 1433–1440, MIT Press, 2009.
- [32] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” in *Proceedings of the European Conference on Computational Learning Theory*, (Barcelona, Spain), pp. 23–37, Springer-Verlag, March 1995.
- [33] P. Wu and T. G. Dietterich, “Improving svm accuracy by training on auxiliary data sources,” in *Proceedings of International Conference on Machine Learning*, (Banff, Alberta, Canada), pp. 871–878, ACM, July 2004.
- [34] W. Jiang, E. Zavesky, S.-F. Chang, and A. Loui, “Cross-domain learning methods for high-level visual concept classification,” in *Proceedings of the IEEE International Conference on Image Processing*, (San Diego, California, United States), pp. 161–164, IEEE, October 2008.
- [35] Y.-G. Jiang, J. Wang, S.-F. Chang, and C.-W. Ngo, “Domain adaptive semantic diffusion for large scale context-based video annotation,” in *Proceedings of the IEEE International Conference on Computer Vision*, (Kyoto, Japan), pp. 1420–1427, IEEE Computer Society, September 2009.
- [36] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank, “Domain transfer svm for video concept detection,” in *Proceedings of the IEEE International Conference*

- on *Computer Vision and Pattern Recognition*, (Miami, Florida, United States), pp. 1375–1381, IEEE Computer Society, June 2009.
- [37] 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, pp. 49–57, July 2006.
- [38] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, “Translated learning: Transfer learning across different feature spaces,” in *Advances in Neural Information Processing Systems 21*, pp. 353–360, MIT Press, 2009.
- [39] B. Kulis, K. Saenko, and T. Darrell, “What you saw is not what you get: Domain adaptation using asymmetric kernel transforms,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (Colorado Springs, United States), pp. 1785–1792, IEEE Computer Society, June 2011.
- [40] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, “Information-theoretic metric learning,” in *Proceedings of International Conference on Machine Learning*, (Corvalis, Oregon, United States), pp. 209–216, ACM, June 2007.
- [41] A. Arnold, R. Nallapati, and W. W. Cohe, “A comparative study of methods for transductive transfer learning,” in *Proceedings of the IEEE International Conference on Data Mining Workshops*, (Omaha, Nebraska, United States), pp. 77–82, IEEE Computer Society, October 2007.
- [42] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. MIT Press, 2009.
- [43] H. Shimodaira, “Improving predictive inference under covariate shift by weighting the log-likelihood function,” *Journal of Statistical Planning and Inference*, vol. 90, pp. 227–244, February 2000.
- [44] A. J. Storkey and M. Sugiyama, “Mixture regression for covariate shift,” in *Advances in Neural Information Processing Systems 19*, pp. 1337–1344, MIT Press, 2007.

- [45] M. Sugiyama and K.-R. Müller, “Input-dependent estimation of generalization error under covariate shift,” *Statistics and Decisions*, vol. 23, no. 4, pp. 249–279, 2005.
- [46] M. Dudík, R. Schapire, and S. Phillips, “Correcting sample selection bias in maximum entropy density estimation,” in *Advances in Neural Information Processing Systems 18*, pp. 323–330, MIT Press, 2006.
- [47] 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*, pp. 601–608, MIT Press, 2007.
- [48] B. Zadrozny, “Learning and evaluating classifiers under sample selection bias,” in *Proceedings of International Conference on Machine Learning*, (Banff, Alberta, Canada), pp. 114–121, ACM, July 2004.
- [49] J. Blitzer, *Domain adaptation of natural language processing systems*. PhD thesis, University of Pennsylvania, January 2008.
- [50] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, “Transferring naive bayes classifiers for text classification,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, (Vancouver, British Columbia, Canada), pp. 540–545, AAAI Press, July 2007.
- [51] D. D. Lewis, *Representation and learning in information retrieval*. PhD thesis, University of Massachusetts - Amherst, January 1992.
- [52] X.-J. Wang, L. Zhang, X. Li, and W.-Y. Ma, “Annotating images by mining image search results,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1919–1932, November 2008.
- [53] Z. Wang, Y. Song, and C. Zhang, “Transferred dimensionality reduction,” in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, (Antwerp, Belgium), pp. 550–565, Springer-Verlag, September 2008.

- [54] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua, “Domain adaptation from multiple sources via auxiliary classifiers,” in *Proceedings of International Conference on Machine Learning*, (Montreal, Quebec, Canada), pp. 289–296, ACM, June 2009.
- [55] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” *ACM Computing Surveys*, vol. 31, pp. 264–323, September 1999.
- [56] R. Xu and D. Wunsch II, “Survey of clustering algorithms,” *IEEE Transactions on Neural Networks*, vol. 16, pp. 645–678, May 2005.
- [57] I. K. Fodor, “A survey of dimension reduction techniques,” Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory, May 2002.
- [58] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, pp. 2323–2326, December 2000.
- [59] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, “Self-taught clustering,” in *Proceedings of International Conference on Machine Learning*, (Helsinki, Finland), pp. 200–207, ACM, July 2008.
- [60] I. S. Dhillon, “Co-clustering documents and words using bipartite spectral graph partitioning,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (San Francisco, California, United States), pp. 269–274, ACM, August 2001.
- [61] S. J. Pan, J. T. Kwok, and Q. Yang, “Transfer learning via dimensionality reduction,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, (Chicago, Illinois, United States), pp. 677–682, AAAI Press, July 2008.
- [62] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” *IEEE Transactions on Neural Networks*, vol. 22, pp. 199–210, February 2011.
- [63] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, “Analysis of representations for domain adaptation,” in *Advances in Neural Information Processing Systems 19*, pp. 137–144, MIT Press, 2007.

- [64] S. Ben-David, T. Luu, T. Lu, and D. Pál, “Impossibility theorems for domain adaptation,” *Journal of Machine Learning Research*, vol. 9, pp. 129–136, May 2010.
- [65] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” *Machine Learning*, vol. 79, pp. 151–175, May 2010.
- [66] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, “Learning bounds for domain adaptation,” in *Advances in Neural Information Processing Systems 20*, pp. 129–136, MIT Press, 2008.
- [67] K. Crammer, M. Kearns, and J. Wortman, “Learning from multiple sources,” *Journal of Machine Learning Research*, vol. 9, pp. 1757–1774, August 2008.
- [68] Y. Mansour, M. Mohri, and A. Rostamizadeh, “Domain adaptation with multiple sources,” in *Advances in Neural Information Processing Systems 21*, pp. 1041–1048, MIT Press, 2009.
- [69] Y. Mansour, M. Mohri, and A. Rostamizadeh, “Domain adaptation: Learning bounds and algorithms,” *Computing Research Repository*, vol. abs/0902.3430, 2009.
- [70] D. Kifer, S. Ben-David, and J. Gehrke, “Detecting change in data streams,” in *Proceedings of the International Conference on Very large Data Bases*, (Toronto, Canada), pp. 180–191, VLDB Endowment, August 2004.
- [71] Y. S. Chan and H. T. Ng, “Domain adaptation with active learning for word sense disambiguation,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, (Prague, Czech Republic), pp. 49–56, ACL, June 2007.
- [72] A. Harpale and Y. Yang, “Active learning for multi-task adaptive filtering,” in *Proceedings of International Conference on Machine Learning*, (Haifa, Israel), p. 431–438, Omnipress, June 2010.
- [73] X. Liao, Y. Xue, and L. Carin, “Logistic regression with an auxiliary data source,” in *Proceedings of International Conference on Machine Learning*, (Bonn, Germany), pp. 505–512, ACM, June 2005.

- [74] S. Rajan, J. Ghosh, and M. M. Crawford, “An active learning approach to knowledge transfer for hyperspectral data analysis,” in *Proceedings of the IEEE International Conference on Geoscience and Remote Sensing Symposium*, (Denver, Colorado, United States), pp. 541–544, IEEE, July 2006.
- [75] X. Shi, W. Fan, and J. Ren, “Actively transfer domain knowledge,” in *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases*, (Antwerp, Belgium), pp. 342–357, Springer-Verlag, September 2008.
- [76] B. M. Shahshahani and D. A. Landgrebe, “The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, pp. 1087–1095, September 1994.
- [77] G.-J. Qi, C. Aggarwal, Y. Rui, Q. Tian, S. Chang, and T. Huang, “Towards cross-category knowledge propagation for learning visual concepts,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (Colorado Springs, United States), pp. 897–904, IEEE Computer Society, June 2011.
- [78] Z.-J. Zha, T. Mei, M. Wang, Z. Wang, and X.-S. Hua, “Robust distance metric learning with auxiliary knowledge,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, (Pasadena, California, United States), pp. 1327–1332, Morgan Kaufmann Publishers Inc., July 2009.
- [79] Y. Zhang and D.-Y. Yeung, “Transfer metric learning by learning task relationships,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (Washington, D.C., United States), pp. 1199–1208, ACM, July 2010.
- [80] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, vol. 65, pp. 386–408, November 1958.

- [81] S. Shalev-Shwartz, *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, Hebrew University, July 2007.
- [82] P. Zhao and S. C. H. Hoi, “Otl: A framework of online transfer learning,” in *Proceedings of International Conference on Machine Learning*, (Haifa, Israel), pp. 1231–1238, Omnipress, June 2010.
- [83] O. Dekel, P. M. Long, and Y. Singer, “Online learning of multiple tasks with a shared loss,” *Journal of Machine Learning Research*, vol. 8, pp. 2233–2264, December 2007.
- [84] R. K. Ando and T. Zhang, “A framework for learning predictive structures from multiple tasks and unlabeled data,” *Journal of Machine Learning Research*, vol. 6, pp. 1817–1853, December 2005.
- [85] H. L. Guo, L. Zhang, and Z. Su, “Empirical study on the performance stability of named entity recognition model across domains,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (Sydney, Australia), pp. 509–516, ACL, July 2006.
- [86] J. Jiang and C. X. Zha, “Instance weighting for domain adaptation in nlp,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, (Prague, Czech Republic), pp. 204–271, ACL, June 2007.
- [87] X. Ling, G.-R. Xue, W. Dai, Y. Jiang, Q. Yang, and Y. Yu, “Can chinese web pages be classified with english data source?,” in *Proceedings of the International Conference on World Wide Web*, (Beijing, China), pp. 969–978, ACM, April 2008.
- [88] A. E. Richman and P. Schone, “Empirical study on the performance stability of named entity recognition model across domains,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, (Columbus, Ohio, United States), pp. 1–9, ACL, June 2008.
- [89] D. Wu, W. S. Lee, N. Ye, and H. L. Chieu, “Empirical study on the performance stability of named entity recognition model across domains,” in *Proceedings of the*

- Conference on Empirical Methods in Natural Language Processing*, (Singapore), pp. 1523–1532, ACL, August 2008.
- [90] K. Liu and J. Zhao, “Cross-domain sentiment classification using a two-stage method,” in *Proceedings of the ACM Conference on Information and Knowledge Management*, (Hong Kong, China), pp. 1717–1720, ACM, November 2009.
- [91] T. Li, C. Ding, Y. Zhang, and B. Shao, “Knowledge transformation for cross-domain sentiment classification,” in *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (Boston, Massachusetts, United States), pp. 716–717, ACM, July 2009.
- [92] T. Li, Y. Zhang, and V. Sindhwani, “A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge,” in *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, (Suntec, Singapore), pp. 244–252, ACL, August 2009.
- [93] T. Li, V. Sindhwani, C. H. Q. Ding, and Y. Zhang, “Bridging domains with words: Opinion analysis with matrix tri-factorizations,” in *Proceedings of the SIAM International Conference on Data Mining*, (Columbus, Ohio, United States), pp. 293–302, SIAM, April 2010.
- [94] C.-W. Seah, I. W. Tsang, Y.-S. Ong, and K.-K. Lee, “Predictive distribution matching svm for multi-domain learning,” in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, (Barcelona, Spain), pp. 231–247, Springer-Verlag, September 2010.
- [95] S. Tan, X. Cheng, Y. Wang, and H. Xu, “Adapting naive bayes to domain adaptation for sentiment analysis,” in *Advances in Information Retrieval - European Conference on IR Research*, (Toulouse, France), pp. 337–349, Springer, April 2009.

- [96] Q. Wu, S. Tan, H. Zhai, G. Zhang, M. Duan, and X. Cheng, “Sentirank: Cross-domain graph ranking for sentiment classification,” in *Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, (Milano, Italy), pp. 309–314, IEEE Computer Society, September 2009.
- [97] B. Chen, W. Lam, I. W. Tsang, and T.-L. Wong, “Location and scatter matching for dataset shift in text mining,” in *Proceedings of the IEEE International Conference on Data Mining*, (Sydney, Austrilia), pp. 773–778, IEEE Computer Society, December 2009.
- [98] P. Luo, F. Zhuang, H. Xiong, Y. Xiong, and Q. He, “Transfer learning from multiple source domains via consensus regularization,” in *Proceeding of the ACM Conference on Information and Knowledge Management*, (Napa Valley, California, United States), pp. 103–112, ACM, October 2008.
- [99] J. Gao, W. Fan, J. Jiang, and J. Han, “Knowledge transfer via multiple model local structure mapping,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (Las Vegas, Nevada, United States), pp. 283–291, ACM, September 2008.
- [100] X. Ling, W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, “Spectral domain-transfer learning,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (Las Vegas, Nevada, United States), pp. 488–496, ACM, September 2008.
- [101] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, “Self-taught learning: Transfer learning from unlabeled data,” in *Proceedings of International Conference on Machine Learning*, (Corvalis, Oregon, United States), pp. 759–766, ACM, June 2007.
- [102] D. Xing, W. Dai, G. Xue, and Y. Yu, “Bridged refinement for transfer learning,” in *Proceedings of the European Conference on Practice of Knowledge Discovery in Databases*, (Warsaw, Poland), pp. 324–335, Springer, September 2007.

- [103] F. Zhuang, P. Luo, H. Xiong, Y. Xiong, Q. He, and Z. Shi, “Cross-domain learning from multiple sources: A consensus regularization perspective,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1664–1678, December 2010.
- [104] F. Zhuang, P. Luo, H. Xiong, Q. He, Y. Xiong, and Z. Shi, “Exploiting associations between word clusters and document classes for cross-domain text categorization,” *Statistical Analysis and Data Mining*, vol. 4, pp. 100–114, February 2011.
- [105] J. Gao, W. Fan, Y. Sun, and J. Han, “Heterogeneous source consensus learning via decision propagation and negotiation,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (Paris, France), pp. 339–348, ACM, November 2009.
- [106] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, “Topic-bridged pls for cross-domain text classification,” in *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (Singapore), pp. 627–634, ACM, July 2008.
- [107] Q. Gu and J. Zhou, “Learning the shared subspace for multi-task clustering and transductive transfer classification,” in *Proceedings of the IEEE International Conference on Data Mining*, (Miami, Florida, United States), pp. 159–168, IEEE Computer Society, December 2009.
- [108] T. Li, C. Ding, Y. Zhang, and B. Shao, “Knowledge transformation from word space to document space,” in *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (Singapore), pp. 187–194, ACM, July 2008.
- [109] L. Chen, D. Xu, I. W. Tsang, and J. Luo, “Tag-based web photo retrieval improved by batch mode re-tagging,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (San Francisco, California, United States), pp. 3440–3446, IEEE Computer Society, June 2010.

- [110] R. Gopalan, R. Li, and R. Chellappa, “Domain adaptation for object recognition: An unsupervised approach,” in *Proceedings of the IEEE International Conference on Computer Vision*, (Barcelona, Spain), pp. 999–1006, IEEE Computer Society, November 2011.
- [111] C. H. Lampert, H. Nickisch, and S. Harmeling, “Learning to detect unseen object classes by between-class attribute transfer,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (Miami, Florida, United States), pp. 951–958, IEEE Computer Society, June 2009.
- [112] C. H. Lampert and O. Krömer, “Weakly-paired maximum covariance analysis for multimodal dimensionality reduction and transfer learning,” in *Proceedings of the European Conference on Computer Vision*, (Heraklion, Crete, Greece), pp. 566–579, Springer-Verlag, September 2010.
- [113] L. Jie, T. Tommasi, and B. Caputo, “Multiclass transfer learning from unconstrained priors,” in *Proceedings of the IEEE International Conference on Computer Vision*, (Barcelona, Spain), pp. 1863–1870, IEEE Computer Society, November 2011.
- [114] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele, “What helps where – and why? semantic relatedness for knowledge transfer,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (San Francisco, California, United States), pp. 910–917, IEEE Computer Society, June 2010.
- [115] T. Tommasi, F. Orabona, and B. Caputo, “Safety in numbers: Learning categories from few examples with multi model knowledge transfer,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (San Francisco, California, United States), pp. 3081–3088, IEEE Computer Society, June 2010.
- [116] X.-T. Yuan and S. Yan, “Visual classification with multi-task joint sparse representation,” in *Proceedings of the IEEE International Conference on Computer Vision*

- and Pattern Recognition*, (San Francisco, California, United States), pp. 3493–3500, IEEE Computer Society, June 2010.
- [117] S. K. Gupta, D. Phung, B. Adams, T. Tran, and S. Venkatesh, “Nonnegative shared subspace learning and its application to social media retrieval,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (Washington, D.C., United States), pp. 1169–1178, ACM, July 2010.
- [118] R. Yan and J. Zhang, “Transfer learning using task-level features with application to information retrieval,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, (Pasadena, California, United States), pp. 1315–1320, Morgan Kaufmann Publishers Inc., July 2009.
- [119] J. Yang and A. G. Hauptmann, “A framework for classifier adaptation and its applications in concept detection,” in *Proceedings of the ACM International Workshop on Multimedia Information Retrieval*, (Vancouver, British Columbia, Canada), pp. 467–474, ACM, October 2008.
- [120] B. Cao, N. N. Liu, and Q. Yang, “Transfer learning for collective link prediction in multiple heterogenous domains,” in *Proceedings of International Conference on Machine Learning*, (Haifa, Israel), pp. 159–166, Omnipress, June 2010.
- [121] B. Li, Q. Yang, and X. Xue, “Can movies and books collaborate?: cross-domain collaborative filtering for sparsity reduction,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, (Pasadena, California, United States), pp. 2052–2057, Morgan Kaufmann Publishers Inc., July 2009.
- [122] B. Li, Q. Yang, and X. Xue, “Transfer learning for collaborative filtering via a rating-matrix generative model,” in *Proceedings of International Conference on Machine Learning*, (Montreal, Quebec, Canada), pp. 617–624, ACM, June 2009.
- [123] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang, “Transfer learning in collaborative filtering for sparsity reduction,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, (Chicago, Illinois, United States), pp. 230–235, AAAI Press, July 2010.

- [124] Y. Zhang, B. Cao, and D. yan Yeung, “Multi-domain collaborative filtering,” in *Proceedings of the Conference in Uncertainty in Artificial Intelligence*, (Catalina Island, California, United States), AUAI Press, July 2010.
- [125] S. J. Pan, D. Shen, Q. Yang, and J. T. Kwok, “Transferring localization models across space,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, (Chicago, Illinois, United States), pp. 1383–1388, AAAI Press, July 2008.
- [126] P. Rashidi and D. J. Cook, “Activity recognition based on home to home transfer learning,” in *Proceedings of the Workshop on Plan, Activity, and Intent Recognition of the AAAI Conference on Artificial Intelligence*, (Chicago, Illinois, United States), pp. 45–52, AAAI Press, July 2010.
- [127] H.-Y. Wang, V. W. Zheng, J. Zhao, and Q. Yang, “Indoor localization in multi-floor environments with reduced effort,” in *Proceedings of the Annual IEEE International Conference on Pervasive Computing and Communications*, (Mannheim, Germany), pp. 244–252, IEEE Computer Society, March 2010.
- [128] J. Yin, Q. Yang, and L. M. Ni, “Learning adaptive temporal radio maps for signalstrength-based location estimation,” *IEEE Transactions on Mobile Computing*, vol. 7, pp. 869–883, July 2008.
- [129] V. W. Zheng, S. J. Pan, Q. Yang, and J. J. Pan, “Transferring multi-device localization models using latent multi-task learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, (Chicago, Illinois, United States), pp. 1427–1432, AAAI Press, July 2008.
- [130] V. W. Zheng, D. H. Hu, and Q. Yang, “Activity recognition based on home to home transfer learning,” in *Proceedings of the International Conference on Ubiquitous Computing*, (Orlando, Florida, United States), pp. 61–70, ACM, September 2009.
- [131] J. Bai, K. Zhou, G. Xue, H. Zha, G. Sun, B. Tseng, Z. Zheng, and Y. Chang, “Multi-task learning for learning to rank in web search,” in *Proceeding of the ACM Conference on Information and Knowledge Management*, (Hong Kong, China), pp. 1549–1552, ACM, November 2009.

- [132] O. Chapelle, P. Shivaswamy, S. Vadrevu, K. Weinberger, Y. Zhang, and B. Tseng, “Multi-task learning for boosting with application to web search ranking,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (Washington, D.C., United States), pp. 1189–1198, ACM, July 2010.
- [133] W. Gao, P. Cai, K.-F. Wong, and A. Zhou, “Learning to rank only using training data from related domain,” in *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (Geneva, Switzerland), pp. 162–169, ACM, July 2010.
- [134] B. Wang, J. Tang, W. Fan, S. Chen, Z. Yang, and Y. Liu, “Heterogeneous cross domain ranking in latent space,” in *Proceeding of the ACM Conference on Information and Knowledge Management*, (Hong Kong, China), pp. 987–996, ACM, November 2009.
- [135] D. Chen, J. Yan, G. Wang, Y. Xiong, W. Fan, and Z. Chen, “Transrank: A novel algorithm for transfer of rank learning,” in *Proceedings of the IEEE International Conference on Data Mining Workshops*, (Pisa, Italy), pp. 106–115, IEEE Computer Society, December 2008.
- [136] D. Chen, Y. Xiong, J. Yan, G.-R. Xue, G. Wang, and Z. Chen, “Knowledge transfer for cross domain learning to rank,” *Information Retrieval*, vol. 13, pp. 236–253, June 2010.
- [137] B. Geng, L. Yang, C. Xu, and X.-S. Hua, “Ranking model adaptation for domain-specific search,” in *Proceeding of the ACM Conference on Information and Knowledge Management*, (Hong Kong, China), pp. 197–206, ACM, November 2009.
- [138] J. Jiang and C. Zhai, “A two-stage approach to domain adaptation for statistical classifiers,” in *Proceedings of the ACM Conference on Information and Knowledge Management*, (Lisbon, Portugal), pp. 401–410, ACM, November 2008.
- [139] E. Zeng, C. Yang, T. Li, and G. Narasimhan, “Clustering genes using heterogeneous data sources,” *International Journal of Knowledge Discovery in Bioinformatics*, vol. 1, no. 2, pp. 12–28, 2010.

- [140] C. Widmer, J. Leiva, Y. Altun, and G. Rätsch, “Leveraging sequence classification by taxonomy-based multitask learning,” in *Proceedings of the Annual International Conference on Research in Computational Molecular Biology*, (Lisbon, Portugal), pp. 522–534, Springer, April 2010.
- [141] X. Yang, S. Kim, and E. Xing, “Heterogeneous multitask learning with joint sparsity constraints,” in *Advances in Neural Information Processing Systems 22*, pp. 2151–2159, MIT Press, 2009.
- [142] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proceedings of the Annual Conference on Computational Learning Theory*, (Madison, Wisconsin, United States), pp. 92–100, ACM, July 1998.
- [143] G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, “Learning the kernel matrix with semidefinite programming,” *Journal of Machine Learning Research*, vol. 5, pp. 27–72, December 2004.
- [144] L. Vandenberghe and S. Boyd, “Semidefinite programming,” *Society for Industrial and Applied Mathematics Review*, vol. 38, pp. 49–95, March 1996.
- [145] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, “Multiple kernel learning, conic duality, and the smo algorithm,” in *Proceedings of International Conference on Machine Learning*, (Banff, Alberta, Canada), pp. 6–13, ACM, July 2004.
- [146] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, “Large scale multiple kernel learning,” *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, December 2006.
- [147] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, “Simplemkl,” *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, November 2008.
- [148] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, “Hierarchical spatio-temporal context modeling for action recognition,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (Miami, Florida, United States), pp. 2004–2011, IEEE Computer Society, June 2009.

- [149] X. Wu, D. Xu, L. Duan, and J. Luo, “Action recognition using context and appearance distribution features,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (Colorado Springs, United States), pp. 489–496, IEEE Computer Society, June 2011.
- [150] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, “Multiple kernels for object detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, (Kyoto, Japan), pp. 606–613, IEEE Computer Society, September 2009.
- [151] B. Schölkopf and A. Smola, *Learning with kernels*. MIT Press, 2002.
- [152] A. Zien and C. S. Ong, “Multiclass multiple kernel learning,” in *Proceedings of International Conference on Machine Learning*, (Corvalis, Oregon, United States), pp. 1191–1198, ACM, June 2007.
- [153] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines.” Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- [154] J. C. Platt, *Advances in Kernel Methods: Support Vector Learning*, ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208. MIT Press, 1999.
- [155] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [156] 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*, vol. 13, pp. 86–91, July 2006.
- [157] A. Yanagawa, W. Hsu, and S.-F. Chang, “Columbia university’s baseline detectors for 374 lscm semantic visual concepts,” ADVENT Technical Report #222-2006-8, Columbia University, March 2007.
- [158] A. F. Smeaton, P. Over, and W. Kraaij, “Evaluation campaigns and TRECVID,” in *Proceedings of the ACM International Workshop on Multimedia Information Retrieval*, (Santa Barbara, California, United States), pp. 321–330, ACM, October 2006.

- [159] D. Xu and S.-F. Chang, “Video event recognition using kernel methods with multilevel temporal alignment,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1985–1997, May 2008.
- [160] A. Corduneanu and T. Jaakkola, “Continuation methods for mixing heterogeneous sources,” in *Proceedings of the Conference in Uncertainty in Artificial Intelligence*, (Edmonton, Alberta, Canada), pp. 111–118, Morgan Kaufmann Publishers Inc., August 2002.
- [161] S.-F. Chang, D. Ellis, W. Jiang, K. Lee, A. Yanagawa, A. C. Loui, and J. Luo, “Large-scale multimodal semantic concept detection for consumer video,” in *Proceedings of the ACM International Workshop on Multimedia Information Retrieval*, pp. 255–264, ACM, September 2007.
- [162] N. Ikizler-Cinbis, R. G. Cinbis, and S. Sclaroff, “Learning actions from the web,” in *Proceedings of the IEEE International Conference on Computer Vision*, (Kyoto, Japan), pp. 995–1002, IEEE Computer Society, September 2009.
- [163] N. Ikizler-Cinbis and S. Sclaroff, “Object, scene and actions: Combining multiple features for human action recognition,” in *Proceedings of the European Conference on Computer Vision*, (Heraklion, Crete, Greece), pp. 494–507, Springer-Verlag, September 2010.
- [164] J. Liu, J. Luo, and M. Shah, “Recognizing realistic actions from videos “in the wild”,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (Miami, Florida, United States), pp. 1996–2003, IEEE Computer Society, June 2009.
- [165] A. C. Loui, J. Luo, S.-F. Chang, D. Ellis, W. Jiang, L. Kennedy, K. Lee, and A. Yanagawa, “Kodak’s consumer video benchmark data set: Concept definition and annotation,” in *Proceedings of the ACM International Workshop on Multimedia Information Retrieval*, (Augsburg, Bavaria, Germany), pp. 245–254, ACM, September 2007.

- [166] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (Anchorage, Alaska, United States), pp. 1–8, IEEE Computer Society, June 2008.
- [167] J. C. Niebles, H. Wang, and L. Fei-Fei, “Unsupervised learning of human action categories using spatial-temporal words,” *International Journal of Computer Vision*, vol. 79, pp. 299–318, September 2008.
- [168] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: A local svm approach,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (Cambridge, United Kingdom), pp. 32–36, IEEE Computer Society, August 2004.
- [169] K. Grauman and T. Darrell, “The pyramid match kernel: Discriminative classification with sets of image features,” in *Proceedings of the IEEE International Conference on Computer Vision*, (Beijing, China), pp. 1458–1465, IEEE Computer Society, October 2009.
- [170] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (New York, New York, United States), pp. 2169–2178, IEEE Computer Society, June 2006.
- [171] D. Xu, T. J. Cham, S. Yan, L. Duan, and S.-F. Chang, “Near duplicate identification with spatially aligned pyramid matching,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, pp. 1068–1079, August 2010.
- [172] P. Peursum, S. Venkatesh, G. A. W. West, and H. H. Bui, “Proceedings of the iee international conference on pervasive computing and communications,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (Dallas-Fort Worth, Texas, United States), pp. 399–406, IEEE Computer Society, March 2003.

- [173] M. Brand, N. Oliver, and A. Pentland, “Coupled hidden markov models for complex action recognition,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (San Juan, Puerto Rico), pp. 994–999, IEEE Computer Society, June 1997.
- [174] N. Oliver, B. Rosario, and A. Pentland, “A bayesian computer vision system for modeling human interactions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 831–843, August 2000.
- [175] I. Laptev and T. Lindeberg, “Space-time interest points,” in *Proceedings of the IEEE International Conference on Computer Vision*, (Nice, France), pp. 432–439, IEEE Computer Society, October 2003.
- [176] Y. Ke, R. Sukthankar, and M. Hebert, “Efficient visual event detection using volumetric features,” in *Proceedings of the IEEE International Conference on Computer Vision*, (Beijing, China), pp. 166–173, IEEE Computer Society, October 2005.
- [177] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in *Proceedings of the International Conference on Computer Communications and Networks*, (San Diego, California, United States), pp. 65–72, IEEE Computer Society, October 2005.
- [178] A. Kovashka and K. Grauman, “Learning a hierarchy of discriminative space-time neighborhood features for human action recognition,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (San Francisco, California, United States), pp. 2046–2053, IEEE Computer Society, June 2010.
- [179] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 2247–2253, December 2007.
- [180] Z. Lin, Z. Jiang, and L. S. Davis, “Recognizing actions by shape-motion prototype trees,” in *Proceedings of the IEEE International Conference on Computer Vision*, (Kyoto, Japan), pp. 444–451, IEEE Computer Society, September 2009.

- [181] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, “Machine recognition of human activities: A survey,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 1473–1488, November 2008.
- [182] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, “Automatic annotation of human actions in video,” in *Proceedings of the IEEE International Conference on Computer Vision*, (Kyoto, Japan), pp. 1491–1498, IEEE Computer Society, September 2009.
- [183] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–110, November 2004.
- [184] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metrix for image retrieval,” *International Journal of Computer Vision*, vol. 40, pp. 99–121, November 2000.
- [185] P. A. Jensen and J. F. Bard, *Operations Research Models and Methods*. John Wiley and Sons, 2003.
- [186] A. J. Smola, T. T. Frieß, and B. Schölkopf, “Semiparametric support vector and linear programming machines,” in *Advances in Neural Information Processing Systems 11*, pp. 585–591, MIT Press, 1999.
- [187] J. T. Kwok and I. W. Tsang, “Learning with idealized kernels,” in *Proceedings of International Conference on Machine Learning*, (Washington, DC, United States), pp. 400–407, AAAI Press, August 2003.
- [188] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. Huang, “Action detection in complex scenes with spatial and temporal ambiguities,” in *Proceedings of the IEEE International Conference on Computer Vision*, (Kyoto, Japan), pp. 128–135, IEEE Computer Society, September 2009.
- [189] O. Pele and M. Werman, “Fast and robust earth mover’s distances,” in *Proceedings of the IEEE International Conference on Computer Vision*, (Kyoto, Japan), pp. 460–467, IEEE Computer Society, September 2009.

- [190] A. Torralba, R. Fergus, and W. T. Freeman, “80 million tiny images: A large data set for nonparametric object and scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1958–1970, November 2008.
- [191] J. Hays and A. A. Efros, “Scene completion using millions of photographs,” *ACM Transactions on Graphics (SIGGRAPH 2007)*, vol. 26, July 2007.
- [192] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples,” *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, December 2006.
- [193] T. Joachims, “Transductive inference for text classification using support vector machines,” in *Proceedings of International Conference on Machine Learning*, (Bled, Slovenia), pp. 200–209, Morgan Kaufmann Publishers Inc., June 1999.
- [194] 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*, pp. 1433–1440, MIT Press, 2008.
- [195] S. Bickel, C. Sawade, and T. Scheffer, “Transfer learning by distribution matching for targeted advertising,” in *Advances in Neural Information Processing Systems 21*, pp. 145–152, MIT Press, 2009.
- [196] M. T. Rosenstein, Z. Marx, and L. P. Kaelbling, “To transfer or not to transfer,” in *NIPS 2005 Workshop on Inductive Transfer: 10 Years Later*, (Whistler, British Columbia, Canada), MIT Press, December 2005.
- [197] T. Evgeniou, C. Micchelli, and M. Pontil, “Learning multiple tasks with kernel methods,” *Journal of Machine Learning Research*, vol. 6, pp. 615–637, April 2005.
- [198] T. Kato, H. Kashima, M. Sugiyama, and K. Asai, “Multi-task learning via conic programming,” in *Advances in Neural Information Processing Systems 20*, pp. 737–744, MIT Press, 2008.

- [199] D. Sheldon, “Graphical multi-task learning,” in *NIPS 2008 Workshop on Structured Input - Structured Output*, (Whistler, British Columbia, Canada), MIT Press, December 2008.
- [200] F. H. Sinz, O. Chapelle, A. Agarwal, and B. Schölkopf, “An analysis of inference with the universum,” in *Advances in Neural Information Processing Systems 20*, pp. 1369–1376, MIT Press, 2008.
- [201] J. Weston, R. Collobert, F. Sinz, L. Bottou, and V. Vapnik, “Inference with the universum,” in *Proceedings of International Conference on Machine Learning*, (Pittsburgh, Pennsylvania, United States), pp. 1009–1016, ACM, June 2006.
- [202] 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,” *Machine Learning*, vol. 54, pp. 5–32, January 2004.
- [203] V. Sindhwani, P. Niyogi, and M. Belkin, “Beyond the point cloud: from transductive to semi-supervised learning,” in *Proceedings of International Conference on Machine Learning*, (Bonn, Germany), pp. 824–831, ACM, June 2005.
- [204] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and Computing*, vol. 14, pp. 199–222, August 2004.
- [205] I. W. Tsang and J. T. Kwok, “Large-scale sparsified manifold regularization,” in *Advances in Neural Information Processing Systems 19*, pp. 1401–1408, MIT Press, 2007.
- [206] T. D. Bie and N. Cristianini, *Semi-supervised learning*, ch. Semi-Supervised Learning Using Semi-Definite Programming, pp. 119–135. MIT Press, 2006.
- [207] V. Vapnik, *Estimation of Dependences Based on Empirical Data*. Springer Verlag, 2nd ed., 2006.
- [208] J. C. Platt, *Advances in Large Margin Classifiers*, ch. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, pp. 61–74. MIT Press, 1999.

- [209] S. Ben-david and R. Schuller, “Exploiting task relatedness for multiple task learning,” in *Proceedings of the Annual Conference on Computational Learning Theory*, (Washington, D.C., United States), pp. 567–580, Springer, August 2003.
- [210] B. Bakker and T. Heskes, “Task clustering and gating for bayesian multitask learning,” *Journal of Machine Learning Research*, vol. 4, pp. 83–99, December 2005.
- [211] A. Argyriou, A. Maurer, and M. Pontil, “An algorithm for transfer learning in a heterogeneous environment,” in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, (Antwerp, Belgium), pp. 71–85, Springer-Verlag, September 2008.
- [212] E. W. Xiang, S. J. Pan, W. Pan, J. Su, and Q. Yang, “Source-selection-free transfer learning,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, (Barcelona, Spain), pp. 2355–2360, IJCAI/AAAI, July 2011.
- [213] Y. Zhu, Y. Chen, Z. Lu, S. J. Pan, G.-R. Xue, Y. Yu, and Q. Yang, “Heterogeneous transfer learning for image classification,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, (San Francisco, California, United States), AAAI Press, August 2011.
- [214] L. Reyzin and R. E. Schapire, “How boosting the margin can also boost classifier complexity,” in *Proceedings of International Conference on Machine Learning*, (Pittsburgh, Pennsylvania, United States), pp. 753–760, ACM, June 2006.
- [215] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable distributions,” in *Proceedings of the Annual Symposium on Computational Geometry*, (Brooklyn, New York, United States), pp. 253–262, ACM, June 2004.
- [216] Y. Weiss, A. Torralba, and R. Fergus, “Spectral hashing,” in *Advances in Neural Information Processing Systems 21*, pp. 1753–1760, MIT Press, 2009.
- [217] J. Yin, Q. Yang, and L. Ni, “Adaptive temporal radio maps for indoor location estimation,” in *Proceedings of the IEEE International Conference on Pervasive*

- Computing and Communications*, (Kauai Island, Hawaii, United States), pp. 85–94, IEEE Computer Society, March 2005.
- [218] M. Rohrbach, M. Stark, and B. Schiele, “Evaluating knowledge transfer and zero-shot learning in a large-scale setting,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (Colorado Springs, United States), pp. 1641–1648, IEEE Computer Society, June 2011.
- [219] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, June 2008.
- [220] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

Appendix A

Appendix

A.1 Derivation of the dual problem of $J(\mathbf{d})$ in DTMKL- f in (3.15)

The primal form of DTMKL- f in (3.15) can be written as:

$$\min_{\mathbf{d} \in \mathcal{D}} \frac{1}{2} \mathbf{d}' \mathbf{p} \mathbf{p}' \mathbf{d} + \theta J(\mathbf{d}),$$

where

$$\begin{aligned} J(\mathbf{d}) = & \min_{\substack{\mathbf{w}_m, b, \xi_i, \\ \xi_i^*, \mathbf{f}_l^{T,m}, \mathbf{f}_u^{T,m}}} \frac{1}{2} \sum_{m=1}^M d_m \|\mathbf{w}_m\|^2 + C \sum_{i=1}^{n+n_u} (\xi_i + \xi_i^*) \\ & + \frac{\zeta}{2} \left(\sum_{m=1}^M \left\| \mathbf{f}_l^{T,m} - \mathbf{y} \right\|^2 + \lambda \sum_{m=1}^M \left\| \mathbf{f}_u^{T,m} - \mathbf{f}_u^{B,m} \right\|^2 \right), \\ \text{s.t.} \quad & \sum_{m=1}^M d_m \mathbf{w}_m' \phi_m(\mathbf{x}_i) + b - \sum_{m=1}^M d_m f_i^{T,m} \leq \epsilon + \xi_i, \xi_i \geq 0, \\ & \sum_{m=1}^M d_m f_i^{T,m} - \sum_{m=1}^M d_m \mathbf{w}_m' \phi_m(\mathbf{x}_i) - b \leq \epsilon + \xi_i^*, \xi_i^* \geq 0. \quad (\text{A.1}) \end{aligned}$$

The Lagrangian of (A.1) can be obtained by introducing the Lagrangian multipliers α_i , β_i , α_i^* , and β_i^* for each constraint in (A.1). Denote $\mathbf{f}^{T,m} = [\mathbf{f}_l^{T,m'}, \mathbf{f}_u^{T,m'}]'$, $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_{n+n_u}]'$, $\boldsymbol{\beta} = [\beta_1, \dots, \beta_{n+n_u}]'$, $\boldsymbol{\alpha}^* = [\alpha_1^*, \dots, \alpha_{n+n_u}^*]'$ and $\boldsymbol{\beta}^* = [\beta_1^*, \dots, \beta_{n+n_u}^*]'$. Setting the derivatives of the Lagrangian with respect to the primal variables ($\mathbf{w}_m, b, \mathbf{f}^{T,m}, \xi_i$

and ξ_i^*) to zeros, we have

$$\begin{aligned} \mathbf{w}_m &= -\Phi_m(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*), \\ \mathbf{f}^{T,m} &= \begin{bmatrix} \mathbf{I}_n \\ \frac{1}{\lambda} \mathbf{I}_{n_u} \end{bmatrix} \frac{d_m(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)}{\zeta} + \begin{bmatrix} \mathbf{y} \\ \mathbf{f}_u^{B,m} \end{bmatrix}, \end{aligned}$$

$\boldsymbol{\alpha}' \mathbf{1}_{n+n_u} = \boldsymbol{\alpha}^{*'} \mathbf{1}_{n+n_u}$, $\alpha_i + \beta_i = C$ and $\alpha_i^* + \beta_i^* = C$, where $\Phi_m = [\phi_m(\mathbf{x}_1), \dots, \phi_m(\mathbf{x}_{n+n_u})]$. Substituting the above back into the Lagrangian, we can arrive at the dual of (A.1) as follows:

$$J(\mathbf{d}) = \max_{(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) \in \mathcal{A}} -\frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)' \tilde{\mathbf{K}}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) - \tilde{\mathbf{y}}'(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) - \epsilon \mathbf{1}'_{n+n_u}(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*),$$

where $\mathcal{A} = \{(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) | \boldsymbol{\alpha}' \mathbf{1}_{n+n_u} = \boldsymbol{\alpha}^{*'} \mathbf{1}_{n+n_u}, \mathbf{0}_{n+n_u} \leq \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \leq C \mathbf{1}_{n+n_u}\}$ is the feasible set of the dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$, $\tilde{\mathbf{K}} = \sum_{m=1}^M d_m \tilde{\mathbf{K}}_m = \sum_{m=1}^M d_m \mathbf{K}_m + \frac{1}{\zeta} \sum_{m=1}^M d_m^2 \begin{bmatrix} \mathbf{I}_n & \\ & \frac{1}{\lambda} \mathbf{I}_{n_u} \end{bmatrix}$ and $\tilde{\mathbf{y}} = \sum_{m=1}^M d_m \tilde{\mathbf{y}}_m = \begin{bmatrix} \mathbf{y} \\ \sum_{m=1}^M d_m \mathbf{f}_u^{B,m} \end{bmatrix}$.

A.2 Proof of Theorem 5.3

Proof: Analogous to LS-SVM [202], let us define $\xi_i = f_i^T - y_i^T$. Assuming $\Omega(f^T) = \frac{1}{2} \|\mathbf{w}\|^2$, $\mathbf{f}^T(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x})$, we can rewrite (5.9) as follows:

$$\min_{\mathbf{w}, \xi_i} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \frac{\lambda_D}{2} \sum_{s=1}^P \gamma_s \|\Phi'_u \mathbf{w} - \mathbf{f}_u^s\|^2 + \frac{\lambda_L}{2} \sum_{i=1}^{n_l} \xi_i^2, \quad (\text{A.2})$$

$$\text{s.t.} \quad \xi_i = \mathbf{w}'\phi(\mathbf{x}_i) - y_i^T, \quad i = 1, \dots, n_l, \quad (\text{A.3})$$

where $\Phi_u = [\phi(\mathbf{x}_{n_l+1}^T), \dots, \phi(\mathbf{x}_{n_T}^T)]$.

The Lagrangian of (A.2) can be obtained as follows by introducing the dual variable α_i for each constraint in (A.3):

$$L = \frac{1}{2} \|\mathbf{w}\|^2 + \frac{\lambda_D}{2} \sum_{s=1}^P \gamma_s \|\Phi'_u \mathbf{w} - \mathbf{f}_u^s\|^2 + \frac{\lambda_L}{2} \sum_{i=1}^{n_l} \xi_i^2 + \sum_{i=1}^{n_l} \alpha_i (\xi_i - \mathbf{w}'\phi(\mathbf{x}_i) + y_i^T). \quad (\text{A.4})$$

By taking derivatives of the Lagrangian (A.4) with respect to $\mathbf{w}, \xi_i$ and setting them zero, respectively, we have:

$$\xi_i = -\frac{\alpha_i}{\lambda_L} \quad \text{and} \quad \mathbf{w} = \left(\mathbf{I}_F + (\lambda_D \sum_{s=1}^P \gamma_s) \Phi_u \Phi'_u \right)^{-1} \left(\lambda_D \Phi_u \sum_{s=1}^P \gamma_s \mathbf{f}_u^s + \Phi_l \boldsymbol{\alpha}_l \right), \quad (\text{A.5})$$

where $\Phi_l = [\phi(\mathbf{x}_1^T), \dots, \phi(\mathbf{x}_{n_l}^T)]$, $\alpha_l = [\alpha_1, \dots, \alpha_{n_l}]'$ and F is the dimension of $\phi(\mathbf{x})$ in the nonlinear feature space. With the Woodbury identity [220]: $(\mathbf{A} + \mathbf{B}\mathbf{C})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{I} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1}$, (A.5) can be rewritten as:

$$\mathbf{w} = (\mathbf{I}_F - \Phi_u (\mathbf{I}_{n_u} + \mathbf{M}\mathbf{K}_u)^{-1} \mathbf{M}\Phi_u') \left(\lambda_D \Phi_u \sum_{s=1}^P \gamma_s \mathbf{f}_u^s + \Phi_l \alpha_l \right), \quad (\text{A.6})$$

where $\mathbf{M} = \lambda_D \sum_{s=1}^P \gamma_s \mathbf{I}_{n_u}$ and $\mathbf{K}_u = \Phi_u' \Phi_u = [k(\mathbf{x}_i^T, \mathbf{x}_j^T)]_{n_u \times n_u}$.

Similarly as in [202], the dual variable α_l can be obtained by solving the dual of (A.2). With the dual variable α_l and $\mathbf{w}$ in (A.6), we finally arrive at the target decision function f^T as follows:

$$f^T(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) = \lambda_D \sum_{s=1}^P \gamma_s \sum_{i=n_l+1}^{n_T} f^s(\mathbf{x}_i^T) \tilde{k}(\mathbf{x}_i^T, \mathbf{x}) + \sum_{i=1}^{n_l} \alpha_i^T \tilde{k}(\mathbf{x}_i^T, \mathbf{x}),$$

where $\tilde{k}(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) - \mathbf{k}_{\mathbf{x}_i}' (\mathbf{I}_{n_u} + \mathbf{M}\mathbf{K}_u)^{-1} \mathbf{M}\mathbf{k}_{\mathbf{x}_j}$ and $\mathbf{k}_{\mathbf{x}} = [k(\mathbf{x}_{n_l+1}^T, \mathbf{x}), \dots, k(\mathbf{x}_{n_T}^T, \mathbf{x})]'$.

A.3 Proof of Theorem 5.4

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

$$\xi_{s,i}, \xi_{s,i}^* \geq 0 \quad (\text{A.7})$$

$$\alpha_{s,i} (\epsilon_s + \xi_{s,i} - \mathbf{w}'\phi(\mathbf{x}_i^s) - b + f^T(\mathbf{x}_i^s)) = 0 \quad (\text{A.8})$$

$$\alpha_{s,i}^* (\epsilon_s + \xi_{s,i}^* + \mathbf{w}'\phi(\mathbf{x}_i^s) + b - f^T(\mathbf{x}_i^s)) = 0 \quad (\text{A.9})$$

Assuming $\epsilon_s > \max_{i=1, \dots, n_s} |\mathbf{w}'\phi(\mathbf{x}_i^s) + b - f^T(\mathbf{x}_i^s)|$, 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 (A.7), (A.8) and (A.9), it is easy to verify $\alpha_{s,i}, \alpha_{s,i}^* = 0$ for every training instance from the source domains.

Therefore, if the inequality $\epsilon_s > \max_{i=1, \dots, n_s} |\mathbf{w}'\phi(\mathbf{x}_i^s) + b - f^T(\mathbf{x}_i^s)|$ holds for each source domain, the optimization problem (5.22) (*resp.* the dual form (5.29)) of UniverDAM can be degraded into the optimization problem (5.13) (*resp.* the dual form (5.18)), namely, UniverDAM reduces to FastDAM.

A.4 Transductive Inference with the Universum

The general setting of transductive inference is as follows: Given a set of l training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$, where $y_i \in \{-1, 1\}$ is the label of $\mathbf{x}_i$, and a set of u test data $\mathbf{x}_i|_{i=l+1}^{l+u}$, the goal is to find the classification of the test data, which classifies the test data with the smallest errors.

As suggested in [26, 207], one can find such a classification for the test data based on the Structural Risk Minimization (SRM) principle. Specifically, prior to the given working set (*i.e.*, the training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$ and the test set $\{\mathbf{x}_i|_{i=l+1}^{l+u}\}$), one can construct the structure on the finite number of equivalence classes¹ $F_1, \dots, F_N$ that are the result of factorization of the given set of functions over the given $l + u$ data vectors. Such a structure is defined as follows:

$$S_1^* \subset \dots \subset S_K^* \subset S^* = \{F_1, \dots, F_N\},$$

where S_k^* contains N_k equivalence classes of functions from $f(\mathbf{x}, \alpha)$, $\alpha \in \Lambda$. One then chooses the appropriate S_k^* by minimizing the upper bound on the test error of a classification model. As shown in [26], finding such a subset S_k^* is equivalent to *maximizing* the number of contradictions on a new collection of unlabeled data $\{\mathbf{x}_i^*|_{i=1}^u\}$ that belong to neither of the two classes or even come from a different data distribution. This additional data collection is called as *Universum*. Universum is considered as prior knowledge to about a relationship between the working set and a set of possible problems, and it thus help find the optimal margin and enhance the generalization ability of the learned classifier. Taking digit classification for example, the strokes of different hand-written digits can help for classifying a certain digit of interest.

The number of contradictions on the Universum is defined as the number of the Universum data points $\mathbf{x}_i^*$ each of which contradicts the equivalence class F_r if in F_r there are functions that classify this vector as belonging to the first category as well as functions that classify $\mathbf{x}_i^*$ as belonging to the second category (*i.e.*, there exist two functions $f(\mathbf{x}_i^*, \alpha_1)$ and $f(\mathbf{x}_i^*, \alpha_2)$ in F_r such that $f(\mathbf{x}_i^*, \alpha_1) < 0$ and $f(\mathbf{x}_i^*, \alpha_2) > 0$) [26,

¹A subset of functions that classify data vectors in the same way (*e.g.*, for binary classification, the classification results of data vectors are all 1 or -1, obtained by using the functions from the same equivalent class) belong to the same equivalence class.

207]. Based on this definition, one wishes to penalize the output of the Universum data as close to zero as possible, as it has a higher chance for the Universum data to contradict equivalence classes. Therefore, the Universum regularizer $\Omega_{\mathcal{U}}(f) = \sum_{i=1}^n |f(\mathbf{x}_i^*)|^2$ was introduced for this purpose, and it has been successfully incorporated in the structural risk functional of SVM [26, 201, 207].

In Chapter 5, we adopt the Universum regularizer for multiple source domain adaptation (see Chapter 5). Specifically, in our proposed method UniverDAM (see Section 5.3.4), we treat the available source data as the Universum in order to further enhance the generalization ability of the target classifier.

Publication

Journal Publications

- **Lixin Duan**, Dong Xu, Ivor W. Tsang, and Jiebo Luo, “Visual Event Recognition in Videos by Learning from Web Data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, accepted, 2011.
- **Lixin Duan**, Ivor W. Tsang, and Dong Xu, “Domain Adaptation from Multiple Sources: A Domain-Dependent Regularization Approach,” *IEEE Transactions on Neural Networks*, accepted, 2011.
- **Lixin Duan**, Wen Li, Ivor W. Tsang, and Dong Xu, “Improving Web Image Search by Bag-Based Re-Ranking,” *IEEE Transactions on Image Processing*, accepted, 2011.
- **Lixin Duan**, Ivor W. Tsang, and Dong Xu, “Domain Transfer Multiple Kernel Learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, accepted, 2011.
- Dong Xu, Tat Jen Cham, Shuicheng Yan, **Lixin Duan**, and Shih-Fu Chang, “Near Duplicate Identification with Spatially Aligned Pyramid Matching,” *IEEE Transactions on Circuits Systems for Video Technology*, vol. 20, no. 8, pp. 1068–1079, August 2010.

Conference Publications

- Wen Li, **Lixin Duan**, Ivor W. Tsang, and Dong Xu, “Batch Mode Adaptive Multiple Instance Learning for Computer Vision Tasks,” *to appear in Proceedings*

of *IEEE International Conference on Computer Vision and Pattern Recognition*, Providence, Rhode Island, United States, June 2012.

- **Lixin Duan**, Dong Xu, and Shih-Fu Chang, “Exploiting Web Images for Event Recognition in Consumer Videos: A Multiple Source Domain Adaptation Approach,” *to appear in Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, Providence, Rhode Island, United States, June 2012.
- Wen Li, **Lixin Duan**, Dong Xu, and Ivor W. Tsang, “Text-Based Image Retrieval using Progressive Multi-Instance Learning,” *in Proceedings of IEEE International Conference on Computer Vision*, pp. 2049–2055, Barcelona, Spain, November 2011.
- Xinxiao Wu, Dong Xu, **Lixin Duan**, and Jiebo Luo, “Action Recognition using Context and Appearance Distribution Features,” *in Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 489–496, Colorado Springs, United States, June 2011.
- **Lixin Duan**, Ivor W. Tsang, Dong Xu, and Jiebo Luo, “Visual Event Recognition in Videos by Learning from Web Data,” *in Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1959–1966, San Francisco, California, United States, June 2010.
- **Lixin Duan**, Ivor W. Tsang, Dong Xu, and Tat-Seng Chua, “Domain Adaptation from Multiple Sources via Auxiliary Classifiers,” *in Proceedings of International Conference on Machine Learning*, pp. 289–296, Montreal, Canada, June 2009.
- **Lixin Duan**, Ivor W. Tsang, Dong Xu, and Stephen J. Maybank, “Domain Transfer SVM for Video Concept Detection,” *in Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, Miami, Florida, United States, pp. 1375–1381, June 2009.