Visual recognition by learning from web data

Li, Wen

2014

Li, W. (2014). Visual recognition by learning from web data. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/62186>

<https://doi.org/10.32657/10356/62186>

VISUAL RECOGNITION BY LEARNING FROM WEB DATA

A thesis submitted to
The School of Computer Engineering
The Nanyang Technological University

by

Wen Li

in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

December 2014

Abstract

With the rapid development of digital cameras, we have witnessed an explosive growth of digital images. Every day, a tremendous amount of images together with rich contextual information (*e.g.*, tags, categories and captions) are posted to Internet. There is an increasing interest in exploiting those web images for building intelligent visual recognition systems. While some works have been proposed to collect large scale image datasets by crawling web images from Internet, considerable human efforts are still required to annotate those images to train classifiers for visual recognition.

In this thesis, we propose to develop novel learning algorithms for visual recognition by learning from web data, in which we aim to use as less as possible human efforts for annotating the training data. First, considering that web images are usually associated with noisy surrounding textual descriptions, we treat the words in the surrounding text as weak labels and formulate the task of learning from web data as a multi-instance learning (MIL) problem. By observing the relevant images usually contain many true positive images, we generalize the traditional MIL constraints on positive bags to that each positive bag contains at least a portion of positive instances. To effectively exploit such constraints on positive bags, we develop a new MIL algorithm called MIL with constrained positive bags (MIL-CPB) for web image retrieval. Observing that the constraints are not always satisfied in the MIL-CPB, a progressive scheme is proposed to further improve the retrieval performance, in which we iteratively partition the top-ranked training web images from the current MIL-CPB classifier to construct more confident positive bags and then use these new bags as training data to learn the subsequent MIL-CPB classifiers.

Second, when the web training data is represented with multiple views of features, we further propose a co-labeling approach to improve the classifiers learnt from web data by using multiple views of features. We model the learning problem on each view as a

weakly labeled learning problem, and use the predicted training labels from the classifier trained on one view to help the classifier on another view. Our co-labeling approach not only can handle the traditional multi-view semi-supervised learning problem, but also can be applied to other multi-view weakly labeled learning problem such as multi-view MIL.

Finally, we observe that there are intrinsic differences between the crawled web training data and the testing images in our daily lives, which is also known as the domain adaptation problem. Particularly, we study the heterogeneous domain adaptation problem, in which the samples in source and target domains are with different feature representations. We build upon the recent Heterogeneous Feature Augmentation (HFA) method, and propose a convex reformulation of HFA, which can guarantee the global optimal solution. We further extend the HFA method to semi-supervised HFA (SHFA), in which we improve the learnt classifiers by exploiting the additional unlabeled data from the target domain. For all our proposed approaches, we conduct extensive experiments on publicly available datasets to demonstrate their effectiveness.

Acknowledgments

First and foremost I would like to express my grateful thanks to my supervisor, Dr. Dong Xu, for his valuable guidance and support during the past four years of my Ph.D. study. Under his supervision, I have learned not only the cutting-edge research on machine learning and computer vision, but also his inspiring and down-to-earth working attitude. From him, I have also learned the way to write qualified research papers and the way to become an independent researcher.

I would also like to specially thank Dr. Ivor Wai-Hung Tsang and Dr. Lixin Duan, who offer me valuable suggestions and inspirations in our collaborated works. With the useful discussions with them, I work out many difficult problems and am also encouraged to explore unknowns in the future.

I would like to thank all my friends and colleagues, Lin Chen, Huazhu Fu, Shenghua Gao, Zhiming Hu, Yi Huang, Meina Kan, Feng Li, Jia Li, Haifeng Lu, Huiying Liu, Qi Mao, Li Niu, Hoang Anh Ng, Wei Shou, Mingkui Tan, Xiao Teng, Jiang Wang, Shikui Wei, Xinxiao Wu, Liu Xiang, Shijie Xiao, Zhiwei Xiong, Xinxing Xu, Yanwu Xu, Zheng Xu, Shengye Yan, Ming Yang, Zinan Zeng, Chi Zhang, Wenjie Zhang, Yuanqing Zheng, and Tianyi Zhou, for their friendships and insightful discussions. Thanks to them, I live a wonderful life here.

Last but not least, I sincerely thank my family from the bottom of my heart for everything.

To my parents.

Contents

Abstract	i
Acknowledgments	iii
List of Figures	xi
List of Tables	xiii
Notations and Abbreviations	xv
1 Introduction	1
1.1 Learning from Weakly Labeled Data	3
1.2 Learning from Multi-view Representations	5
1.3 Learning from Heterogeneous Domain	6
1.4 Connections	7
1.5 Thesis Outline	8
1.6 Notations	10
2 Literature Review	13
2.1 Weakly labeled learning	13
2.1.1 Supervised Learning and Support Vector Machines	13
2.1.2 Multiple Instance Learning	15
2.1.3 Weakly Labeled Learning	20
2.2 Multi-view Learning	22
2.3 Domain Adaptation	23
2.3.1 Homogeneous Domain Adaptation	24
2.3.2 Heterogeneous Domain Adaptation	26
2.4 Summary	27

3	Progressive Multiple Instance Learning for Image Retrieval	29
3.1	Introduction	29
3.2	Related Work	31
3.3	Multi-Instance Learning for Image Retrieval	32
3.3.1	MIL with constrained positive bags (MIL-CPB)	34
3.3.2	Progressive MIL-CPB (PMIL-CPB)	37
3.4	Experiments	38
3.4.1	Image retrieval on the NUS-WIDE data set	38
3.4.2	Image re-ranking on the Google data set	43
3.5	Summary	44
4	Co-Labeling for Multiple View Weakly Labeled Learning Problems	47
4.1	Co-Labeling for Multi-view Weakly Labeled Learning	48
4.1.1	Background on Co-training	48
4.1.2	Background on Weakly Labeled Learning	48
4.1.3	Multi-view Weakly Labeled Learning	50
4.2	Generate the Pseudo-label Vector Set	52
4.2.1	Handling General Weakly Labeled Constraints	52
4.2.2	Handling the Bias of Weakly Labeled Learning	53
4.2.3	Combining the Pseudo-label Vector Set from Previous Iterations	53
4.2.4	Summary and Discussion	54
4.3	The Implementation of Co-Labeling	55
4.3.1	The Formulation	55
4.3.2	The MKL solution	55
4.3.3	The Algorithm	57
4.4	Generalization Bound Analysis	58
4.5	Related Work	60
4.6	Experiments	61
4.6.1	Multi-View Multiple Instance Learning	61
4.6.2	Multi-View Semi-Supervised Learning	63
4.7	Summary	68

5	Semi-supervised Heterogeneous Feature Augmentation	71
5.1	Introduction	71
5.2	Heterogeneous Feature Augmentation	74
5.2.1	A Review of Heterogeneous Feature Augmentation	74
5.2.2	A Convex Formulation	77
5.2.3	Solution	79
5.2.4	Convergence Analysis	80
5.3	Semi-supervised Heterogeneous Feature Augmentation	81
5.3.1	Formulation	82
5.3.2	Convex Relaxation	83
5.3.3	Solution	84
5.3.4	ℓ_p -MKL Extension	87
5.4	Experiments	88
5.4.1	Setup	88
5.4.2	Classification Results	92
5.4.3	Augmented Features <i>v.s.</i> Common Features	98
5.4.4	Performance Variations using Different Parameters	98
5.4.5	Time Analysis	101
5.5	Summary	101
6	Conclusions and Future Work	103
6.1	Conclusions	103
6.2	Future Work	104
	References	107
A	Appendix	121
A.1	Proof of Theorem 4.1	121
A.1.1	Generalization Bound for Weakly Labeled Learning	121
A.1.2	Generalization Bound for Co-Labeling	125
A.2	Proof of Theorem 5.3	127
A.3	Proof of Proposition 1	130
	Publication	133

List of Figures

1.1	An illustration on the rest of this thesis.	9
2.1	An illustration on Support Vector Machine classifier. Solid circles are positive samples, while hollow circles are negative samples. The samples with red circles are support vectors.	14
3.1	Flowchart of the proposed approach.	33
3.2	MAPs of all methods over 81 concepts on the NUS-WIDE data set with respect to different numbers (<i>i.e.</i> , N) of positive/negative bags.	40
3.3	Average purities of Init_Ranking and PMIL-CPB over 81 concepts on the NUS-WIDE data set using different numbers (<i>i.e.</i> , m) of positively predicted instances.	42
4.1	Comparison between co-training and our co-labeling.	49
4.2	The objective values with respect to the number of iterations on two views.	67
5.1	The samples from different domains are represented by different features, where red crosses, blue strips, orange triangles and green circles denote source positive samples, source negative samples, target positive samples and target negative samples, respectively. By using two projection matrices $\mathbf{P}$ and $\mathbf{Q}$, the heterogenous samples from two domains are transformed into an augmented feature space.	73
5.2	Classification accuracies of all methods with respect to different number of target training samples per class (<i>i.e.</i> , $m = 5, 10, 15$ and 20) on the Reuters multilingual dataset. Spanish is considered as the target domain, and in each subfigure the results are obtained by using one language as the source domain.	97

5.3	The performances of our SHFA using different parameters on the Reuters multilingual dataset.	99
-----	--	----

List of Tables

3.1	MAPs (%) of all methods over 81 concepts on the NUS-WIDE data set. Each result in the table is the best among all results obtained by using different numbers (<i>i.e.</i> , N) of positive/negative training bags.	40
3.2	Mean precisions (%) at 15% recall over 7 categories on the Google data set. Each result in the table is the best among all results obtained by using different numbers (<i>i.e.</i> , N) of positive/negative training bags.	44
3.3	Per-category precisions (%) at 15% recall for 7 categories (<i>i.e.</i> , “Airplane”, “Cars rear”, “Face”, “Guitar”, “Leopard”, “Motorbike” and “Wrist watch”) on the Google data set. For better presentation, we denote the name of each category by using its first letter in this table.	44
4.1	Mean Average Precisions (MAPs) over 81 concepts from different methods on the NUS-WIDE data set.	62
4.2	Summarization of the data sets used in SSL. $d1$ and $d2$ are the feature dimensions of two views. $\#c$, $\#l$, $\#u$ and $\#t$ are the numbers of classes, labeled data, unlabeled data and test data, respectively.	64
4.3	Performances of different methods on three data sets. The best results are denoted in boldface. MAPs and standard deviations are reported on BBC&BBCSport. $\uparrow$ denotes the result is significantly better than the others according to the t-test with a significance level at 0.05. PRBEPs are reported on WebKB. The results of SVM, TSVM and Co-LapSVM are from Table 1 in [1]. Since the standard deviations of these methods are not available, we only report the average PRBEPs on WebKB.	65
4.4	Performance of different versions of co-labeling on BBC-Sports.	67

4.5	Average training times of different multi-view SSL methods per class per split on three data sets (in seconds).	68
5.1	Summarization of the object dataset including 31 categories.	89
5.2	Summarization of the Reuters multilingual dataset including 6 classes. . .	90
5.3	Summarization of the Cross-Lingual Sentiment dataset including 3 categories and 2 classes.	91
5.4	Means and standard deviations of classification accuracies (%) of all methods on the object dataset by using SURF features. Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.	93
5.5	Means and standard deviations of classification accuracies (%) of all methods on the object dataset by using DeCAF features. Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.	94
5.6	Means and standard deviations of classification accuracies (%) of all methods on the Reuters multilingual dataset by using 10 and 20 labeled training samples per class from the target domain Spanish . Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.	95
5.7	Means and standard deviations of classification accuracies (%) of all methods on the Cross-Lingual Sentiment dataset by using 100 labeled training samples from the target domain. Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.	98
5.8	Means and standard deviations of classification accuracies (%) of our SHFA ($p = 1.5$) and SHFA_commFeat ($p = 1.5$) on the Reuters multilingual dataset.	100
5.9	The average per class training time (in seconds) comparisons of all methods on the Cross-Lingual Sentiment dataset.	100

List of Notations and Abbreviations

In this thesis, a lowercase character represents a scalar (*i.e.*, b) and a bold lowercase character denotes a vector (*i.e.*, $\mathbf{x}$). A matrix is denoted by a bold uppercase character (*i.e.*, $\mathbf{A}$). We list other frequently used notations and abbreviations as follows.

List of Notations:

$'$	Transpose of a vector or matrix
$\mathbf{0}_n$	$n \times 1$ vector of all zeros
$\mathbf{1}_n$	$n \times 1$ vector of all ones
$\mathbf{y} \circ \boldsymbol{\alpha}$	Element-wise product of two vectors $\mathbf{y}$ and $\boldsymbol{\alpha}$
$\mathbf{A} \circ \mathbf{B}$	Element-wise product of two matrices $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{K}$	Kernel matrix
$\mathbf{I}_n$	$n \times n$ identity matrix
$\text{diag}(\boldsymbol{\alpha})$	Diagonal matrix with $\boldsymbol{\alpha}$ as its diagonal elements
$\mathcal{H}$	Reproducing Kernel Hilbert Space (RKHS)
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	n -dimensional real value space
$\phi(\cdot)$	Nonlinear feature mapping function
$\ \mathbf{x}\ $	ℓ_2 norm of a vector $\mathbf{x}$
$\ \mathbf{x}\ _p$	ℓ_p norm of a vector $\mathbf{x}$
$\text{tr}(\mathbf{A})$	The trace-norm of a matrix $\mathbf{A}$
$\mathbf{A} \succeq 0$	$\mathbf{A}$ is a positive semi-definite matrix
$\mathbf{a} \geq \mathbf{b}$	$a_i \geq b_i$ for $i = 1, \dots, n$ with $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$

List of Abbreviations:

MIL	Multi-Instance Learning
SSL	Semi-Supervised Learning
MMC	Maximum Margin Clustering
HDA	Heterogenous Domain Adaptation
MMD	Maximum Mean Descrepancy
RKHS	Reproducing Kernel Hilbert Space
TBIR	Text Based Image Retrieval
SVM	Support Vector Machine
LapSVM	Laplacian SVM
TSVM	Transductive SVM
MKL	Multiple Kernel Learning
ℓ_1 MKL	ℓ_1 -norm Multiple Kernel Learning
ℓ_p MKL	ℓ_p -norm Multiple Kernel Learning
MIL-CPB	Multiple Instance Learning with Constrained Positive Bags
PMIL-CPB	Progressive MIL-CPB
HFA	Heterogeneous Feature Augmentation
SHFA	Semi-supervised HFA
PSD	Positive Semi-Definite
QP	Quadratic Programming
SDP	Semi-definite Programming
MIP	Mixed Integer Programming

Chapter 1

Introduction

With the advance of digital cameras, smart phones, and webcams, a large number of images and videos are captured by users every day. With the increasing number of images and videos, it becomes more and more important to build intelligent systems to help users manage their images and videos. In this context, image/video recognition, as one of the most fundamental computer vision problems, has attracted more and more attentions from both academics and industries in recent decades.

In computer vision and machine learning systems, data always serves an important role. Therefore, researchers have collected various datasets [2–10] to facilitate the design and evaluation of learning algorithms. With the increasing demands on building more powerful recognition systems, the number of labeled samples has increased from around one thousand (*e.g.*, COIL-20 dataset [2]) to millions nowadays (*e.g.*, the recently proposed ImageNet dataset [10] contains more than 14 millions of images). Meanwhile, the coverage of dataset (number of concepts) also increases from only twenty objects to more than twenty thousands of various real-world concepts including animals, vehicles, foods, *etc.* With the continuous release of larger datasets, researchers can develop algorithms for recognizing thousands of concepts of images at a certain accuracy. However, there is still a long way to build an intelligent system that can deal with the tremendous amount of semantic words in our daily lives.

Moreover, it is never a trivial task to build large scale image datasets. A great amount of human efforts have been dedicated to collect and annotate those images, which is both expensive and time consuming. In such a big data era, it is hard to imagine how many human efforts still need to be spent on building larger datasets to fulfill the increasing

demands on building intelligent recognition systems from users. Meanwhile, thanks to the development of network techniques, there are a huge number of photos and videos being uploaded to Internet every day. For example, according to the statistics on explored photos from *Flickr.com*, the number of uploaded public photos every day is up to 1.6 million in 2013, which does not yet include private photos uploaded by Flickr users¹. Similarly, about 100 hours of video are uploaded to *YouTube.com* very minute². More images and videos are uploaded to *Facebook.com*, *Instagram*, *Twitter*, *etc.*, too. Those web images and videos contain valuable information for researchers in both academics and industries to build more powerful and intelligent visual recognition systems.

In this thesis, we aim to develop new techniques for visual recognition by learning from web data. While we focus on the image recognition problem, all the issues discussed in this thesis can also be similarly applied to the problem of learning from other web data like web videos, because the video and image recognition tasks can be modeled as the same learning problem after representing each image or video as one feature vector. We summarize the advantages for learning from web data as follows: first, with a tremendous number of images on Internet, we can easily collect a sufficient number of training images; second, different from traditional works that rely on well-labeled datasets, learning from web data requires much less human efforts for annotating the training data.

However, learning from web data is a challenging task. Several major challenges are listed as follows,

- First, although web images are usually associated with surrounding textual descriptions, one can only extract noisy supervision information. Such problem is also known as the *weak supervision* problem, for which directly using those noisy supervision information can hardly obtain satisfactory recognition performance.
- Second, web data usually can be represented with different types of features. For example, one can easily extract different types of features from one image. The surrounding textual of one image can also be represented in different languages.

Learning from the training data with multiple types of features is also known as

¹The number is from <https://www.flickr.com/photos/franckmichel/6855169886/>, visited on July 31, 2014.

²The number is from <https://www.youtube.com/yt/press/statistics.html>, visited on July 31, 2014.

the *multi-view learning* problem, in which each type of features is referred to as one view. How to effectively utilize *multiple views* of information is also a key issue when learning from web data, especially in the context of weak supervision.

- Finally, we note that there are intrinsic differences between the crawled web training data and the testing images in daily lives, which is also known as the *domain adaptation* problem.

In this thesis, we investigate the above three issues in learning from web data, and develop novel learning algorithms to address those three issues. We also conduct extensive experiments to evaluate our proposed methods. A brief introduction on those three issues and our corresponding contributions are summarized as follows.

1.1 Learning from Weakly Labeled Data

Web images are usually associated with some surrounding text (*e.g.*, caption, tags, *etc.*), which somehow provides a rough description on the content of the image. By extracting words from those surrounding textual descriptions, we can obtain *weak labels* for each web images. Apparently, those weak labels are not accurate, and contain considerable noise. We refer to such web images with inaccurate labels as *weakly labeled data*.

Intuitively, if we directly use the weakly labeled data as the training data for supervised learning, the learnt classifiers are usually unsatisfactory due to the noisy training labels. To learn robust classifiers from weakly labeled data, we propose to use the multiple instance learning or multi-instance learning (MIL) model. In the multi-instance learning problem, the training data is provided in the form of training bags. Each training bag contains a small group of training instances. Only the label of each training bag is known (positive or negative), and the labels of training instances inside the training bags remain unknown. This is the major difference between MIL and the traditional supervised learning, where traditional supervised learning assumes each training instance is associated with a label. Moreover, in MIL, we usually assume that each positive bag contains at least one true positive instance, and all instances in the negative bags are negative.

When using MIL for learning from web images, we treat each image as an instance. Specifically, given a class name, we retrieve *relevant images* from Internet whose surrounding texts contains the class name, and also retrieve the same number of *irrelevant images* whose surrounding texts do not contain the class name. Then we partition the relevant images (*resp.*, irrelevant images) into small subsets, and treat those subsets as positive training bags (*resp.*, negative training bags). As the relevant images usually contain a portion of images related to the class name, the MIL constraint for positive training bags (*i.e.*, each positive bag contains at least one true positive instance) should be satisfied. Similarly, the MIL constraint for negative training bags should also be satisfied, because it is unlikely that the irrelevant images contain any image related to the class name. Then, we can train MIL classifiers by using those constructed training bags. The major advantage of using MIL model for learning from web data is that, in MIL, we assume the the labels of training instances in positive bags are unknown, which allows us to discover the training labels for those instances during the training process. Intuitively, if we can correctly recover the training labels, the performance of the learnt classifier should be improved, when compared with that learnt by the traditional supervised learning methods, because they can only use the noisy labels.

However, recovering the labels for training instances in positive bags is a nontrivial task. By formulating the MIL into an optimization problem, we arrive at a mixed-integer programming problem, in which we not only need to optimize the classifier variables, but also need to infer the labels of training instances in positive bags. The previous work in [11] used a heuristic approach to solve this optimization problem by alternately updating the classifier variables and inferring the instance labels, which can only obtain a local minimum. In this work, we design a new multiple instance learning algorithm called *Multiple Instance Learning with Constrained Positive Bags (MIL-CPB)*, in which we relax the multiple instance learning problem as a convex optimization problem, and optimize it to the global optimum. Specifically, as the labels for the instances in positive training bags are uncertain, we generate a set of candidate label vectors, each of which is a possible labeling for the training instances. Instead of directly optimizing the label vector in the candidate label vector set, we aim to find an optimal linear combination of those label vector candidates such that the learnt classifier achieves minimum loss and

maximum margin. We show that the relaxed formulation shares a similar form with the multiple kernel learning (MKL) problem, and can be solved to the global optimum by using existing MKL solvers. We also additionally propose a progressive scheme (PMIL-CPB) to iteratively construct new positive bags by using the top-ranked web images. We conduct experiments on publicly available datasets (*i.e.*, NUS-WIDE and Google datasets), and the experimental results demonstrate our methods outperform existing multiple instance learning methods for image retrieval task.

1.2 Learning from Multi-view Representations

Besides multiple instance learning, other learning scenarios have also been exploited for learning from weakly labeled data. For example, in semi-supervised learning (SSL) [12, 13], the training data consists of a limited number of labeled training samples and a large number of unlabeled training samples. In the recent work [14], training data with uncertain labels was referred to as weakly labeled data, and those learning problems were uniformly referred to as the *weakly labeled learning* problem. A unified approach was proposed in [14] for solving various learning problems with weakly labeled data including SSL, MIL and clustering, in which they learnt an optimal classifier from all possible labelings of training data.

When training data is represented with multiple feature representations, researchers have developed many multi-view learning approaches to improve the performance by utilizing information from different views [1, 15–23]. Most of those works (*e.g.*, co-training [1, 15–19]) in multi-view learning were proposed for the multi-view SSL scenario. It has been shown that, with multi-view information, the classifiers trained on different views can effectively help each other to better utilize the unlabeled training data. Nevertheless, it is unclear how to extend those works to the more general weakly labeled learning problem.

In this work, we study the learning problem by using weakly labeled training data with multiple views, which is referred to as the *multi-view weakly labeled learning problem*. We propose a novel *co-labeling* approach to learn robust classifiers by effectively utilizing different types of weakly labeled training data with multiple views of features from a broad

range of applications including the traditional multi-view SSL problem and the multi-view MIL problem. We also present a theoretical analysis on our co-labeling approach. We first analyze the generalization bound of weakly labeled learning method on each view, and then introduce a generalization bound of the final classifier for multi-view weakly labeled learning by integrating the classifiers from all views. Finally, we conduct extensive experiments for different weakly labeled learning scenarios including multi-view SSL and multi-view MIL. The experimental results show that our co-labeling approach for multi-view weakly labeled learning outperforms the existing multi-view methods and the related state-of-the-art methods for SSL or MIL.

1.3 Learning from Heterogeneous Domain

One basic assumption in traditional machine learning is that the training and test samples are independently sampled from an identical distribution (*i.e.*, the *i.i.d.* assumption). However, such assumption may not be satisfied in the problem of visual recognition by learning from web data, because the training web images usually are different from the test images. For example, suppose our task is to classify the objects in offices, such as computer, desk, chair, cup and so on. After using those words as keywords to collect the images from Internet, we usually observe that the collected images look different from the images of office objects in terms of background, viewpoints, illuminations, *etc.* From the perspective of statistical learning, this leads to a statistical difference between the training samples and the test samples, which is also known as the *domain adaptation* problem in machine learning.

Usually, we assume that we have abundant labeled training data (referred to as *source domain*) whose distribution is different from the test data, and we also have a limited number of labeled data or only unlabeled data which has the same distribution as the test data (referred to as *target domain*). The goal of domain adaptation is to learn robust classifiers that perform well on the test data.

Many domain adaptation works in computer vision assume that the samples in source and target domains have the same feature representation, and the major concern in those works is to reduce the distribution mismatch between the training samples in source and

target domains. However, we often encounter the situation that the feature representation of source domain samples is different from that of target domain samples, which is also known as the *heterogeneous domain adaptation (HDA)* problem. For example, various datasets have been released on Internet for public access. However, some datasets only contain the extracted features, while the raw data is not provided due to the privacy or license issues. If we utilize those features for other tasks (*e.g.*, office object recognition), their feature representations are different from that of the test data. Moreover, with heterogeneous domain adaptation, one can also integrate the knowledge in different medias into the learning task, *e.g.*, image recognition by learning from videos or textual descriptions.

To this end, we build upon the recent work on Heterogeneous Feature Augmentation (HFA) [24], and reformulate it into a convex optimization problem which shares a similar form with multiple kernel learning (MKL). In this way, the global optimum can be guaranteed. We further extend HFA to semi-supervised HFA (SHFA) to further utilize the unlabeled data in the target domain, which again can be reformulated as an MKL problem. We conduct experiments on image recognition, multi-lingual document classification, and cross-lingual sentiment classification, which demonstrate that the proposed methods outperform the existing HDA methods.

1.4 Connections

In the above sections, we have introduced three major issues in learning from web data and our correspondingly proposed algorithms for those issues. From the perspective of application, those three works are motivated by the task of learning from web data. Specifically, the first topic is to solve the weak supervision problem of the web data, where we introduce MIL model to handle the weakly labeled training data. In the second topic, on one hand, we extend our MIL method to handle other forms of weakly labeled web data; on the other hand, we also extend our MIL method for weakly labeled data with multi-view representations. Finally, we aim to apply the classifier learnt from web data to visual recognition tasks our daily lives, which induces the domain adaptation problem. We also consider to use as less as possible human efforts by considering a semi-supervised setting in the target domain.

In terms of methodology, the algorithms proposed in the above three topics are based on support vector machines (SVMs) and multiple kernel learning (MKL). Particularly, for the issue of weak supervisor, we propose the MIL-CPB method, in which we use an MKL formulation to solve the MIL problem. The MKL based formulation is a convex and tight relaxation to the original MIL problem. Second, in the multi-view weakly labeled learning problem, the weakly labeled learning problem on each view can also be formulated as an MKL based formulation similarly as in MIL-CPB. We further consider the multi-view property of training data by using the classifier on one view to construct the base kernels for the classifier on another view. Finally, for the heterogeneous domain adaptation, we reformulate the HFA problem as an MKL problem, which guarantees the global optimal solution. Based on this reformulation, we further extend HFA to semi-supervised HFA by exploiting the unlabeled samples in the target domain. We show that SHFA can also be formulated as an MKL problem after certain relaxation.

1.5 Thesis Outline

The rest of this thesis is organized into four chapters. We show the structure of this thesis in Figure 1.5, and summarize the main content of each chapter as follows:

Chapter 2. Literature Review In this chapter, we give a comprehensive review of the related works on the three topics studied in this thesis, *i.e.*, weakly labeled learning, multi-view learning, and domain adaptation.

Chapter 3. Progressive Multiple Instance Learning for Image Retrieval In this chapter, we study the label noise issue when learning from web data. A new multi-instance learning method called multi-instance learning with constrained positive bags (MIL-CPB) is presented for text-based image retrieval. We also extend our MIL-CPB method to Progressive MIL-CPB (PMIL-CPB) by progressively selecting confident relevant images to construct new training bags. We demonstrate the effectiveness of our proposed methods for image retrieval and image classification tasks.

Chapter 4. Co-Labeling for Multi-view Weakly Labeled Learning Problems In this chapter, we study the problem of learning from data with multi-view representations. We propose a unified co-labeling approach to handle the multi-view data with various

Figure 1.1: An illustration on the rest of this thesis.

types of weak labels, which can be used to solve many traditional multi-view learning problems, such as multi-view semi-supervised learning, multi-view multi-instance learning, and so on. We also analyze the generalization bound for our proposed method. Extensive experiments on multi-view semi-supervised learning for document classification and multi-view multi-instance learning for image retrieval demonstrate the effectiveness of our proposed approach.

Chapter 5. Semi-supervised Heterogeneous Feature Augmentation In this chapter, we study the heterogeneous domain adaptation problem. Based on the recent heterogeneous feature augmentation (HFA) method, we also map the source and target domain samples into a common subspace by using two projection matrices. By decomposing the transformation metric into a linear combination of rank-one matrices, we reformulate the learning problem into a convex formulation, which shares a similar form with the MKL problem, and can be optimized to obtain the global optimal solution. We also theoretically prove the convergence of our proposed algorithm. When the target domain contains additional unlabeled samples, we extend the proposed HFA method to semi-supervised HFA, which again can be relaxed into an MKL problem. Extensive experiments including image classification, document classification and sentiment classification clearly demonstrate the effectiveness of our proposed HFA and SHFA methods.

Chapter 6. Conclusions and Future Work In this chapter, we conclude the contributions of this work, and also point out the possible directions of learning from web data in the future.

1.6 Notations

For ease of presentation, we introduce the notations used in the following chapters. A list of notations is also summarized in page xv for reference. We use a lowercase letter (*e.g.*, a) to represent a scalar, and a boldface lowercase letter to represent a vector (*e.g.*, $\mathbf{a}$). A matrix is denoted by a boldface uppercase letter (*e.g.*, $\mathbf{A}$). We also use the superscript $'$ to denote the transpose of a vector or a matrix.

We define $\mathbf{I}_n$ as the $n \times n$ identity matrix and $\mathbf{O}_{n \times m}$ as the $n \times m$ matrix of all zeros. We also define $\mathbf{0}_n, \mathbf{1}_n \in \mathbb{R}^n$ as the $n \times 1$ column vectors of all zeros and all ones,

respectively. For simplicity, we also use $\mathbf{I}$, $\mathbf{O}$, $\mathbf{0}$ and $\mathbf{1}$ instead of $\mathbf{I}_n$, $\mathbf{O}_{n \times m}$, $\mathbf{0}_n$ and $\mathbf{1}_n$ when the dimension is obvious. The ℓ_p -norm of a vector $\mathbf{a} = [a_1, \dots, a_n]'$ is defined as $\|\mathbf{a}\|_p = (\sum_{i=1}^n a_i^p)^{\frac{1}{p}}$. We also use $\|\mathbf{a}\|$ to denote the ℓ_2 -norm. The inequality $\mathbf{a} \leq \mathbf{b}$ means that $a_i \leq b_i$ for $i = 1, \dots, n$. Moreover, $\mathbf{P} \circ \mathbf{Q}$ (*resp.*, $\mathbf{a} \circ \mathbf{b}$) denotes the element-wise product between two matrices $\mathbf{P}$ and $\mathbf{Q}$ (*resp.*, two vectors $\mathbf{a}$ and $\mathbf{b}$). The inequality $\mathbf{v} = [v_1, v_2, \dots, v_j]' \geq \mathbf{0}$ means that $v_i \geq 0$ for $i = 1, \dots, j$, and $\mathbf{H} \succeq 0$ means that $\mathbf{H}$ is a positive semi-definite (PSD) matrix.

Chapter 2

Literature Review

In this chapter, we review the related works on the three major issues when learning from web data, including weakly labeled learning, multi-view learning and domain adaptation.

2.1 Weakly labeled learning

In this section, we review the related works on weakly labeled learning. We first give a brief review on supervised learning and the support vector machine method. Then, we review the works on multi-instance learning. Finally, we summarize the works on other weakly labeled learning scenarios.

2.1.1 Supervised Learning and Support Vector Machines

Supervised learning aims to learn classifiers by using a set of training samples and their labels, such that the learnt classifiers could predict the unseen test samples as accurately as possible. Specifically, let us denote the training data as $\mathcal{S} = \{(\mathbf{x}_i, y_i) |_{i=1}^n\}$ where $\mathbf{x}_i \in \mathcal{X} = \mathbb{R}^d$ is the i -th training sample with d as the feature dimension, $y_i \in \mathcal{Y} = \{+1, -1\}$ is the corresponding training label, and n is the number of training samples. The task of supervised learning is to learn a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ to predict any unseen test sample $\mathbf{x}$. Suppose the ground truth label of test sample $\mathbf{x}$ is y , in traditional supervised learning, we usually assume that the test sample $(\mathbf{x}, y)$ is sampled independently from the identical distribution with the training samples $\mathcal{S}$, which is also known as the *i.i.d.* assumption.

Many algorithms have been proposed to solve the supervised learning problem, such as k -Nearest Neighbors (k NN), Support Vector Machines (SVM), Neural Networks (NN,

By introducing a dual variable α_i for each of the constraint $y_i(\mathbf{w}'\phi(\mathbf{x}_i) + b) \geq 1 - \xi$ in (2.1) and setting the derivatives of Lagrangian w.r.t. the primal variables as zeros, we arrive at the dual form of (2.1) as follows,

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \boldsymbol{\alpha}' (\mathbf{K} \circ \mathbf{y} \mathbf{y}') \boldsymbol{\alpha} - \mathbf{1}' \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq C \mathbf{1}, \end{aligned} \tag{2.2}$$

where $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]'$ is the vector of dual variables, $\mathbf{K} = [k_{ij}] \in \mathbb{R}^{n \times n}$ is the kernel matrix with $k_{ij} = \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j)$ being the inner product of the two samples, and $\mathbf{y} = [y_1, \dots, y_n]'$ is the vector of training labels. It can be observed that in the dual form (2.2), one only needs to define the kernel matrix $\mathbf{K}$ rather than to explicitly define the feature mapping function $\phi(\cdot)$ in (2.1). Thus, any positive semi-definite (PSD) matrix can be used as the kernel matrix to train SVM classifiers, which allows us to design various nonlinear kernel functions without explicitly defining feature mapping function, such as Gaussian kernel, Histogram Intersection kernel, *etc.* The space of the feature mapping function induced by a kernel matrix $\mathbf{K}$ is also called the Reproduce Kernel Hilbert Space (RKHS). It is worth mentioning that the dual form of SVM in (2.2) is a quadratic programming (QP) problem, which means the global optimum solution can be guaranteed by using many existing optimization techniques [26]. Researchers have also developed efficient algorithm such as Sequential Minimal Optimization (SMO) [25] to solve the above problem. An excellent toolbox on SVM called libSVM [26] can be found in <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

2.1.2 Multiple Instance Learning

In multi-instance learning (MIL), the training data is provided in the form of bags, each of which contains multiple instances. Formally, let us denote the training data as $\{(\mathcal{B}_I, Y_I) |_{I=1}^N\}$, where $\mathcal{B}_I$ is the I -th training bag, $Y_I \in \{+1, -1\}$ is the corresponding bag label, and N is the total number of training bags. Each training bag can be further represented as $\mathcal{B}_I = \{\mathbf{x}_i | i \in \mathcal{I}_I\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is an training instance with d being the feature dimension, $\mathcal{I}_I$ is the set of indices of instances that belong to the I -th training bag. Usually, we assume $\mathcal{I}_I \cap \mathcal{I}_J = \emptyset$ for any $I \neq J$, in other words, each instance only belongs to one training bag. Note that in MIL, only the bag labels are known,

while the labels of training instances are unknown. Moreover, the labels of instances should satisfy the MIL constraints that each positive training bag contains at least one positive instance, and all the instances in negative training bags are negative, which can be formally described as,

$$\begin{cases} \sum_{i \in \mathcal{I}_I} \frac{y_i + 1}{2} \geq 1, & \text{if } Y_I = 1 \\ y_i = -1, \forall i \in \mathcal{I}_I, & \text{if } Y_I = -1, \end{cases} \quad (2.3)$$

where each $y_i \in \{-1, +1\}$ is the label of the i -th instance. Specifically, the constraint in the first row of (2.3) describes that each positive training bag contains at least one positive instance, and the constraint in the second row of (2.3) describes that all instances in negative bags are true negative. As the second constraint is obvious, we give more explanation on the first constraint in the following. For any instance $\mathbf{x}_i$ with $i \in \mathcal{I}_I$, if $y_i = 1$ (*i.e.*, it is positive), then we have $\frac{y_i + 1}{2} = 1$; otherwise, we have $y_i = -1$, and $\frac{y_i + 1}{2} = 0$. In other words, $\sum_{i \in \mathcal{I}_I} \frac{y_i + 1}{2}$ actually counts the number of positive instances in bag $\mathcal{B}_I$. So the inequality $\sum_{i \in \mathcal{I}_I} \frac{y_i + 1}{2} \geq 1$ describes that there is at least one positive instance in each positive bag $\mathcal{B}_I$. Finally, the goal of MIL is to train a classifier $f(\mathbf{x})$ to predict the label of an unseen instance, or a classifier $f(\mathcal{B})$ to predict the label of an unseen bag.

The MIL problem originated from the drug activity prediction problem, where the goal is to predict whether a new molecule can be used for producing some drug [27]. Usually, a molecule may have many different shapes, and one (or a few) of these shapes can be used for making drug (*i.e.*, are active). After measuring the shape of molecule, one can represent each shape of molecule as a feature vector. Then the drug activity prediction problem can be modeled as an MIL problem, where each type of molecule is a training bag, and different shapes of molecule are training instances. The task is to predict whether a dose of drug, which contains various shapes of molecules, is active or not.

Dietterich *et al.* [27] proposed three *axis-parallel rectangle (APR)* algorithms to solve the MIL problem, in which they attempt to find appropriate rectangles parallel to axes by the conjunction of features. They first initialize APR to cover all the instances in positive bags, and then gradually shrink the APR such that at least one instance in each positive bag can be covered and the instances in negative bags are excluded from the APR

as possible. After that, researchers have developed many algorithms to solve the MIL problem. Maron and Lozano-Pérez [28] proposed the Diverse Density (DD) method, in which they regard each bag as a manifold and assume that different positive bags should intersect at several points (which are referred as high diverse density area). Intuitively, as each training bag contains at least one of those active shapes, the intersection point (*i.e.*, high diverse density area) should correspond to the active shapes of molecule. Wang and Zucker [29] proposed two methods called Citation- k NN and Bayesian- k NN, and Zhang and Goldman [30] extended the DD method to EM-DD.

It is worth noting that in Maron and Lozano-Pérez’s paper [28], they not only focused on the drug activity prediction problem, but also applied MIL to other applications such as image classification and stock selection problem, which generalizes MIL as a general learning problem. In their following work [31], they also applied DD to the natural scene classification problem, in which they divide each image into many small square patches, and treat each images as a bag and the patches in the image as instances¹. After that, researchers also applied MIL to other computer vision applications. Chen *et al.* proposed two methods DD-SVM [32] and MILES [33] for region-based image categorization. They first divide each image into different regions using segmentation method, and they treat each region as an instance and the whole image as a bag. Viola *et al.* [34] proposed a method called MILBoost by extending the boosting algorithm into MIL scenario, and use it for the object detection task. They assume that only the rough position of the object is known, and then sample many subimages around the object as positive training instances. By treating all the subimages around one object as a positive training bags, they formulate it as an MIL problem and use the MILBoost method to solve it. In the following work [35], Babenko *et al.* extended MILBoost for sequential training bags and apply it for object tracking, and two works on online MIL [36] or batch mode MIL [37] also can be applied to the object tracking task.

Recently, several works have been proposed for handling label noise in web data, in which they first group the images/videos retrieved from the Internet into different groups, and then treat each group of images/videos as a training bag, and the images/videos

¹In their paper, they also evaluated other configurations by treating a rows of patches, a patches and its four neighbors, a pair of patches, or a pair of patches as well as their neighbors as an instance.

insides the groups as the instances. Vijayanarasimhan and Grauman [38] used a variant of Sparse MIL (sMIL) [39] for image re-ranking and categorization. Leung *et al.* [40] used a variant of MILBoost for video classification. Li *et al.* [41] proposed the MIL-CPB method for web image retrieval and categorization, in which they generalize the traditional MIL constraint as that a positive bag contains at least a portion of positive instances, which is more suitable for this task as most of the associated textual descriptions are more or less related to the content of web images.

SVM Based MIL Methods: In terms of methodology, the proposed algorithms in this thesis are based on SVM. So we review the related SVM-based MIL methods below. Generally, MIL algorithms can be categorized into instance-based methods and bag-based methods. The goal of bag-based methods is to learn a classifier $f(\mathcal{B})$ to perform the bag prediction. The basic idea of bag-based methods is to convert each training bag into a single training sample. Recall that in MIL, although the labels of instances are unknown, the labels of training bags are given. By converting each training bag as a single training sample, the MIL problem becomes a standard supervised learning problem, and thus can be solved by using the SVM classifier.

Many approaches have been designed to convert the training bags into training samples. The DD-SVM method [32] first selects some prototypes $\{\mathbf{p}_m \in \mathbb{R}^d\}_{m=1}^M$ in the instance feature space that correspond to the peaks and valleys of the DD function. Then they calculate the distance between a bag $\mathcal{B}$ and prototype $\mathbf{p}_m$ (*i.e.*, $\text{dist}(\mathcal{B}, \mathbf{p}_m)$) by using the minimal weighted distance between each instance inside the bag and the prototype. Finally, they concatenate all distances $\text{dist}(\mathcal{B}, \mathbf{p}_m)$ to form an M -dimensional feature vector for each training bag, and used standard SVM to learn the bag classifier. The MILES method [33] integrated the prototype selection and classifier learning process together, in which they used all the instances as the prototypes, and employ an ℓ_1 -norm SVM to jointly learn the classifier and select features (prototypes). Recently, Fu *et al.* [42] proposed an approach to iteratively select the prototypes and train an SVM classifier.

Gartner *et al.* [43] proposed an approach to convert an instance-based kernel as a bag-based kernel. Specifically, the feature mapping function is defined as $\psi(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{I}_I} \phi(\mathbf{x}_i)$ where $\phi(\mathbf{x}_i)$ is the non-linear feature of the i -th instance in the RKHS

space. It was also shown in [43] the bag-based kernel obtained in this way is an valid kernel by verifying the properties of kernel matrix. Based on the MI-kernel, Bunescu and Mooney [44] proposed a method called sparse MIL (sMIL) in which they also adjust the label of positive bag by using the average of the labels inside the positive bag². Zhou *et al.* [45] proposed two methods MIGraph and miGraph, in which they define bag-based kernel by considering each bag as the a graph of instances. It is worth noting that although bag-based MIL methods are originally designed for bag prediction, they can also be applied for instance prediction by identifying the importance of instance [32, 33], or by directly considering each test instance as a single-instance bag [38, 43, 44].

Instance-based MIL methods, on the other hand, aim to learn a prediction function $f(\mathbf{x})$ to predict each test instance. However, it is a nontrivial task, because the labels of training instances are unknown. Based on SVM, the learning problem can be formulated as follows [11],

$$\begin{aligned} \min_{\mathbf{y} \in \mathcal{Y}, \mathbf{w}, b, \xi_i} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}'\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i = 1, \dots, n, \end{aligned} \quad (2.4)$$

where $\mathbf{y} = [y_1, \dots, y_n]'$ is the label vector of all training instances, n is the total number of training instances, $\mathcal{Y}$ is the feasible set of label vector $\mathbf{y}$ which satisfies the MIL constraints in (2.3).

Compared with the formulation of SVM in (2.1), the objective in (2.4) needs to additionally optimize the label vector $\mathbf{y}$, which leads to a mixed-integer programming (MIP) problem. It is generally NP-hard, and is difficult to solve. Andrews *et al.* [11] proposed two heuristic methods, mi-SVM and MI-SVM, to solve the above problem. In mi-SVM, they first initialize the labels of instances in positive bags as positive and the labels of instances in negative bags as negative. Then they iteratively train an SVM classifier and use the SVM classifier to predict the labels of instances. They ensure the predicted labels satisfying the MIL constraints by setting the label of the instance with largest decision value in each positive bag as positive, and the labels of all instances in

²By considering the worst case, *i.e.*, each positive bag contains only one positive instance, the label of a positive bag can be represented as $\frac{2-|\mathcal{B}|}{|\mathcal{B}|}$

negative bags as negative. In MI-SVM, they use the instances in negative bags as the negative training samples, and select one instance from each positive bag as the positive training samples. They first initialize the positive samples by using the average of the instances in each positive bag. Then they iteratively train an SVM classifier, and use the learnt classifier to select the instance with highest decision value in each positive bag as the positive training samples.

As the mi-SVM and MI-SVM methods are easily stuck in a local optimum, Li *et al.* [41] proposed a method called MIL-CPB for web image retrieval task. Instead of directly optimizing the label vector, they optimize the linear combination of feasible label vectors in the dual form, which leads to an MKL problem. It has been shown that such an MKL problem is a tight convex relaxation of the original problem [41, 46]. Similarly, Li *et al.* [47] proposed the KI-SVM method for interest region selection task, which can be deemed as an convex relaxation of MI-SVM.

Moreover, a comprehensive review on early MIL works can be found in [48]. Foulds and Frank gave a discussion on different MIL assumptions in [49], and Amores also reviewed existing MIL methods and conducted experimental analysis in [50].

2.1.3 Weakly Labeled Learning

Besides multi-instance learning, researchers have also studied other learning scenarios when the training labels are incomplete.

Semi-supervised Learning: One popular learning scenario is semi-supervised learning (SSL). Specifically, in SSL, the training data consists of a limited number of labeled training data and a large number of unlabeled training data. Formally, we represent the training data as $\mathcal{S} = \mathcal{S}_l \cup \mathcal{S}_u$ with $\mathcal{S}_l = \{(\mathbf{x}_i, y_i) |_{i=1}^l\}$ and $\mathcal{S}_u = \{x_i |_{i=l+1}^n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the i -th training sample, y_i is the corresponding label for the i -th labeled sample, l is the number of labeled samples, and n is the total number of training samples. Zhu gave a comprehensive survey on SSL in [51]. The task of SSL is to learn a robust classifier by additionally leveraging the large number of unlabeled samples. In the literature, two common assumptions on the distribution of unlabeled data, *cluster assumption* and *manifold assumption*, have been widely employed by SSL methods in explicit or implicit ways [52].

The cluster assumption assumes that two samples are likely to belong to the same class if they are connected by a line cross a high density region, which means the classification hyperplane should be lie in the low density region area. Many algorithms have been proposed based on the cluster assumption, such as Transduction SVM (TSVM) [12], Low density Separation (LDS) [53], and Semi-Supervised SVM (S3SVM) [13]. The most representative work in this category is TSVM, which aims to simultaneously learn the SVM classifier and infer the labels of unlabeled training samples. Similarly as in instance-based MIL, inferring the labels for unlabeled training samples is NP-hard. Collobert *et al.* [54] employed the concave-convex procedure (CCCP) to solve TSVM problem, which can be applied to large scale problems.

The manifold-based assumption, on the other hand, assumes that the training samples lie in a low-dimensional manifold, so the learnt target function should change smoothly along the manifold. The most representative work is Laplacian SVM (LapSVM) [52, 55], in which a graph laplacian based regularizer is applied on the unlabeled data to enforce the decision values of two unlabeled samples to be similar if they are close to each other in the graph. It has been shown that LapSVM can be converted into a standard SVM problem by using an embedded kernel [55].

Max-margin Clustering: Another learning scenario with incomplete training labels is the unsupervised learning, where all the training labels are unknown, and the task is to cluster the training data into several clusters. The most related unsupervised learning methods are the max-margin clustering (MMC) approaches [46, 56, 57], in which the task is to cluster the training data into two clusters separated with a large margin. Similarly as in instance-based MIL, the MMC problem is also an NP-hard mix-integer programming problem, and is difficult to solve. Xu *et al.* [56] relaxed it as a semi-definite programming (SDP) problem, and Zhang *et al.* [57] employed an iterative method by sequentially solving a series of SVM or support vector regression (SVR) problems. Li *et al.* [46] proposed a tighter relaxation of MMC, which shares a similar formulation as the MKL problem. Based on the cutting-plane algorithm, an efficient algorithm was developed by iteratively solving an MKL problem.

By observing that the essence of traditional MIL, SSL and MMC tasks is the training data contains unknown labels, the recent works [14, 58] unified those learning scenarios

as one problem called *weakly labeled learning*. Li *et al.* [14] proposed a unified framework called WellSVM to solve the weakly labeled learning problem, while Joulin and Bach [58] solved another form of convex relaxation for weakly labeled learning. Other learning scenarios for using data with incomplete labels include Relative Outlier Detection (ROD) [59–61], Positive and Unlabeled (PU) learning [62], and Multi-instance Semi-supervised (MISSL) [63], and so on.

2.2 Multi-view Learning

When the training data is represented with multiple views of features, researchers have developed many algorithms to integrate information from different views. Formally, let us denote $\mathcal{S} = \{(\mathbf{x}_i, y_i) |_{i=1}^n\}$ as the training data. Different from the single-view learning where each training sample $\mathbf{x}_i$ is a feature vector, in multi-view learning, each training sample can be represented as $\mathbf{x}_i = \{\mathbf{x}_i^1, \dots, \mathbf{x}_i^V\}$, where each $\mathbf{x}_i^v$ for $v = 1, \dots, V$ is a feature vector on the v -th view, and V is the total number of views. Note that the dimensions of features in different views are not necessarily the same. A recent survey on multi-view learning algorithms can be found in [64]. According to the supervision information on training data, we review the related works on multi-view learning in three folds, unsupervised multi-view learning, supervised multi-view learning, and semi-supervised multi-view learning.

In unsupervised multi-view learning, the labels of training data y_i 's are not used, and the task is to identify the view-invariant information among different views, or to cluster the multi-view data into different clusters. One of the most representative work is the Canonical Correlation Analysis (CCA) method [65, 66], which learns two projection matrices for two views by maximizing the canonical correlation between the features. It was also extended to kernel CCA (KCCA) [66] for learning projection matrices in RKHS. For the multi-view clustering, Bickel and Scheffer studied two approaches in [67], and Chaudhuri *et al.* employed CCA in [68]. Recently, Kumar *et al.* [20, 21] proposed two methods for multi-view clustering by exploiting the co-training and co-regularization approaches, respectively.

In supervised multi-view learning, the labels of all training samples are known, and the task is to learn robust classifiers for predicting the unseen multi-view samples. Farquhar *et al.* [69] proposed the SVM-2K method, in which they aim to learn two SVM classifiers on two-views, and also enforce the decision values of training samples from those two classifiers should be similar. The recent works on multiple kernel learning (MKL) [70–73] can also be applied to the supervised multi-view learning by constructing a kernel for each view of features. Given multiple kernels on training samples, the MKL method aims to learn an optimal linear combination of those kernels and simultaneously train an SVM classifier based on the combined kernel.

In semi-supervised multi-view learning (or multi-view semi-supervised learning), similarly as semi-supervised learning, only a limited number of training labels are known. One of the pioneering works is the co-training method [15], which was originally proposed for semi-supervised learning problems with two views. It was extended by using SVMs as base classifiers in [23], which was subsequently adapted for unsupervised learning [22]. Co-training was also extended to tri-training [74] and co-forest [75] to handle more than two views. However, the co-training style algorithms work under strict assumptions that each view is sufficient to train a low-error classifier and both views are conditionally independent, which might not be satisfied on real world datasets [76]. Many works attempt to relax those assumptions from various perspectives, such as weak dependence [77], α -expansion [78], large diversity [76] and label propagation [79]. Recently, co-training with insufficient views has also been theoretical analyzed in [80]. Besides co-training style methods, other methods such as co-regularization based approaches [1, 17–19] were also proposed to train classifiers on different views based on a so-called *co-regularization* criterion, which is used to minimize the differences of the decision values from the classifiers on different views. Recently, Li *et al.* [81] proposed a novel *co-labeling* approach to solve the general multi-view weakly labeled problem, including multi-view semi-supervised learning, multi-view multi-instance learning, multi-view max-margin clustering and so on.

2.3 Domain Adaptation

Domain adaptation aims to solve the problem that the distribution of test data is different from that of the labeled training data. In traditional machine learning, one basic

assumption is that the training and test samples are drawn independently from an identical distribution, which is also known as the *i.i.d.* assumption. Since many statistical learning principles are built on this assumption, they may not hold in domain adaptation problem, which means the learnt classifier may be significantly degraded when being applied on unseen test samples. The task of domain adaptation is to train a robust classifier that performs well in the domain of unseen test samples. A few works have reviewed the domain adaptation methods from different perspectives [82–88].

Usually, we refer to the domain of test samples as *target domain*, while the domain contains ample labeled training data is referred to as *source domain*. We also assume there are a limited number of labeled samples or only unlabeled samples from target domain in the training process, such that we can improve the classifier by leveraging the training samples from both domains. Moreover, we only discuss the single source domain adaptation works, and the works on domain adaptation with multiple sources can be founded in the aforementioned reviews. Formally, let us denote the training data as $\mathcal{X} = \mathcal{S} \cup \mathcal{T}$, where $\mathcal{S}$ is the source training data and $\mathcal{T}$ is the target training data. In more details, we denote the source training data as $\mathcal{S} = \{(\mathbf{x}_i^s, y_i^s) |_{i=1}^{n_s}\}$, where $\mathbf{x}_i^s$ is the i -th training sample in source domain, y_i^s is the corresponding label, and n_s is the total number of training samples in source domain. Similarly, we denote the target training data as $\mathcal{T} = \{(\mathbf{x}_i^t, y_i^t) |_{i=1}^{n_t}\}$, where $\mathbf{x}_i^t$ is the i -th training sample in the target domain, y_i^t is the corresponding label, and n_t is the total number of training sample in the target domain. Note the labels the target domain training samples might be only partially observed or even completely unknown in different domain adaptation scenarios, and the feature representations of $\mathbf{x}^s$ and $\mathbf{x}^t$ are also not necessarily the same (see details below). Let us also denote the source distribution as P^s (*i.e.*, $(\mathbf{x}_i^s, y_i^s) \sim P^s$), and the target distribution as P^t (*i.e.*, $(\mathbf{x}_i^t, y_i^t) \sim P^t$), we have $P^s \neq P^t$ in domain adaptation problem. The goal of domain adaptation is to learn a classifier $f(\mathbf{x}^t)$, which can correctly predict the label of an unseen sample $\mathbf{x}^t$ given $(\mathbf{x}^t, y^t) \sim P^t$.

2.3.1 Homogeneous Domain Adaptation

Many previous works are on homogeneous domain adaptation, in which the source samples $\mathbf{x}_i^s$'s and the target samples $\mathbf{x}_i^t$'s share the same feature representation. According

to the availability of supervision information in the target domain, it can be categorized as supervised domain adaptation where only a limited number of labeled samples are given in the target domain, semi-supervised domain adaptation where a limited number of labeled samples as well as ample unlabeled samples are provided in target domain, and unsupervised domain adaptation where only unlabeled samples are provided in the target domain.

The supervised domain adaptation approaches aim to jointly learn a good target classifier, or to adapt the existing source classifier to the target domain. Yang *et al.* [89] proposed an adaptive SVM (A-SVM) approach to adapt the source classifier to the target domain by learning a perturbation classifier. Duan *et al.* [90] also proposed two domain adaptation methods based multiple kernel learning by adapting the source classifier. Daumé III [91] proposed a simple approach by augmenting the source and target samples with their original features and zeros, such that the similarities of samples within the same domain can be enhanced after calculating the kernel.

Daumé III *et al.* [92] also extend the work in [91] to semi-supervised domain adaptation by employing the co-regularization technique. Other works on semi-supervised domain adaptation can also be found in [93–95].

For unsupervised domain adaptation, Huang *et al.* [96] proposed to learn the density ratio between the source and target distributions based on the Maximum Mean Discrepancy (MMD) criterion [97], and then train an weighted SVM classifier by using the density ratio to reweight the source samples. Pan *et al.* [98] proposed Transfer Component Analysis (TCA) to project the source and target samples into a common subspace. Recently, unsupervised domain adaptation has attracted lots of attentions from researchers in computer vision community. Gopalan *et al.* [99] proposed an approach to generate intermediate domains by interpolating the subspaces of source and target domains on Grassmann manifold, and then obtain the domain invariant feature by projecting the source and target domain samples into those subspaces. Gong *et al.* [100] extend [99] by using infinite number of interpolation points, which leads to the Geodesic Flow Kernel (GFK). Baktashmotlagh *et al.* [101] proposed to learn a domain invariant projection based on the MMD criterion, and Fernando *et al.* [102] proposed to learn a projection to align the subspaces of source and target domains. Besides those feature-based methods,

other works were also proposed to directly learn a robust classifier [103–105]. Bruzzone and Marconcini [103] proposed the DASVM approach to adjust the source classifier to the target domain by gradually adding newly labeled target samples and removing some source samples. Chu *et al.* [104] proposed to jointly learn a weighted SVM classifier and the density ratio between two domains, and Li *et al.* [105] proposed to a convex formulation implementing the similar idea.

There are also many works studied the domain adaptation problem with multiple sources [106–108], which are not included in above discussion. Moreover, researchers have also presented various theoretical analysis on domain adaptation problem, please refer to the works in [109–119] and references therein for details.

2.3.2 Heterogeneous Domain Adaptation

When the source samples $\mathbf{x}_i^s$'s and target samples $\mathbf{x}_i^t$'s are represented with different types of features, the problem becomes heterogeneous domain adaptation (HDA) problem. The pioneering works [120–124] are limited to some specific HDA tasks, because they required additional information to transfer the source knowledge to the target domain. For instance, Dai *et al.* [120] and Zhu *et al.* [124] proposed to use either labeled or unlabeled text corpora to aid image classification by assuming images are associated with textual annotations. Such textual annotations can be additionally utilized to mine the word co-occurrence from textual annotations of images and words in text documents, which is served as a bridge to transfer knowledge from the text documents to images.

To handle more general HDA tasks, other methods have been proposed to explicitly discover a common subspace [125–127] without using additional information, such that original data from the source and target domains can be measured in the common subspace. Specifically, Shi *et al.* [125] proposed to learn feature mapping matrices based on a spectral transformation for domains with different features. Wang *et al.* [126] proposed to learn the feature mapping by using the manifold alignment. However, such manifold assumption may not be satisfied in real-world applications with very diverse data. Recently, Kulis *et al.* [127] proposed a nonlinear metric learning method to learn an asymmetric feature transformation for the source and target data with high dimensions. They assume that if one source sample and one target sample are from the same

category, the learned similarity between this pair of samples should be large; otherwise, the similarity should be small. Duan *et al.* [24] proposed Heterogeneous Feature Augmentation by extending the feature augmentation method [91] to HDA problem. They handle the feature differences by using two projection matrices such that the source and target domain features can be projected into one common space, in which the two projection matrices are learnt jointly with an SVM classifier. Li *et al.* [128] further extended HFA to semi-supervised HFA (SHFA) by leveraging the additional unlabeled data from target domain. They first reformulated the HFA problem as an MKL problem, based on which they further inferred the labels of target domain unlabeled samples, which can also be relaxed to another MKL problem. Hoffman *et al.* [129] proposed to learn a single transformation when learning the classifiers for different categories.

2.4 Summary

In this chapter, we have summarized the related works on weakly labeled learning, multi-view learning and domain adaptation. Specifically, we first briefly introduce the supervised learning problem and the support vector machine method. Then, we review the history of multi-instance learning and its applications. We then summarize existing SVM-based multi-instance learning algorithms, including bag-based MIL algorithms and instance-based MIL algorithms, and also discuss their connections with SVM in supervised learning scenario. Besides MIL methods, we also review other weakly labeled learning methods, including semi-supervised learning and max-margin learning, and introduce the recent works on weakly labeled learning at the end. Secondly, we review the related works on multi-view learning in different learning problems including multi-view unsupervised learning, multi-view supervised learning, and multi-view semi-supervised learning. The recent work on general multi-view weakly labeled learning as well as other multi-view learning scenarios are also discussed. Finally, we give a review on domain adaptation works. We review the homogeneous domain adaptation works and the heterogeneous domain adaptation works. For homogeneous domain adaptation, we summarize the popular algorithms for different domain adaptation settings, including supervised domain adaptation, semi-supervised domain adaptation, and unsupervised domain adaptation. We specifically summarize the recent advances on unsupervised domain adaptation

in computer vision. Some theoretical works on domain adaptation are also included. In the end, we give a summary on heterogeneous domain adaptation works.

In this thesis, we develop new approaches on multi-instance learning, multi-view weakly labeled learning, and heterogeneous domain adaptation for learning from web data. Although there are many works related to those learning problems in the literature as summarized above, in this thesis, we adapt those learning scenarios to handle different real-world tasks of learning from web data, and correspondingly design effective algorithms. Specifically, in Chapter 3, we propose a new MIL algorithm called multi-instance learning with constrained positive bags (MIL-CPB) for web image retrieval, in which we generalize the traditional MIL constraint on positive bags as that each positive bag contains at least *a portion of* true positive instances, considering the relevant images usually contain many true positive images. We solve a tight relaxation of MIL in our MIL-CPB by using MKL, and also extend our MIL-CPB to progressive MIL-CPB (PMIL-CPB) by gradually constructing new positive bags by using the most confident relevant images. In Chapter 4, we propose a novel co-labeling algorithm for multi-view weakly labeled learning, which can be used to learn from various type of weakly labeled web data. Different from traditional multi-view learning algorithms, which are only designed for one specific learning scenario, our co-labeling aims to solve the general weakly labeled learning problem with multi-view data, and thus can be applied to broad range of multi-view learning problems, including multi-view semi-supervised learning and multi-view multi-instance learning. Moreover, in our co-labeling approach, we model the learning problem on each view as a weakly labeled learning problem, which can be solved via an MKL formulation. In Chapter 5, we build upon the recent proposed heterogeneous feature augmentation (HFA) algorithm by reformulating it as an MKL problem, which guarantees the global solution. We further extend HFA to semi-supervised HFA (SHFA) by further considering the unlabeled training samples in the target domain. Compared with existing HDA methods, our HFA can simultaneously learn the projection matrices and the target classifier. Moreover, our SHFA is the first work to exploit the semi-supervised HDA problem.

Chapter 3

Progressive Multiple Instance Learning for Image Retrieval

In this chapter, we propose a new approach to learn a robust classifier for text-based image retrieval (TBIR) using relevant and irrelevant training web images, in which we explicitly handle noise in the loose labels of training images. Specifically, we first partition the relevant and irrelevant training web images into clusters. By treating each cluster as a “bag” and the images in each bag as “instances”, we formulate this task as a multi-instance learning problem with constrained positive bags, in which each positive bag contains at least a portion of positive instances. We present a new algorithm called MIL-CPB to effectively exploit such constraints on positive bags and predict the labels of test instances (images). Observing that the constraints on positive bags may not always be satisfied in our application, we additionally propose a progressive scheme (referred to as Progressive MIL-CPB, or PMIL-CPB) to further improve the retrieval performance, in which we iteratively partition the top-ranked training web images from the current MIL-CPB classifier to construct more confident positive “bags” and then add these new “bags” as training data to learn the subsequent MIL-CPB classifiers. We conduct experiments on two challenging real-world web image data sets to demonstrate the effectiveness of our approach.

3.1 Introduction

With the rapid adoption of digital cameras, we have witnessed an explosive growth of digital photos. Everyday, a tremendous amount of images together with rich contextual

information (*e.g.*, tags, categories and captions) are posted to the Web. There is an increasing interest in developing new systems to help users to retrieve web images. While a large number of content-based image retrieval (CBIR) algorithms (see the recent survey in [130]) have been developed over the past decades, it is more desirable and practical for a user to retrieve photos from the database using textual queries (*i.e.*, tags) as opposed to example images. However, this is a non-trivial task because the performance of text-based image retrieval (TBIR) systems may be significantly degraded when the associated contextual information is noisy and incomplete [131].

Meanwhile, the massive and valuable web data (*i.e.*, web images and the associated rich textual descriptions) have been also exploited for various computer vision tasks. For instance, Torralba *et al.* [9] used a k NN classifier for several object/scene classification tasks by leveraging 80 million loosely labeled tiny images. Wang *et al.* [132] also employed millions of web data for image annotation. Given the massive web data, relevant and irrelevant web images can be readily obtained by using keyword based search. In [8, 38, 133–135], such relevant and irrelevant web images were used as the loosely labeled training data for image categorization and retrieval (see Section 3.2 for more details).

In this work, we explicitly handle noise in the loose labels of training images. We first respectively partition the relevant web images and the randomly sampled irrelevant web images into clusters. Following [38, 133], we formulate our task as a multi-instance learning (MIL) problem by treating each cluster as a “bag” and the images in each bag as “instances”. Observing that most of the associated textual descriptions are more or less related to the content of web images, we assume that *each positive bag has at least a portion of positive instances*. Note that this new assumption on positive bags is more suitable for our TBIR task when compared with the traditional assumption in most existing MIL work (*e.g.*, [38]) that one positive bag contains at least one positive instance. However, we still use the traditional assumption on negative bags (*i.e.*, each negative bag does not contain any positive instances), as the experiments in [133] show that this assumption generally holds for negative bags. We present an algorithm called MIL-CPB to effectively solve the MIL problem with *constrained positive bags* and predict the labels of instances (images).

Since the labels of relevant training web images are quite noisy, the constraints on positive bags may not always be satisfied in our application. To cope with this issue,

in this work we propose a progressive scheme (referred to as Progressive MIL-CPB, or PMIL-CPB) to refine the quality of the positive bags. Specifically, we construct more confident “bags” from the top-ranked training web images using the current MIL-CPB classifier. After that, these new “bags” are added up to the training data in order to learn the subsequent MIL-CPB classifiers in an iterative fashion. In Section 3.4, we conduct comprehensive experiments using the challenging NUS-WIDE and Google data sets and the results clearly demonstrate the effectiveness of our approach.

3.2 Related Work

Loosely labeled training web data has been used for image categorization and retrieval. Fergus *et al.* [134] extended pLSA to handle the large intra-class variability among images returned by the image search engine. Schroff *et al.* [8] developed a multi-modal approach to collect a large image data set, in which different regularization parameters were used for the positive and negative training instances when learning the visual feature based SVM classifier. Meanwhile, researchers also proposed new tag re-ranking methods [131, 136], which are specifically designed for *Flickr* photos that are initially associated with noisy tags. In contrast to [8, 131, 134, 136], we formulate a new MIL problem with constrained positive bags to solve the learning problem with ambiguity in the training samples.

MIL methods [33, 137] have been successfully used in region-based image retrieval. In these applications, images are considered as bags; while regions in the images are considered as instances. In this work, we treat one image cluster as one “bag” and the images inside the bag as “instances”. Moreover, the existing MIL approaches (*e.g.*, Diverse Density (DD) [28], EMDD [30], Citation k NN [29]) that can predict the labels of instances are computationally expensive, making them unsuitable for large-scale TBIR applications. In contrast, our method can also efficiently predict the labels of instances (images).

Our work is quite related to [38], in which a variant (referred to as WsMIL here) of Sparse MIL (sMIL) [39] was proposed for image re-ranking and categorization by iteratively updating the weighted mean of instances in the constraint. However, our work is different from [38] in two aspects: 1) the work in [38] used the traditional assumption

that one positive bag contains at least one positive instance. We argue that our new assumption that a positive bag contains at least a portion of positive instances is more suitable for this task as most of the associated textual descriptions are more or less related to the content of web images [133]; 2) the work in [38] used pre-defined and fixed “bags” when learning the classifiers and the high-quality positive bags cannot always be obtained due to the noisy web data, while in our work we progressively improve the quality of the positive bags and iteratively add new “bags” to the training data to learn a more robust classifier. Our experiments also demonstrate that PMIL-CPB is better than WsMIL and sMIL for the TBIR task.

3.3 Multi-Instance Learning for Image Retrieval

Given a textual query q , the recent TBIR system in [138] first automatically collects relevant web images whose surrounding texts contain the word q as well as irrelevant web images whose surrounding texts do not contain the word q . Then, these relevant (*resp.*, irrelevant) web images were directly used as positive (*resp.*, negative) training data to learn classifiers (*e.g.*, SVM), and the database images are finally ranked according to their decision values from the learned classifier. We observe that a large portion of relevant web images returned by the image search engine and the method suggested in [138] are accompanied by incorrect labels because the web images are generally associated with noisy textual descriptions (*e.g.*, tags, categories and captions). Therefore, the learned classifiers based on these loosely labeled web images are not robust, which significantly degrades image retrieval performances.

To effectively utilize the noisy web data for image re-ranking and categorization, Vijayanarasimhan and Grauman [38] proposed a variant (called WsMIL here) of the existing MIL method sMIL [39] to learn classifiers by iteratively updating the weights of the instances in each positive bag. However, in their work the positive bags are assumed to be constructed by using image search engines in multiple languages. Moreover, their approach follows the traditional MIL assumption [11, 27–30, 33, 39, 139] that a positive bag contains at least one positive instance.

In this work, we first respectively partition the relevant web images and the randomly sampled irrelevant web images into clusters of the same size. Following [38, 133], we also

Figure 3.1: Flowchart of the proposed approach.

formulate our task as a multi-instance learning (MIL) problem by treating each cluster as a “bag” and the images in each bag as “instances”. Based on the observation that negative bags generally do not contain any positive instances, we formulate our task as a MIL problem with *constrained positive bags*, in which *each positive bag contains at least a portion of positive instances*.

Let us denote each instance as $\mathbf{x}_i$ with its label $y_i \in \{\pm 1\}$, where $i = 1, \dots, n$. We also represent the label of the bag B_I as $Y_I \in \{\pm 1\}$. The new MIL constraints can be interpreted as follows:

$$\begin{aligned} \sum_{i: \mathbf{x}_i \in B_I} \frac{y_i + 1}{2} &\geq \sigma |B_I| \quad \text{for } Y_I = 1, \\ \text{and } y_i &= -1 \quad \text{for } Y_I = -1. \end{aligned} \tag{3.1}$$

Note that our new MIL constraints differ from the traditional MIL assumption in that positive bags are provided with more information, namely, each positive bag contains at least a portion (*i.e.*, σ) of positive instances. Moreover, the traditional MIL assumption is just a special case of our new assumption when setting $\sigma = 1/|B_I|$.

To effectively exploit the constraints on positive bags, we first present an algorithm called MIL-CPB in Section 3.3.1. We further develop a progressive scheme called Progressive MIL-CPB (PMIL-CPB) in Section 3.3.2. The overall flowchart of our proposed approach is shown in Fig. 3.1.

3.3.1 MIL with constrained positive bags (MIL-CPB)

In our TBIR application, we rank the database images based on their decision values from the decision function (*i.e.*, classifier). And the decision function is to be learned with the instances from the training bags. Let us define the vector of instance labels as $\mathbf{y} = [y_1, \dots, y_n]'$, and denote the feasible set of $\mathbf{y}$ as $\mathcal{Y} = \{\mathbf{y} | y_i \in \{\pm 1\}, \mathbf{y} \text{ satisfies the conditions in (3.1)}\}$. We employ the formulation of Lagrangian SVM with the square bias penalty b^2 and the square hinge loss function. We propose to learn the decision function $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ in MIL-CPB by minimizing the following structural risk functional:

$$\begin{aligned} \min_{\mathbf{y} \in \mathcal{Y}, \mathbf{w}, b, \rho, \xi_i} \quad & \frac{1}{2} \left(\|\mathbf{w}\|^2 + b^2 + C \sum_{i=1}^n \xi_i^2 \right) - \rho, \\ \text{s.t.} \quad & y_i(\mathbf{w}'\phi(\mathbf{x}_i) + b) \geq \rho - \xi_i, \quad i = 1, \dots, n, \end{aligned} \quad (3.2)$$

where $\phi : \mathbf{x} \rightarrow \phi(\mathbf{x})$ is a mapping function that maps $\mathbf{x}$ from the original space into a high dimensional space $\phi(\mathbf{x})$, $C > 0$ is a regularization parameter, ξ_i 's are slack variables and $2\rho/\|\mathbf{w}\|$ defines the separation between the two opposite classes. It is worth noting that the constraints in (3.2) are defined for all n training instances, which means for each positive bag $\mathcal{B}_I$, there are $|\mathcal{B}_I|$ constraints defined for the instances inside this bag. In contrast, the constraints in the existing work sMIL [39] and WsMIL [38] for each positive bag are defined on the (weighted) mean of instances inside this bag, so there is only one constraint for each positive bag $\mathcal{B}_I$. Since the support vectors of the learnt classifier are selected from the training samples correspond to the constraints in (3.2), one could expect that in our work there would be more support vectors (defined on the instances) to represent a more precise decision boundary, and in return the learned classifier could achieve better prediction performance for instances.

We solve (3.2) in its dual form listed below by introducing a dual variable α_i for each constraint in (3.2):

$$\min_{\mathbf{y} \in \mathcal{Y}} \max_{\boldsymbol{\alpha} \in \mathbf{A}} -\frac{1}{2} \boldsymbol{\alpha}' \left(\tilde{\mathbf{K}} \circ \mathbf{y} \mathbf{y}' + \frac{1}{C} \mathbf{I} \right) \boldsymbol{\alpha}, \quad (3.3)$$

where $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]'$, $\mathbf{A} = \{\boldsymbol{\alpha} | \boldsymbol{\alpha} \geq \mathbf{0}, \mathbf{1}'\boldsymbol{\alpha} = 1\}$ is the feasible set of $\boldsymbol{\alpha}$, $\tilde{\mathbf{K}} = \mathbf{K} + \mathbf{1}\mathbf{1}'$ and $\mathbf{K} = [k(\mathbf{x}_i, \mathbf{x}_j)] \in \mathbb{R}^{n \times n}$ is the kernel matrix with the kernel function k deduced from the feature mapping $\phi(\cdot)$ (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j)$). Note that (3.3) is an integer programming problem with respect to the instance labels y_i 's, which is an NP-hard problem and therefore is computationally intractable.

Inspiring by [46], we propose to relax the problem in (3.3) by using a multiple kernel learning (MKL) [71] formulation,

$$\min_{\mathbf{d} \in \mathbf{D}} \max_{\boldsymbol{\alpha} \in \mathbf{A}} -\frac{1}{2} \boldsymbol{\alpha}' \left(\sum_{t: \mathbf{y}^t \in \mathcal{Y}} d_t \tilde{\mathbf{K}} \circ \mathbf{y}^t \mathbf{y}^{t'} + \frac{1}{C} \mathbf{I} \right) \boldsymbol{\alpha}. \quad (3.4)$$

where $\mathbf{y}^t$ is any feasible solution in $\mathcal{Y}$, d_t is the corresponding linear combination coefficient of $\mathbf{y}^t$, and $\mathbf{d}$ is the vector of d_t 's and $\mathcal{D} = \{\mathbf{d} | \mathbf{d} \geq \mathbf{0}, \mathbf{1}'\mathbf{d} = 1\}$ is the feasible set of $\mathbf{d}$.

Intuitively, instead of directly optimizing the label vector $\mathbf{y}$ in (3.3), we alternatively seek for an optimal linear combination of all feasible label vector in (3.4). As shown in [46], the problem in (3.4) is the lower bound of the problem in (3.3), and it is also a tight relaxation to (3.3).

The optimization problem (3.4) shares a similar formulation with the MKL problem. Note that the number of base kernels $\tilde{\mathbf{K}} \circ \mathbf{y}^t \mathbf{y}^{t'}$ in (3.4) is exponential in size. As a result, the existing state-of-the-art multiple kernel learning methods such as SimpleMKL [71] or LpMKL [72] cannot be used to solve (3.4), due to the heavy computational cost. To this end, we employ the cutting-plane algorithm [140] to find a subset $\mathcal{C} \in \mathcal{Y}$ of the constraints that can well approximate the original optimization problem. To find the optimal solution, we alternately update the dual variable $\boldsymbol{\alpha}$ and the linear combination coefficient $\mathbf{d}$ during the optimization procedure. Algorithm 1 lists the detailed optimization procedure for MIL-CPB. After obtaining the optimal $\boldsymbol{\alpha}$ and $\mathbf{d}$, we can derive the decision function as follows:

$$f(\mathbf{x}) = \sum_{i: \alpha_i \neq 0} \alpha_i \tilde{y}_i \tilde{k}(\mathbf{x}, \mathbf{x}_i), \quad (3.5)$$

where $\tilde{y}_i = \sum_{t: \mathbf{y}^t \in \mathcal{C}} d_t y_i^t$ and $\tilde{k}(\mathbf{x}, \mathbf{x}_i) = k(\mathbf{x}, \mathbf{x}_i) + 1$.

Convergence: In Algorithm 1, we repeatedly solve an MKL problem which minimizes the objective function in (3.4) with respect to $\boldsymbol{\alpha}$ and $\mathbf{d}$, and update the label vector set $\mathcal{C}$. Note the label vector set $\mathcal{C}$ is augmented at each iteration, in the worst case the optimal solution of MKL at the current iteration should be the same as that at last iteration by setting the entry in the coefficient vector $\mathbf{d}$ corresponding to the newly added label vector to zero. Therefore, the objective value of MKL should decrease monotonically as the number of iterations increases.

Finding violated $\mathbf{y}$: Finding the most violated constraint for $\mathbf{y}^t$ in Step 3 of Algorithm 1 is problem-specific, and it is also the most challenging part in the cutting-plane

Algorithm 1 Cutting-plane algorithm for MIL-CPB

- 1: Initialize $y_i = Y_I$ for $\mathbf{x}_i \in B_I$ as $\mathbf{y}^1$, and set $\mathcal{C} = \{\mathbf{y}^1\}$;
 - 2: Use SimpleMKL [71] to solve $\boldsymbol{\alpha}$ and $\mathbf{d}$ in (3.4) with $\mathcal{C}$;
 - 3: Use $\boldsymbol{\alpha}$ to select a violated $\mathbf{y}^t$ and set $\mathcal{C} \leftarrow \mathbf{y}^t \cup \mathcal{C}$;
 - 4: Repeat Steps 2 and 3 until convergence.
-

algorithm. We first equivalently reformulate the problem in (3.4) as follows,

$$\begin{aligned} \max_{\theta, \boldsymbol{\alpha} \in \mathbf{A}} \quad & -\theta \\ \text{s.t.} \quad & \theta \geq \frac{1}{2} \boldsymbol{\alpha}' \left(\tilde{\mathbf{K}} \circ \mathbf{y}^t \mathbf{y}^{t'} + \frac{1}{C} \mathbf{I} \right) \boldsymbol{\alpha}, \forall \mathbf{y}^t \in \mathcal{Y}, \end{aligned} \tag{3.6}$$

According to (3.6), *the most violated* $\mathbf{y}^t$ is essentially the solution to the optimization problem as $\max_{\mathbf{y} \in \mathcal{Y}} \boldsymbol{\alpha}' (\tilde{\mathbf{K}} \circ \mathbf{y} \mathbf{y}') \boldsymbol{\alpha}$. We employ a similar process as in [133] to solve this integer optimization problem by enumerating all possible candidates of $\mathbf{y}^t$. Rather than conducting the enumeration for all instances in positive and negative bags as in [133], we only need to enumerate the possible labeling candidates of the instances in positive bags in this work because all instances in the negative bags are assumed to be negative. In the experiments, we set the portion (*i.e.*, σ) of positive instances as 0.6, and we also enforce each bag to contain 15 instances in order to make this problem solvable.

3.3.2 Progressive MIL-CPB (PMIL-CPB)

In our TBIR task, we use the relevant training images to construct positive bags. However, the labels of relevant training web images are usually quite noisy, so the constraints on positive bags in our MIL-CPB algorithm may not always be satisfied. We therefore develop a new scheme called Progressive MIL-CPB to progressively select more confident positive bags.

Specifically, given a textual query q , we first uniformly partition the top-ranked relevant web images and the randomly sampled irrelevant web images into $2N_{\text{init}}$ clusters (*i.e.*, N_{init} positive bags and N_{init} negative bags). The top-ranked relevant web images can be randomly chosen from the images returned by the image search engine. For the *Flickr* images associated with noisy tags, we define the following ranking score [131] for each relevant image $\mathbf{x}$ to collect the top-ranked relevant images:

$$r(\mathbf{x}) = -\tau + \frac{1}{\delta}, \tag{3.7}$$

Algorithm 2 Progressive MIL-CPB (PMIL-CPB)

- 1: Initialize N_{init} positive bags and N_{init} negative bags for the given textual query q ;
 - 2: Train the MIL-CPB classifier and rank the remaining relevant images from the training data set;
 - 3: Construct N_{inc} positive bags from the top-ranked relevant training images and N_{inc} negative bags from the randomly sampled remaining irrelevant training images;
 - 4: Repeat steps 2 and 3 until the maximum number of iterations is reached.
-

where δ is the total number of tags in image $\mathbf{x}$, and τ is the rank position of the textual query q in the tag list of image $\mathbf{x}$.

The $2N_{\text{init}}$ bags are used as training data to learn an initial MIL-CPB classifier (see (3.5)). Then, we rank the relevant images in the training data set according to their decision values from the initial instance-level decision function, and we choose the $15N_{\text{inc}}$ top-ranked images to construct N_{inc} positive bags with each bag containing 15 instances. We additionally choose N_{inc} negative bags by randomly sampling $15N_{\text{inc}}$ irrelevant images. The $2N_{\text{inc}}$ new bags are added up to the training data to learn an updated MIL-CPB classifier. We iteratively partition the top-ranked training images from the current MIL-CPB classifier to construct more confident “bags” and then add these new “bags” to the training data to learn the subsequent MIL-CPB classifier until the maximum number of iterations is reached. Note that the instances that are already in the positive/negative training bags will not be selected in the subsequent iterations. The overall process is summarized in Algorithm 2.

3.4 Experiments

In this work, we first test our PMIL-CPB on the NUS-WIDE data set [141] for the large-scale image retrieval task, and then we compare PMIL-CPB with [8, 38] on the Google data set [134] for image re-ranking. We also report the results of MIL-CPB without conducting the progressive bag selection process discussed in Section 3.3.2.

3.4.1 Image retrieval on the NUS-WIDE data set

The NUS-WIDE data set [141] consists of 269,648 images collected from *Flickr.com* and their ground-truth annotations for 81 concepts. All images in the NUS-WIDE data set

are associated with noisy tags. Therefore, we represent each image by using its visual and textual features. Similar to [141], we extract three types of global visual features: Grid Color Moment (225 dim), Wavelet Texture (128 dim) and Edge Direction Histogram (73 dim). For each image, we further concatenate the three types of visual features to form a 426-dimensional feature vector. PCA is further employed to reduce the dimension of each feature vector by projecting it into a 119-dimensional visual feature space, which preserves 90% of the energy. We also extract the textual feature for each image from its associated tags. Specifically, we have removed the stop words (*e.g.*, “a”, “the”) and converted the remaining tags into their prototypes. We select the top-200 words with the highest frequency in the training set of NUS-WIDE as the vocabulary. And a 200-dimensional term-frequency feature is then extracted as the textual feature for each image. Above all, for each image, we further concatenate its visual feature $\mathbf{v}$ and its textual feature $\mathbf{t}$ to form the final feature vector representation $\mathbf{x}$, namely, $\mathbf{x} = [\lambda\mathbf{v}', \mathbf{t}']'$, where the parameter λ is empirically fixed as 0.1 in the experiments.

Considering that APR [27], EM-DD [28] and Citation kNN [29] are inefficient for this large-scale TBIR application, we focus on comparisons between our PMIL-CPB and the MIL methods mi-SVM¹ [11], Single Instance Learning SVM (SIL-SVM) [39], MILES [33], sMIL[39] and WsMIL [38]. Note that in SIL-SVM all the instances in positive (*resp.*, negative) bags are treated as positive (*resp.*, negative) samples. For MILES [33], we treat each test image as one test bag in this application, because it can only predict the labels of bags.

For mi-SVM, SIL-SVM, MILES, sMIL, WsMIL and MIL-CPB, N positive training bags are constructed by using the top-ranked relevant images decided according to the ranking scores from (3.7), and N negative training bags are formed by randomly sampling the irrelevant images, where we set $N = 5, 7, 10, 12, 15, 17, 20$ and 25 in the experiments. For our PMIL-CPB, we use the same strategy to construct N_{init} positive training bags as well as N_{init} negative training bags, where $N_{\text{init}} = 5$. With the progressive approach (see Section 3.3.2), we iteratively add N_{inc} positive and N_{inc} negative training bags, where N_{inc} is set as 2, 3 or 5 in order to fairly compare PMIL-CPB with other methods using

¹MI-SVM and mi-SVM achieve similar results in this work, so we only take mi-SVM as an example to report the results.

Table 3.1: MAPs (%) of all methods over 81 concepts on the NUS-WIDE data set. Each result in the table is the best among all results obtained by using different numbers (*i.e.*, N) of positive/negative training bags.

	SIL-SVM	mi-SVM	MILES	sMIL	WsMIL	MIL-CPB	PMIL-CPB
MAP	57.54	58.63	53.82	59.71	59.71	61.49	63.26

Figure 3.2: MAPs of all methods over 81 concepts on the NUS-WIDE data set with respect to different numbers (*i.e.*, N) of positive/negative bags.

the same number of training bags. Note that we fix the number of instances in each bag to 15 for all the methods. It is worth mentioning that *the ground-truth labels of the training images are not used* in the learning process for all methods.

For all methods, we train one-versus-all classifiers using the RBF kernel with the bandwidth parameter set as $1/A$, where A is the mean value of the square distances between training images. The training set consisting of 161,789 images. We evaluate the performances on the test set with 107,859 test images. For performance evaluation, we calculate the non-interpolated Average Precision (AP) [131] based on the 100 top-ranked images. We also define Mean Average Precision (MAP) as the mean of APs over all 81 concepts.

Results: For each method, Table 3.1 reports its best result among all results obtained by using different N . Fig. 3.2 shows the MAPs of all methods over 81 concepts using N positive and N negative training bags. From Fig. 3.2 and Table 3.1, we have the

following observations:

- 1) MILES achieves the worst MAP in this application possibly because it cannot effectively predict the labels of instances. SIL-SVM generally performs worse than mi-SVM except the case when setting $N = 12$. The explanation is that mi-SVM can cope with the labeling ambiguities on positive bags to some extent while in SIL-SVM all the training instances in positive bags are used as positive samples.
- 2) sMIL and WsMIL both outperform SIL-SVM and mi-SVM, which demonstrates that the MIL methods sMIL and WsMIL are more suitable for this application. Moreover, sMIL and WsMIL achieve very similar performances on this data set, which indicates that the iterative update of the weights for the instances in positive bags cannot lead to significant performance improvement for the large-scale TBIR task.
- 3) The performances of most MIL methods generally increase when using more training bags. The best MAP of MIL-CPB is better than those of sMIL and WsMIL, which demonstrates the effectiveness of MIL-CPB for exploiting the new MIL constraints on the positive training bags.
- 4) Our progressive approach PMIL-CPB consistently achieves the best results, and it quickly reaches the best MAP by using only 12 positive/negative training bags and then its performance becomes stable. It demonstrates that PMIL-CPB is an effective method for exploiting the new MIL constraints on positive bags, and it also shows that it is beneficial to employ the progressive bag selection scheme to construct high-quality positive training bags.

Purity comparison: We experimentally investigate the *quality* of the training instances from the positive bags in PMIL-CPB by using the ground truth annotations of these images. Note that in (3.5) the predicted label $\tilde{y}_i$ of each training instance $\mathbf{x}_i$ is obtained by $\tilde{y}_i = \sum_{t: \mathbf{y}^t \in \mathcal{C}} d_t y_i^t$. If $\tilde{y}_i > 0.5$, we predict $\mathbf{x}_i$ as a positive instance; otherwise, $\mathbf{x}_i$ is predicted as a negative instance. For each concept, we define purity $p = m'/m$ as the evaluation metric to analyze the quality of the added relevant images, where m is the number of training instances (referred to as *positively predicted training instances*) that are predicted as positive instances by PMIL-CPB² and m' is the number of truly positive

²In PMIL-CPB, the positively predicted training instances are ranked according to the bag orders of the positive training bags. Recall that we use the progressive approach to increase the number of training

Figure 3.3: Average purities of Init_Ranking and PMIL-CPB over 81 concepts on the NUS-WIDE data set using different numbers (*i.e.*, m) of positively predicted instances.

instances (according to the ground-truth annotations) among the m positively predicted training instances. For PMIL-CPB, we analyze the purity based on the classifier learned using 20 positive and 20 negative training bags.

For comparison, we also report the purity of the top-ranked m relevant training images from *Init_Ranking* in which the images are ranked according to the initial ranking scores from (3.7). In the experiment, we set $m = 50, 75, 100, 125, 150, 175$ and 200. The average purities of Init_Ranking and PMIL-CPB over 81 concepts on the NUS-WIDE data set are shown in Fig. 3.3. From Fig. 3.3, we observe that the average purity of PMIL-CPB is always much better than that of Init_Ranking, which demonstrates that PMIL-CPB can better cope with the noise in loose labels by effectively exploiting the new MIL constraints as well as using the progressive bag selection scheme. We also observe that the average purity of PMIL-CPB generally increases as the number of positively predicted instances increases. There are two explanations. On one hand, more confident “bags” that contain more pure positive instances are progressively added after each iteration. On the other hand, the learned MIL-CPB classifiers using more confident “bags” also become more robust. We have similar observations with different numbers (*i.e.*, N) of training positive/negative bags.

bags. Therefore, a bag that is added in an earlier iteration has a higher rank. The initial $N_{\text{init}} = 5$ positive bags are also ranked, because they are constructed by uniformly partitioning the top-ranked relevant images.

3.4.2 Image re-ranking on the Google data set

The Google data set [134] consists of the images returned by Google Image Search. In total, this data set has 7 categories with each having about 600 images on average. According to [134], on average there are 30% “good” images with clear view, 20% “ok” images with extensive occlusion and image noise, and 50% “junk” images that are unrelated to the category. As suggested in [38, 134], we also use four interest point detectors (*i.e.*, Kadir&Brady Saliency operator, Harris-Hessian detector, Difference of Gaussians (DoG) [142], and Edge-Laplace detector), and then we extract the SIFT descriptor [142] from each salient region. For each detector, we construct an independent codebook by clustering the descriptors from the training images with k -means, where we set $k = 200$. The final vocabulary consists of 800 visual words from the combined codebooks. Each image is finally represented as an 800 dimensional token frequency (tf) feature. Since the images in this data set are not associated with any textual descriptions, only the visual feature is used in both the training and testing stages.

In this work, we re-rank the images from the Google data set by using different methods including SIL-SVM, mi-SVM, MILES, MIL-CPB and our proposed method PMIL-CPB. We also report the existing image re-ranking results from [8, 38] on this data set. Following [8, 38], we treat the “ok” images as positive samples in the experiment because we believe the images with extensive occlusion and image noise are the challenging examples for evaluating different methods. For the given category, we construct each positive bag by randomly sampling 15 instances from this category, and each negative bag is obtained by randomly selecting 15 instances from other categories. For all methods, similarly as on the NUS-WIDE data set, we set $N = 5, 7, 10, 12, 15, 17$ and 20 (*i.e.*, $N_{\text{init}} = 5$ and $N_{\text{inc}} = 2, 3$ or 5 for PMIL-CPB). We also adopt the RBF kernel by empirically setting its bandwidth parameter as $0.25/A$, where A is the mean value of the square distances between training images. Following [8, 38], the per-category precision³ at 15% recall is used for performance evaluation.

In Table 3.2, we report the mean precision at 15% recall of each method over all 7 categories. We observe that MIL-CPB is better than SIL-SVM, mi-SVM and MILES.

³In this work, we follow the same setting as in [8, 38], and report the mean of precisions from five rounds of random positive/negative bag construction processes.

Table 3.2: Mean precisions (%) at 15% recall over 7 categories on the Google data set. Each result in the table is the best among all results obtained by using different numbers (*i.e.*, N) of positive/negative training bags.

	SIL-SVM	mi-SVM	MILES	MIL-CPB	PMIL-CPB
MP	74.45	75.81	74.89	85.61	86.64

Table 3.3: Per-category precisions (%) at 15% recall for 7 categories (*i.e.*, “Airplane”, “Cars rear”, “Face”, “Guitar”, “Leopard”, “Motorbike” and “Wrist watch”) on the Google data set. For better presentation, we denote the name of each category by using its first letter in this table.

	A	C	F	G	L	M	W	Mean
WsMIL [38]	100	81	57	52	66	79	95	75.71
Schroff’s [8]	58.5	–	–	70.0	49.6	74.8	98.1	70.20
PMIL-CPB	100	75.34	89.91	82.74	86.15	76.63	95.72	86.64

Moreover, PMIL-CPB outperforms MIL-CPB, which indicates that our proposed progressive bag selection scheme can automatically construct better positive bags to further improve the performance.

In Table 3.3, we compare the per-category precision and mean precision at 15% recall of the proposed method PMIL-CPB with the results reported in [8, 38]. Compared with the existing work [8, 38], our method PMIL-CPB enjoys significant performance improvements in 3 out of the 7 categories (*i.e.*, “face”, “guitar” and “leopard”), and also achieves the same or similar performances for some categories (*i.e.*, “airplane”, “motorbike” and “wrist watch”). The performance of PMIL-CPB is also much better when compared with [8] and [38] in terms of the mean precision at 15% recall.

3.5 Summary

In this chapter, we have proposed a new TBIR approach that can effectively exploit loosely labeled web images to learn robust SVM classifiers. First, we partition the relevant and irrelevant web images into clusters, and then we treat each cluster as a “bag” and the images in each bag as “instances”. Observing that each positive bag may contain at least a portion of positive instances, we formulate this task as an MIL problem with such constraints on positive bags. To predict the labels of instances (images), we present MIL-CPB algorithm. Moreover, we propose a novel progressive scheme called PMIL-CPB to

automatically and progressively select more confident positive bags, which leads to more robust classifiers. We conduct comprehensive experiments using the challenging NUS-WIDE data set and Google data set, and the results clearly demonstrate the effectiveness of our TBIR approach.

Chapter 4

Co-Labeling for Multiple View Weakly Labeled Learning Problems

In this chapter, we aim to effectively utilize multiple feature representations when learning from web data. With a similar formulation as MIL-CPB, we show that many traditional learning problems including multi-instance learning, semi-supervised learning, maximum margin clustering can be reformulated into a unified weakly labeled learning problem. Based on this perspective, we study a new learning problem called multi-view weakly labeled learning, in which we aim to develop a unified approach to learn robust classifiers by utilizing different types of weakly labeled multi-view data. We propose an effective approach called *co-labeling* to solve the multi-view weakly labeled learning problem.

Specifically, we model the learning problem on each view as a weakly labeled learning problem, in which we aim to learn an optimal classifier from a set of pseudo-label vectors generated by using the classifiers trained from other views. Unlike traditional co-training approaches that use a single pseudo-label vector for training each classifier, our co-labeling approach explores different strategies to handle the predictions from different views, biases and iterations for generating the pseudo-label vectors, which makes our approach more robust for real-world applications. We also analyze the generalization bound of our proposed co-labeling approach. Promising results for text-based image retrieval on the NUS-WIDE dataset as well as news classification and text categorization on several real-world multi-view datasets clearly demonstrate that our proposed co-labeling approach achieves state-of-the-art performance for various multi-view weakly labeled learning problems including multi-view SSL, multi-view MIL.

4.1 Co-Labeling for Multi-view Weakly Labeled Learning

In this section, we first review the existing work on multi-view learning and weakly labeled learning, and then present our co-labeling approach for the multi-view weakly labeled learning problem.

4.1.1 Background on Co-training

In multi-view learning, the training data are represented with multiple views of features. Typically, a classifier f^v (*e.g.* an SVM classifier) is trained on the v -th view and the final classifier is fused by using the classifiers from all views, *i.e.* $\tilde{f}(\mathbf{x}) = \frac{1}{V} \sum_{v=1}^V f^v(\mathbf{x}^v)$.

Co-training [15] was originally proposed for the SSL problem with two views. The basic idea of co-training is to iteratively add some pseudo-labeled samples into the pool of labeled training samples to re-train the classifiers on both views. The pseudo-labeled samples are selected from the pool of unlabeled training samples, and are labeled by at least one classifier which has a confident prediction. Finally, the classifiers from different views are fused to perform the classification.

While the original co-training algorithm feeds newly labeled training samples to each view, it can also be deemed as a learning process by iteratively updating the pseudo-labels of unlabeled data on each view [22]. We illustrate the co-training method in Fig. 4.1 (a). At each iteration, the classifiers on one view (*e.g.*, f^1 or f^2) generate pseudo-label vectors for learning the classifier on the other view (*e.g.*, f^2 or f^1) by using the supervised learning approach.

4.1.2 Background on Weakly Labeled Learning

To reduce human effort on labeling training data, various learning scenarios have been proposed in the literature to learn classifiers based on only weakly labeled data. For example, in SSL, one is given a limited number of labeled samples and a large amount of unlabeled samples. In MIL, the training data are given in the form of training bags, with each bag containing a certain number of training instances. While the label of each bag is given, the labels of instances inside each bag remain unknown.

Figure 4.1: Comparison between co-training and our co-labeling.

Recently, the work in [14] studied the weakly labeled learning problem by unifying the above learning scenarios into a general learning problem with weakly labeled data. The weakly labeled learning problem is formulated as the following optimization problem [14]:

$$\min_{f, \mathbf{y} \in \mathcal{Y}} \|f\|^2 + C\ell(f, \mathbf{y}), \quad (4.1)$$

where $\mathcal{Y}$ is the so-called *label candidate set*, f is the target classifier, and $\ell(\cdot)$ is the loss function. The label candidate set $\mathcal{Y}$ contains all the possible labelings of the training samples. Intuitively, the weakly labeled learning problem aims to learn an optimal classifier from all the possible labelings.

By defining different constraints on the label candidate set $\mathcal{Y}$, the weakly labeled learning problem in (4.1) unifies various traditional learning scenarios with different weakly labeled data. Let us denote the label vector as $\mathbf{y} = [y_1, \dots, y_n]'$ where $y_i \in \{+1, -1\}$ is the possible label for $\mathbf{x}_i$ and n is the number of training samples. We give several examples of the definition on the label candidate set $\mathcal{Y}$ corresponding to MIL and SSL.

As discussed in last chapter, in MIL, the constraints are that all instances in the negative bags are negative and at least one instance (or a portion of the instances) in each positive bag is positive [11, 41]. Let us denote $\mathcal{B}_I$ as the I -th training bag and Y_I as the corresponding given bag label. Then we can define the label candidate set as $\mathcal{Y} = \{\mathbf{y} | \sum_{i: \mathbf{x}_i \in \mathcal{B}_I} (y_i + 1)/2 \geq \varepsilon, \text{ if } Y_I = 1; y_i = -1, \text{ if } Y_I = -1\}$, where we have $\varepsilon = 1$ for the traditional MIL constraint [11] and $\varepsilon = \mu|\mathcal{B}_I|$ for the general MIL constraint [41] with μ being the portion parameter and $|\cdot|$ being the cardinality function.

In SSL [12], the training data are composed of n_l labeled samples and a large number of unlabeled samples. Usually, the unlabeled samples are required to satisfy a balance constraint. Then the label candidate set can be represented as $\mathcal{Y} = \{\mathbf{y} | y_i = g_i, i = 1, \dots, n_l, \sum_{i=n_l+1}^n y_i = \sigma\}$, where g_i is the ground truth label of $\mathbf{x}_i$, and σ is the parameter for the balance constraint.

4.1.3 Multi-view Weakly Labeled Learning

We observe that most traditional multi-view learning approaches are limited to semi-supervised learning, and the recently proposed weakly labeled learning works for semi-supervised learning and multi-instance learning are limited to single-view training data. In practice, traditional multi-view learning methods can benefit from the recent progress on weakly labeled learning, and also weakly labeled learning methods can be improved with multi-view information. Based on these observations and motivations, in this chapter we propose to study a new learning problem called multi-view weakly labeled learning which aims to solve the weakly labeled learning problem with multi-view training data.

Specifically, we focus on the binary classification problem in this work. The multi-class classification problem can be converted into a set of binary classification problems using the one-versus-all strategy. Inspired by the work on co-training and the weakly labeled learning, we formulate the multi-view weakly labeled learning problem as follows:

$$\min_{f^v, \mathbf{y}^v \in \mathcal{C}^v} \sum_{v=1}^V \|f^v\|^2 + C\ell(f^v, \mathbf{y}^v), \quad (4.2)$$

where $\mathbf{y}^v$ is the pseudo-label vector for the training samples on the v -th view, $\mathcal{C}^v$ is the set of possible pseudo-label vectors for the v -th view generated by using the predictions

from other views. Similar to co-training, those V classifiers in (4.2) are not individually learnt. Specifically, we use the classifier trained on one view to help the training process on the other views through the pseudo-label vector sets (see the details below). Finally those V classifiers are equally fused for prediction as in co-training.

To solve the multi-view weakly labeled learning problem in (4.2), we propose a novel approach called “co-labeling” by tackling two major challenges: how to effectively use the classifier trained on one view to help the training process on other views, and how to improve the weakly labeled learning on each view for training a more robust classifier.

We use the pseudo-label vector set $\mathcal{C}^v$ to exchange information from the classifier trained on one view to another. Specifically, similar to co-training, in our co-labeling approach the pseudo-label vector set $\mathcal{C}^v$ is generated by using the predictions from classifiers trained on the other views. To cope with different weakly labeled data, we first propose a projection operator, which converts the predictions (*i.e.*, the decision values) to pseudo-label vectors by considering the constraints associated with different weakly labeled learning scenarios. Moreover, different from traditional co-training methods, in which the classifiers exchange information through a single pseudo-label vector, we use a set of pseudo-label vectors in (4.2) which contain more information for learning a robust classifier. Those pseudo-label vectors can be generated from different views, biases and iterations (see Section 4.2 for details).

To learn a classifier on each view, the traditional co-training algorithm adopted the supervised learning method by using single pseudo-label vector as the ground-truth of unlabeled data, which may be sensitive to noise in the pseudo-label vector. To overcome this problem, in our co-labeling we treat the learning problem on each view as a weakly labeled learning problem, by learning an optimal classifier from a set of pseudo-label vectors (see Fig. 4.1). We show that it can be formulated as an MKL problem similarly as in MIL-CPB (see Section 4.3).

We list our co-labeling algorithm in Algorithm 3, in which we iteratively update the set of pseudo-label vectors and train a classifier on each view by solving a weakly labeled learning problem. The details of generating the pseudo-label vector set $\mathcal{C}^v$ for each view are introduced in Section 4.2, and the Multiple Kernel Learning (MKL) algorithm for solving the weakly labeled learning problem on each view is presented in Section 4.3.

Algorithm 3 The Co-Labeling Algorithm

-
- 1: Initialize the pseudo-label vector set $\mathcal{C}^v$ for each view.
 - 2: **repeat**
 - 3: Train a classifier f^v based on $\mathcal{C}^v$ by solving a weakly labeled learning problem for each view (see Sec. 4.3).
 - 4: Obtain the predictions $\mathbf{z}_v$ of training samples using f^v for each view.
 - 5: Update the pseudo-label vector set $\mathcal{C}^v$ using $\mathbf{z}_j$'s ($j \neq v$).
 - 6: **until** The stopping criterion is reached.
 - 7: **return** f^v 's.
-

4.2 Generate the Pseudo-label Vector Set

How to generate a set of pseudo-label vectors on one view using the information from the other views is the first key issue of our co-labeling approach. Inspired by co-training [15], we generate the pseudo-label vectors on one view by using the predictions (*i.e.*, decision values) from the classifiers on other views. Unlike the co-training method which uses only a single pseudo-label vector, we need to consider several issues as described below to better handle the general multi-view weakly labeled learning scenarios and train a more robust classifier.

4.2.1 Handling General Weakly Labeled Constraints

To cope with general weakly labeled learning scenarios, we need to find a projection operator $\mathbf{y} = \pi(\mathbf{z})$, which converts a prediction $\mathbf{z}$ (*i.e.*, the vector of decision values) to a pseudo-label vector $\mathbf{y}$ by considering the constraints associated with different weakly labeled learning scenarios. Formally, the projection operator can be defined as follows:

$$\pi(\mathbf{z}) = \arg \min_{\mathbf{y} \in \mathcal{Y}} \|\mathbf{y} - \mathbf{z}\|, \quad (4.3)$$

where $\|\cdot\|$ is the ℓ_2 -norm. Since the weakly labeled data are usually associated with linear constraints (see Section 4.1.2), the problem (4.3) can be solved efficiently by using the sorting and thresholding operators.

Specifically, for training samples with known labels (*e.g.*, the labeled samples in SSL and the instances in negative bags in MIL), we directly assign their ground-truth labels. For the weakly labeled samples, we use the thresholding function to convert their decision values into binary values (*i.e.* +1 or -1).

For example, in SSL, the weakly labeled training samples are the unlabeled samples, which are associated with a balance constraint $\sum_{i=n_l+1}^n y_i = \sigma$. So we can sort the decision values of these unlabeled samples in descending order, and adjust the threshold to assign the first $(\sigma + n - n_l)/2$ samples to be positive and the remaining ones to be negative. In MIL, the ambiguous training samples are the positive instances. We sort the instances in each positive bag, and obtain the pseudo-label vectors according to the sign of decision values (*i.e.* the threshold is set to zero). If the labeling for any positive bag does not satisfy the constraint (*i.e.* at least ϵ instances should be positive) as in Section 4.1.2, we then assign the top ϵ instances to be positive. The projection operator for other weakly labeled training data with linear constraints can also be similarly defined.

4.2.2 Handling the Bias of Weakly Labeled Learning

Moreover, the learned classifiers in the weakly labeled learning problems can be easily biased. For example, in MIL one common approach is to initialize all the instances in positive bags as positive training samples, so it is more likely that the initial classifier will predict the negative samples to be positive. Moreover, in SSL the limited number of labeled training data may also easily bias the classifier. A possible solution is to adjust the bias term of the learnt classifier. However, it is a nontrivial task since we do not have the ground truth labels for adjustment. Instead of explicitly adjusting the bias term, we simply perturb the predictions from the learnt classifier by using different biases, which is motivated by the recent work on domain adaptation [143].

Formally, given any prediction $\mathbf{z}$, we can obtain a set of perturbed predictions as $\{\tilde{\mathbf{z}}_s |_{s=1}^S\}$ by using different predefined biases, *i.e.* $\tilde{\mathbf{z}}_s = \mathbf{z} + b_s$, where S is the number of biases and $b_s \in \mathbb{R}$ is a bias term to adjust the predictions. After that, we generate a set of pseudo-label vectors using the projection operator defined as in (4.3) on these perturbed predictions, and use the weakly labeled learning method (*i.e.*, the MKL method in Section 4.3) to determine the optimal biases during the learning process.

4.2.3 Combining the Pseudo-label Vector Set from Previous Iterations

Thus far, we only consider the pseudo-label vectors obtained by using the predictions from the latest iteration, which means the pseudo-label vector set on each view can be

Algorithm 4 Algorithm for Generating Pseudo-label Vectors

Require: The current pseudo-label vector set $\mathcal{C}_t^v$, and the decision values from the classifiers on different views at the t -th iteration $\mathbf{z}_{v,t}$, for $v = 1, \dots, V$.

```

1: for  $v = 1, \dots, V$  do
2:   for  $s = 1, \dots, S$  do
3:     Obtain the perturbed prediction with the  $s$ -th bias:  $\tilde{\mathbf{z}}_{v,s,t} = \mathbf{z}_{v,t} + b_s$ .
4:     Project the perturbed prediction into the pseudo-label vector:  $\mathbf{y}_{v,s,t} = \pi(\tilde{\mathbf{z}}_{v,s,t})$ 
5:   end for
6:    $\mathcal{O}_{v,t} = \{\mathbf{y}_{v,s,t} |_{s=1}^S\}$ 
7: end for
8: for  $v = 1, \dots, V$  do
9:   Obtain the pseudo-label vector set on the  $v$ -th view:  $\mathcal{C}_{t+1}^v = \mathcal{C}_t^v \cup \{\mathcal{O}_{j,t} |_{j=1, j \neq v}^V\}$ 
10: end for
    
```

Ensure: The pseudo-label vector set on each view: $\mathcal{C}_{t+1}^v$ for $v = 1, \dots, V$.

changed at different iterations. As a result, one potential problem is that the algorithm may not converge.

Inspired by the recent weakly labeled learning works [13, 14, 46], we construct the pseudo-label vector set by using the pseudo-label vectors obtained from all previous iterations. In other words, at each iteration we augment the pseudo-label vector set by appending the newly obtained pseudo-label vectors into the previous pseudo-label vector set. Since the feasible set $\mathcal{C}^v$ is enlarged at each iteration (*i.e.*, $\mathcal{C}_{t+1}^v \supseteq \mathcal{C}_t^v$), the feasible set of the optimization problem in (4.2) is gradually enlarged as the number of iterations increases. Therefore, in the worst case the optimal objective value of the problem in (4.2) should be at least as low as that at the previous iteration. As a result, the objective of (4.2) will monotonically decrease as the number of iterations increase.

4.2.4 Summary and Discussion

We illustrate the entire process for generating the pseudo-label vectors in Algorithm 4. We first perturb the prediction $\mathbf{z}_{v,t}$ at the t -th iteration from the v -th view with S predefined biases and obtain a set of perturbed predictions $\{\tilde{\mathbf{z}}_{v,s,t} |_{s=1}^S\}$. Then, we use the projection operator to convert each $\tilde{\mathbf{z}}_{v,s,t}$ to a pseudo-label vector $\mathbf{y}_{v,s,t}$. By defining $\mathcal{O}_{v,t} = \{\mathbf{y}_{v,s,t} |_{s=1}^S\}$ as the pseudo-label vectors generated from the v -th classifier at the t -th iteration, the pseudo-label vector set for training the classifier on the v -th view at

the next iteration can be obtained by combining the pseudo-label vectors from all other views, different biases and all the previous iterations, *i.e.*, $\mathcal{C}_{t+1}^v = \mathcal{C}_t^v \cup \{\mathcal{O}_{j,t}|_{j=1, j \neq v}^V\}$.

4.3 The Implementation of Co-Labeling

After generating a pseudo-label vector set for each view, the remaining problem is to learn a robust classifier using the pseudo-label vector set. In this section, we formulate the weakly labeled learning problem on each view as an MKL problem by combining each pseudo-label vector with the input kernel.

4.3.1 The Formulation

We adopt the maximum margin classifier $f(\mathbf{x}) = \mathbf{w}'\psi(\mathbf{x}) + b$ where $\psi(\cdot)$ is the feature mapping function, and use ρ -SVM with the squared hinge loss¹ to solve the weakly labeled learning problem. Based on (4.1), we arrive at the following optimization problem for the v -th view:

$$\begin{aligned} \min_{\mathbf{y}^v \in \mathcal{Y}^v} \min_{\mathbf{w}^v, b^v, \rho^v, \xi_i} \quad & \frac{1}{2} \left(\|\mathbf{w}^v\|^2 + b^{v^2} + C \sum_{i=1}^n \xi_i^2 \right) - \rho^v, \\ \text{s.t.} \quad & y_i^v (\mathbf{w}^{v'} \psi(\mathbf{x}_i^v) + b^v) \geq \rho^v - \xi_i, \quad i = 1, \dots, n, \end{aligned} \quad (4.4)$$

where C is the tradeoff parameter, $2\rho^v/\|\mathbf{w}^v\|$ defines the margin and y_i is the i -th element of the label vector $\mathbf{y}$. The superscript v denotes the v -th view, in other words, we shall optimize the above objective function on each view. For simplification, we omit the superscript v unless necessary.

4.3.2 The MKL solution

By introducing the dual variable $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]'$ for the constraints in (4.4), we derive the dual problem of (4.4) as:

$$\min_{\mathbf{y} \in \mathcal{Y}} \max_{\boldsymbol{\alpha} \in \mathbf{A}} -\frac{1}{2} \boldsymbol{\alpha}' \left(\mathbf{K} \circ \mathbf{y} \mathbf{y}' + \frac{1}{C} \mathbf{I} \right) \boldsymbol{\alpha}, \quad (4.5)$$

¹It has been suggested in [57] that the squared hinge loss is more robust than the hinge loss for solving the MMC problem.

where $\mathbf{K} = \hat{\mathbf{K}} + \mathbf{1}\mathbf{1}'$, $\hat{\mathbf{K}} = [k(\mathbf{x}_i, \mathbf{x}_j)] = [\psi(\mathbf{x}_i)' \psi(\mathbf{x}_j)]$ is the kernel matrix, and $\mathbf{A} = \{\boldsymbol{\alpha} | \boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\alpha}'\mathbf{1} = 1\}$ is the feasible set of $\boldsymbol{\alpha}$.

Instead of directly solving the mixed-integer problem in (4.5), we seek to optimize the linear combination of $\mathbf{y}\mathbf{y}'$'s similarly as in MIL-CPB. Then, the above problem is relaxed to an MKL problem which is a lower bound of (4.5). The relaxed formulation is as follows:

$$\min_{\mathbf{d} \in \mathcal{D}} \max_{\boldsymbol{\alpha} \in \mathbf{A}} -\frac{1}{2} \boldsymbol{\alpha}' \left(\sum_{m=1}^M d_m \mathbf{K} \circ \mathbf{y}_m \mathbf{y}_m' + \frac{1}{C} \mathbf{I} \right) \boldsymbol{\alpha}, \quad (4.6)$$

where $\mathcal{Y}$ is the pseudo-label vector set on each view, $M = |\mathcal{Y}|$ is the cardinality of the pseudo-label vector set $\mathcal{Y}$, $\mathbf{d} \in \mathbb{R}^M$ is a vector of combination coefficients for the base kernels $\mathbf{K} \circ \mathbf{y}_m \mathbf{y}_m'$'s, and $\mathcal{D} = \{\mathbf{d} | \mathbf{d} \geq \mathbf{0}, \mathbf{d}'\mathbf{1} = 1\}$ is the feasible set of $\mathbf{d}$. Although similar relaxations have also been exploited in our MIL-CPB [41] as well as other works on different weakly labeled learning scenarios [13, 14, 46], we would highlight that the pseudo-label vector sets in those single-view methods were obtained by iteratively finding the most violated constraint, which is another NP-hard problem. In contrast, the pseudo-label vector set in our co-labeling algorithm is obtained by using the predictions from other views, which is very efficient.

We can also write the primal form of (4.6) as follows,

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

where d_m is the m -th entry of the combination coefficient vector $\mathbf{d}$, y_i^m is the i -th entry of the pseudo-label vector $\mathbf{y}_m$, and $\phi(\cdot)$ is the nonlinear feature mapping induced by $\mathbf{K} = \hat{\mathbf{K}} + \mathbf{1}\mathbf{1}'$. The above problem can be efficiently optimized by using existing MKL solvers [72], and the global optimal solutions to $\mathbf{d}$ and $\mathbf{w}_m$'s are guaranteed. After solving the MKL on each view, the final classifier is given by:

$$f(\mathbf{x}) = \sum_{v=1}^V f^v(\mathbf{x}^v) = \sum_{v=1}^V \frac{1}{\rho_v} \left(\sum_{i=1}^n \alpha_i^v \sum_{t=1}^{|\mathcal{Y}^v|} d_t^v y_{t,i}^v (k(\mathbf{x}_i^v, \mathbf{x}^v) + 1) \right).$$

It is worth noting that the prediction is as fast as the SVM prediction on each view.

Algorithm 5 The Algorithm of Co-Labeling with MKL**Require:** Training samples with V views, initial labels $\mathbf{y}_0$.**Ensure:** Classifier variables $\mathbf{d}^v$'s, $\boldsymbol{\alpha}^v$'s and label candidate sets $\mathcal{Y}^v$'s.

- 1: Initialize the label candidates set $\mathcal{Y}^v = \{\mathbf{y}_0\}$ for each view.
- 2: **repeat**
- 3: Solve $\boldsymbol{\alpha}^v$ and $\mathbf{d}^v$ in (4.6) based on $\mathcal{Y}^v$.
- 4: Get the decision values of training data $\mathbf{z}^v$ using $\boldsymbol{\alpha}^v$ and $\mathbf{d}^v$.
- 5: Vary the biases to obtain a set of decision values $\mathbf{Z}^v$ for each view.
- 6: Get the label candidate set $\mathbf{O}^v$ by enforce each prediction in $\mathbf{Z}^v$ to satisfy the constraints.
- 7: Set each $\mathcal{Y}^v = \mathcal{Y}^v \cup \mathbf{O}^p$ for any $p \neq v$.
- 8: **until** The stop criterion is reached.
- 9: **return** $\boldsymbol{\alpha}^v$'s, $\mathbf{d}^v$'s and $\mathcal{Y}^v$'s.

4.3.3 The Algorithm

We summarize our detailed algorithm in Algorithm 5. The initial labels $\mathbf{y}_0$ are problem dependent. For example, in SSL we can use the prediction of the classifier trained on the labeled data and in MIL we usually initialize all instances in positive bags as positive and all instances in negative bags as negative. After the initialization, the classifier will be trained on each view by solving (4.6). Then we use the classifier on each view to predict the labels of training samples, and obtain a set of predictions by varying the bias. All the predictions are required to satisfy the constraints (*i.e.*, bag constraints in MIL or balance constraint in SSL). Finally the label candidate set of each view is augmented by merging the predictions from the other views. This process is repeated until the stop criterion is reached.

Stop Criterion: One observation is that the objective value in Algorithm 5 decreases monotonously on each view. The reason is as follows: We solve an MKL problem which minimizes the objective function in (4.6) with respect to $\boldsymbol{\alpha}^v$ and $\mathbf{d}^v$. Since the pseudo-label vector set $\mathcal{Y}^v$ is augmented at each iteration, in the worst case the optimal solution of MKL at the current iteration should be the same as that at the last iteration by setting the entries in the coefficient vector $\mathbf{d}^v$ corresponding to the newly added label candidates to zeros. Therefore the objective values of MKL on each view should decrease monotonically as the number of iterations increases. Experimental results of convergence are presented in Section 4.6.2.4. So the stop criterion in the co-labeling is that the

difference of the objective values between two iterations is less than a small value on all views or the maximum number of iterations is reached. Usually, we observe that the algorithm runs fast and stops in fewer than 10 iterations.

Time Complexity: The main cost in Algorithm 5 is the training process of MKL and the testing process on the training samples. Since the kernel was computed in the training process, the testing process is only a matrix multiplication operator and the cost can be ignored compared with the training process. Considering the non-linear case, let us denote the complexity of training an MKL as $O(\text{MKL})$ ² and suppose our algorithm runs T iterations, then the total time complexity of Algorithm 5 is $T \cdot V \cdot O(\text{MKL})$ where V is the total number of views. For the linear case, there exists a fast algorithm to solve the SVM problem in the primal form, so our co-labeling can be much faster than in the non-linear case.

4.4 Generalization Bound Analysis

Now we analyze the generalization bound of our co-labeling algorithm. We first give the generalization bound of our weakly labeled learning method on each view, and then present the generalization bound for the final classifier.

At each iteration of our co-labeling method, we train an MKL classifier f^v on each view by using (4.7) based on the current pseudo-label vectors. The learnt MKL classifier can be written as $f(\mathbf{x}) = \sum_{m=1}^M d_m f_m(\mathbf{x})$ with $f_m(\mathbf{x}) = \sum_i \alpha_i y_i^m \phi(\mathbf{x}_i)' \phi(\mathbf{x})$, where d_m , y_i^m and $\phi(\cdot)$ are as defined in (4.7). We denote the empirical loss for the weakly labeled learning problem in (4.7) as $\hat{\ell}(f) = \frac{1}{n} \sum_{i=1}^n \xi_i^2$. We also assume that both the decision value of each base classifier and the squared hinge loss of f can be bounded as $|f_m(\mathbf{x}_i)|^2 \leq U$ and $\xi_i^2 \leq U$ for some fixed U . Then, we obtain the generalization bound as follows:

Theorem 4.1 *For fixed $B, U > 0$, $0 < \delta < 1$, and $\mathcal{F} = \{f \mid \|f\|_{\mathcal{H}}^2 \leq B^2\}$, with probability*

²The time complexity of MKL has not been theoretically analyzed. Usually, the MKL solver needs to train an SVM for tens of iterations. The empirical analysis shows that optimizing the QP problem in SVM is $O(n^{2.3})$ where n is the number of training samples. Therefore, the complexity of MKL is $O(kn^{2.3})$ where k is the number of iterations in MKL.

at least $1 - \delta$, every function $f \in \mathcal{F}$ satisfies

$$\begin{aligned} \text{err}(f) \leq & \frac{1}{\rho^2} \hat{\ell}(f) + \frac{8U \|\mathbf{d}\|_1}{n\rho^2} \sum_{m=1}^M d_m h(\mathbf{g}, \mathbf{y}_m) \\ & + \frac{8B}{n\rho} \sqrt{\text{tr}(\mathbf{K})} + 3\sqrt{\frac{\ln(2/\delta)}{2n}}, \end{aligned} \quad (4.8)$$

where $\text{err}(f)$ is the expectation of error on the testing samples, $h(\mathbf{g}, \mathbf{y}_m)$ is the Hamming distance between the pseudo-label vector $\mathbf{y}_m$ and the ground truth labeling $\mathbf{g}$ (i.e., $h(\mathbf{g}, \mathbf{y}_m) = \sum_{i=1}^n I(g_i \neq y_i^m)$ with $I(\cdot)$ being the indicator function.), ρ is defined as in (4.4), and $\text{tr}(\mathbf{K})$ is the trace of $\mathbf{K}$.

The proof is based on the generalization bound of SVM [144]. Compared with the standard SVM, the challenge is that the ground truth labeling $\mathbf{g}$ is unknown in our weakly labeled learning scenario. In Theorem 4.1, we bound the ground truth error of f by using the empirical error $\hat{\ell}(f)$ and the linear combination of the Hamming distance between $\mathbf{y}_m$ and the ground-truth labeling $\mathbf{g}$, where the combination coefficients d_m 's are learnt by MKL. With such a linear combination, the second term on the righthand side of (4.8) will not be large, and thus the bound is reasonably tight. We give a detailed proof and explanations in the Appendix.

When our co-labeling algorithm terminates, we obtain an MKL classifier f^v on each view for $v = 1, \dots, V$. The final classifier is obtained by using the average of the decision values from all views, i.e., $\tilde{f}(\mathbf{x}) = \frac{1}{V} \sum_{v=1}^V \frac{1}{\rho^v} f^v(\mathbf{x}^v)$. Based on Theorem 4.1, we give the generalization bound for the final classifier $\tilde{f}(\mathbf{x})$ as follows:

$$\begin{aligned} \text{err}(\tilde{f}) \leq & \frac{1}{V} \sum_{v=1}^V \frac{1}{(\rho^v)^2} \left(\hat{\ell}(f^v) + \frac{8U \|\mathbf{d}^v\|_1}{n} \sum_{m=1}^M d_m^v h(\mathbf{g}, \mathbf{y}_m^v) \right) \\ & + \frac{8B}{nV} \sum_{v=1}^V \frac{1}{\rho^v} \sqrt{\text{tr}(\mathbf{K}^v)} + 3\sqrt{\frac{\ln(2/\delta)}{2n}}, \end{aligned} \quad (4.9)$$

where the superscript v in the variables $\mathbf{d}^v$, d_m^v , $\mathbf{y}_m^v$, ρ^v denotes that the corresponding variables are on the v^{th} view. We also provide a detailed proof in the Appendix.

4.5 Related Work

Our work is related to traditional multi-view learning approaches. Most traditional multi-view learning methods were proposed for semi-supervised learning (SSL). One of the pioneering works is the co-training method [15], which was originally proposed for semi-supervised learning problems with two views. It was further extended to co-EM [16], in which they label all the unlabeled data at each iteration without considering confidence. It was also extended by using SVMs as base classifiers in [23], which was subsequently adapted for unsupervised learning [22]. Co-training was also extended to tri-training [74] and co-forest [75] to handle more than two views. However, the co-training style algorithms work under strict assumptions that each view is sufficient to train a low-error classifier and both views are conditionally independent, which might not be satisfied on real world datasets [76]. Many works attempted to relax those assumptions from various perspectives, such as weak dependence [77], α -expansion [78], large diversity [76] and label propagation [79]. Recently, co-training with insufficient views has also been theoretical analyzed in [80].

Besides co-training style methods, other methods such as co-regularization based approaches [1, 17–19] were also proposed to train classifiers on different views based on a so-called *co-regularization* criterion, which is used to minimize the differences of the decision values from the classifiers on different views. Similar ideas have also been employed in multi-view clustering [20, 21]. However, similar to co-training style methods, all these methods are specifically designed for a specific learning scenario, thus they cannot be directly applied to handle general multi-view weakly labeled data.

Our work is also related to various learning scenarios with different types of weakly labeled training data. Specifically, besides the above mentioned multi-view SSL works, SSL methods have also been widely studied for single view training data [12, 13]. A comprehensive survey on SSL can be found in [51]. MIL is another widely studied learning scenario with weakly labeled training data, in which the training data is provided in the form of bags of instances. Only the labels of training bags are given, and the labels of instances inside each training bag are unknown. Many works have been proposed to solve the MIL problems in the literature [11, 39, 41, 43, 47], and a survey on MIL can be found in [48]. Another example is maximum margin clustering (MMC) [46, 56], in which the goal is

to learn a discriminative classifier to partition unlabeled training samples into two disjoint clusters. Recent work [14] uniformly referred to the above learning scenarios as weakly labeled learning, and proposed a unified scheme called WellSVM to solve it. Another work which can handle different types of weakly labeled data was also proposed in [58]. Other learning scenarios related to weakly labeled data include relative outlier detection (ROD) [59–61] and multi-instance semi-supervised learning (MISSL) [63]. However, all those works were proposed for only single-view training data. In contrast, in this work, we study a new learning problem called multi-view weakly labeled learning, in which we further explore the multi-view information of different types of weakly labeled training data to learn more robust classifiers, and our proposed co-labeling approach is applicable for not only multi-view SSL, but also other tasks such as multi-view MIL.

4.6 Experiments

In this section, we evaluate our co-labeling approach on five real-world datasets for two cases, multi-view multiple instance learning and multi-view semi-supervised learning.

4.6.1 Multi-View Multiple Instance Learning

MIL has been successfully exploited in Text-Based Image Retrieval (TBIR), so we also evaluate our co-labeling approach for TBIR under the multiple view setting. Similar to Chapter 3, we conduct the experiment on the large-scale NUS-WIDE data set, which consists of 269,648 images from 81 annotated concepts collected from the website *Flickr.com*. Three groups of features are extracted:

- 1) **The textual feature** is extracted from the associated tags for each image. The vocabulary is constructed using 1000 words with the highest frequency. Then, a 1000 dimensional term-frequency feature is extracted for each image.
- 2) **The global visual features** are extracted according to the procedure in Chapter 3, then we concatenate three types of global features and apply PCA to obtain 119-d features.

Table 4.1: Mean Average Precisions (MAPs) over 81 concepts from different methods on the NUS-WIDE data set.

	TG	TL	TG+TL
MIL-CPB	61.43	57.84	77.07
mi-SVM	59.25	59.26	77.18
sMIL	60.01	62.09	75.48
Co-Labeling	62.56	61.71	79.09

- 3) **The local visual feature** is based on the SIFT feature by preserving LCC coding, which is extracted for each image using the method in [145]. We train a codebook with 4096 visual words and use a three-level spatial pyramid, and finally we get an 86016 dimensional sparse vector for each image.

Two views are constructed by using these three types of features. Specifically, we partition the 1000-d textual feature into two 500-d vectors, $\mathbf{t}_1$ and $\mathbf{t}_2$, with even dimensions and odd dimensions respectively. Then we further concatenate the two textual vectors with the global visual feature vector and local visual feature vector to form the final feature vectors for two views with the form $\mathbf{x} = [\gamma\mathbf{t}; \lambda\mathbf{v}]$, which are denote as TG view and TL view, respectively. For the TG view, following Chapter 3, we set $\gamma = 1$ and $\lambda = 0.1$. For TL view, we empirically set $\gamma = 0.1$ and $\lambda = 0.9$.

We compare our co-labeling approach with three MIL methods, MIL-CPB [146], mi-SVM [11] and sMIL [39] which have the best performances as studied in Chapter 3. Since they are single-view methods, we use the late fusion method to average the decision values of the classifiers from different views ³. For all methods, we construct 25 positive bags using the top relevant images and 25 negative bags using randomly selected irrelevant images, with each bag containing 15 instances. A Gaussian kernel is used on the TG view and a linear kernel is used on the TL view. A binary classifier is trained for each concept, and the top-100 Mean Average Precisions (MAPs) are reported in the experiments.

The MAPs over 81 concepts for different methods on two views as well as the results after using late fusion on the NUS-WIDE data set are reported in Table 4.1. We have the following observations:

³We also tried using early fusion (the average kernel) for these methods. However, the results are worse than those of late fusion.

- Our co-labeling approach achieves the best late fusion result, which demonstrates the effectiveness of the proposed method. The improvement over the second best is 1.9% in terms of MAP over 81 concepts.
- The performances of baseline methods on the TG view are consistent with the reported results in Chapter 3, and our method also achieves the best result.
- On the TL view, we observe that the instance-based methods MIL-CPB and mi-SVM are worse than the bag-based method sMIL. A possible explanation is the labels inferred in the learning process are quite noisy because the data cannot be well separated by using a linear classifier. However, the co-labeling still obtains a good result as it considers not only the input TL features but also uses the outputs from the TG view.

Moreover, the late fusion result of sMIL is worse than other methods, although it has good performance on each view. It is more likely the classifiers from two views are not as complementary to each other as in other methods. A possible reason is that the bag-level MIL methods cannot effectively use all information on training instances [146]. It also demonstrates the advantage of multi-view methods which can effectively use the information from different views.

4.6.2 Multi-View Semi-Supervised Learning

4.6.2.1 Data sets

We evaluate our co-labeling approach for semi-supervised learning on two applications, news classification and webpage classification. The BBC data set and BBCSport data set are used for news classification; and the WebKB data set is used for webpage classification. The details of these data sets are summarized in Table 4.2 and described as follows:

The BBC and BBCSport data sets: The two data sets [147] contain news articles collected from the BBC⁴. The BBC data set includes 2225 documents from five topics (business, entertainment, politics, sports and technology) and the BBCSport data set includes 737 sports news documents from five classes (athletics, cricket, football, rugby and tennis). We randomly segment the feature into two views as in [20]. We partition

⁴The features can be downloaded from <http://mlg.ucd.ie/datasets/bbc.html>

Table 4.2: Summarization of the data sets used in SSL. $d1$ and $d2$ are the feature dimensions of two views. $\#c$, $\#l$, $\#u$ and $\#t$ are the numbers of classes, labeled data, unlabeled data and test data, respectively.

Data sets	$d1$	$d2$	$\#c$	$\#l$	$\#u$	$\#t$
BBC	4817	4818	5	10	1104	1111
BBCSport	2306	2307	5	10	360	367
WebKB	3000	1840	2	12	1039	

the data sets into the training set and the test set, each of which contains 50% of the documents per class. Ten labeled data are further selected from the training set.

The WebKB data set: The WebKB data set has been widely used to evaluate multi-view methods [1, 15, 148]. It contains 1051 web pages belonging to two categories: course or non-course. For each web page, there are two views, the page view which contains the textual content of this page, and the link view which contains the anchor-text on links from other webpages pointing to this page. 3000-d features are extracted for the page view and 1840-d features for the link view, respectively. 12 samples are selected as the labeled data and the remaining samples are used as unlabeled data⁵.

4.6.2.2 Experimental Setting

We compare our co-labeling with following baselines:

- SVM: The standard SVM trained with the labeled data, which is a commonly used baseline in semi-supervised learning.
- TSVM [54]: The transductive SVM trained with the labeled and unlabeled data. The two views are later fused to obtain the final result.
- Co-LapSVM [1]: The Laplacian SVM in the multi-view setting.
- 2V-TSVM [148]: The TSVM version of co-regularization, in which both ramp loss and co-regularization are used to cope with the two-view setting.

⁵The indices of labeled data and features are publicly available in <http://people.cs.uchicago.edu/~vikass/manifoldregularization.html>

Table 4.3: Performances of different methods on three data sets. The best results are denoted in boldface. MAPs and standard deviations are reported on BBC&BBCSport. $\uparrow$ denotes the result is significantly better than the others according to the t-test with a significance level at 0.05. PRBEPs are reported on WebKB. The results of SVM, TSVM and Co-LapSVM are from Table 1 in [1]. Since the standard deviations of these methods are not available, we only report the average PRBEPs on WebKB.

	BBC			BBCSport			WebKB		
	View1	View2	View1+2	View1	View2	View1+2	page	link	page+link
SVM	66.53(4.08)	63.11(3.67)	74.26(3.37)	70.69(3.42)	66.43(3.98)	76.99(3.76)	74.4	77.8	84.4
TSVM	71.99(5.48)	66.83(3.54)	75.72(3.16)	74.62(5.73)	65.51(3.36)	79.21(6.43)	85.5	91.4	92.2
Co-LapSVM	70.30(3.39)	68.04(4.56)	76.97(3.41)	70.70(3.43)	66.43(3.99)	77.14(3.29)	94.3	93.3	94.2
2V-TSVM	52.70(3.96)	52.61(5.34)	58.39(5.25)	64.00(3.08)	63.50(3.86)	69.82(3.78)	85.7	86.7	87.3
PMC	—	—	71.57(6.37)	—	—	79.48(5.41)	—	—	88.6
Co-Labeling	78.41(3.79) $\uparrow$	77.61(3.01) $\uparrow$	81.37(3.14) $\uparrow$	82.10(5.41) $\uparrow$	79.60(4.44) $\uparrow$	84.22(5.11) $\uparrow$	92.5	93.1	95.1

- PMC [149]: An improved version of co-training which is designed to split the single view features into two views. We apply PMC on the joint-view of these three data sets. Due to the randomness in initializing the feature splits, we run each split five times and use the split with the minimum objective value of the feature split algorithm PMD (see [149]).

A binary classifier is trained for all the methods, and linear kernels are used. The experiments on the BBC and BBCSport data sets are repeated on 10 training/testing data splits. The experiments on the WebKB data set are run with 100 training/testing data splits as in [1].

4.6.2.3 Performance

The results on these three data sets are reported in Table 4.3. Mean average precisions (MAPs) of 5 classes over 10 rounds are reported for the BBC and BBCSport data sets. Following [1], the average of precision-recall break-even point (PRBEP) over 100 splits are reported for the WebKB data set. The results of SVM, TSVM and Co-LapSVM on the WebKB data set are copied from Table 1 in [1].

From the results, we can see that our method achieves the best results for the joint-view, which demonstrates the advantage of our co-labeling for semi-supervised learning. The improvements over the second best in the three data sets are 4.40%, 4.74% and 0.9%, respectively. It is worth noting that the co-labeling is significantly better than other methods on the BBC and BBCSport data sets for each single view as well as the joint-view. Although the co-labeling does not achieve best results for the page view and link view on the WebKB data set, the late fusion result is the best which again demonstrates that our method can effectively use multi-view information. The relative improvement is not as significant as on the BBC and BBCSport data sets because it already has a very high performance, which can be confirmed by the fact our method achieves 99.11% in the measurement of MAP over 100 splits.

4.6.2.4 Convergence Analysis

We have analyzed that the objective value of MKL on each view decreases with respect to the number of iterations (see Section 4.3.3). Taking the first topic (*i.e.* “athletics”)

Figure 4.2: The objective values with respect to the number of iterations on two views.

Table 4.4: Performance of different versions of co-labeling on BBC-Sports.

	View1	View2	View1+View2
Co-Labeling(iter)	58.19	61.56	69.77
Co-Labeling(nobias)	74.21	72.00	77.43
Co-Labeling	82.10	79.60	84.22

of the BBC-Sports data set as an example, we show the objective values of two views in Figure 4.2. It can be observed the objective values of the two views converge very fast and it takes only six iterations to reach the stop criterion. The average number of iterations for all 5 classes over 10 splits is 6.36.

4.6.2.5 Strategy Analysis

Three strategies for updating the label candidate set are discussed in Section 4.2. Here we evaluate those three strategies by introducing two simplified versions of our co-labeling algorithms: the first version is *Co-Labeling(iter)*, in which we only use the labeling from the latest iteration; and the other one, *Co-Labeling(nobias)*, uses labelings from all previous iterations but does not vary biases at each iteration. We also take the BBCSport data set as an example, and the results of those two simplified versions of our co-labeling approach are shown in Table 4.4. We can observe that the performance of *Co-Labeling(iter)* is degraded without using MKL and the bias. It is even worse than the results of SVM reported in Table 4.3 which only uses the labeled data. This is not surprising since the labeling from the trained classifiers may be very noisy. Directly feeding the labeling as

Table 4.5: Average training times of different multi-view SSL methods per class per split on three data sets (in seconds).

	BBC	BBCSport	WebKB
Co-LapSVM	52.45	2.136	16.69
2V-TSVM	1108	497.3	446.4
PMC	30.64	7.215	55.41
Co-Labeling	36.50	5.111	21.27

the training labels to the other view cannot guarantee a robust classifier. After using MKL, the performances of Co-Labeling(nobias) are improved a lot, which are comparable to other multi-view methods as reported in Table 4.3. However, Co-Labeling(nobias) still suffers from the bias problem since we use the prediction as labeling while other methods do not have this issue. After varying the biases, the final results are significantly better than other methods.

4.6.2.6 Time Comparison

The training time of different multi-view methods are reported in Table 4.5. All methods are performed on a workstation with a 3.3GHz CPU. We implement the co-labeling with unoptimized MATLAB code using LIBSVM⁶. We observe that the co-labeling is comparable to the Co-LapSVM and PMC in training time on the three data sets. It demonstrates that the co-labeling is practical and scalable for real applications considering it can be further sped up after a better implementation and using a faster solver (for example LIBLINEAR for the linear case).

4.7 Summary

To effectively utilize different types of multi-view weakly labeled data, in this chapter we have studied a new problem called multi-view weakly labeled learning, which covers various weakly labeled learning problems including SSL and MIL under the multi-view setting. We firstly propose a co-labeling framework to solve the multi-view weakly labeled learning problem using pseudo-label vectors. For each view, we formulate it as a weakly labeled learning problem to train a more robust classifier based on a set of pseudo-label

⁶It can be further accelerated by using LIBLINEAR which solves the SVM in the primal form.

vectors generated from different iterations, biases and views. We have also presented a theoretical analysis for our co-labeling approach. Extensive experimental results for MIL and SSL on real-world multi-view datasets demonstrate that our proposed approach achieves state-of-the-art results.

Chapter 5

Semi-supervised Heterogeneous Feature Augmentation

In this chapter, we study the domain adaptation issue when learning from web data. Specifically, we consider the heterogeneous domain adaptation problem, in which the feature representations of the source and target training samples are different. To solve this problem, we build upon the recent proposed heterogeneous feature augmentation (HFA) method, and reformulate it as an MKL problem, which is convex and thus the global optimum can be guaranteed. We further extend HFA to Semi-supervised HFA (SHFA), which can effectively utilize the additional unlabeled samples in the target domain. We also formulate SHFA as an MKL problem. We conduct experiments on various applications including image recognition, document classification, and sentiment classification. Extensive experimental results demonstrate the effectiveness of our methods.

5.1 Introduction

In real-world applications, it is often expensive and time-consuming to collect the labeled data. Transfer learning (a.k.a., domain adaptation), as a new machine learning strategy, has attracted growing attention because it can learn robust classifiers with very few or even no labeled data from the target domain by leveraging a large amount of labeled data from other existing domains (a.k.a., source/auxiliary domains).

Domain adaptation methods have been successfully used for different research fields such as natural language processing and computer vision [90, 91, 108, 150–153]. According

to the supervision information in the target domain, the domain adaptation methods can generally be divided into three categories: supervised domain adaptation by only using the labeled data in the target domain, semi-supervised domain adaptation by using both the labeled and unlabeled data in the target domain, and unsupervised domain adaptation by only using unlabeled data in the target domain. However, most existing methods assume that the data from different domains are represented by the same type of features with the same dimension. Thus, they cannot deal with the problem where the dimensions of data from the source and target domains are different, which is known as heterogeneous domain adaptation (HDA) [120, 123].

In the literature, a few approaches have been proposed for the HDA problem. To discover the connection between different features, some work exploited an auxiliary dataset which encodes the correspondence between different types of features. Dai *et al.* [120] proposed to learn a feature translator between two features from two domains, which is modeled by the conditional probability of one feature given the other one. Such feature translator is learnt from an auxiliary dataset which contains the co-occurrence of these two types of features. A similar assumption was also used in [123, 124] for text-aid image clustering and classification. Others proposed to use an explicit feature correspondence, for example, the bilingual dictionary in cross-language text classification task. Based on the structural correspondence learning (SCL) [150], two methods [121, 122] were recently proposed to extract the so-called *pivot* features from the source and target domains, which are specifically designed for the cross-language text classification task. These pivot features are constructed by text words which have explicit semantic meanings. They either directly translated the pivot features from one language to the other or modified the original SCL to select pairs of pivot words from different languages. However, it is unclear how to build such correspondence for more general HDA tasks such as the object recognition task where only the low-level visual features are provided.

For more general HDA tasks, Shi *et al.* [125] proposed a method called Heterogeneous Spectral Mapping (HeMap) to discover a common feature subspace by learning two feature mapping matrices as well as the optimal projection of the data from both domains. Harel and Mannor [154] learnt rotation matrices to match source data distributions to that of the target domain. Wang *et al.* [126] used the class labels of training

Figure 5.1: The samples from different domains are represented by different features, where red crosses, blue strips, orange triangles and green circles denote source positive samples, source negative samples, target positive samples and target negative samples, respectively. By using two projection matrices $\mathbf{P}$ and $\mathbf{Q}$, the heterogeneous samples from two domains are transformed into an augmented feature space.

data to learn the manifold alignment by simultaneously maximizing the intra-domain similarity and the inter-domain dissimilarity. By kernelizing the method in [155], Kulis *et al.* [127] proposed to learn an asymmetric kernel transformation to transfer feature knowledge between the data from the source and target domains. However, these existing HDA methods were designed for the supervised learning scenario. For these methods, it is unclear how to learn the projection matrices or transformation metric by utilizing the abundant unlabeled data in the target domain which is usually available in many applications.

Our work builds up the recent proposed Heterogeneous Feature Augmentation (HFA)[24] for supervised heterogeneous domain adaptation. The HFA method transforms the data from the source and target domains into a common subspace by using feature mapping functions associated with two projection matrices $\mathbf{P}$ and $\mathbf{Q}$ (as shown in Figure 5.1). By combining with SVM, a joint objective function is proposed to learn the project matrices and the classifier. In the previous work on HFA [24], an alternating optimization algorithm was used to iteratively learn an individual transformation metric $\mathbf{H}$ (which is a combination of two projection matrices $\mathbf{P}$ and $\mathbf{Q}$) and a classifier for each class. However, there may be pre-mature convergence, in other words, the algorithm may not converge to a stationary point, so the global convergence cannot be guaranteed. In this work, we equivalently reformulate it into a convex optimization problem by decomposing $\mathbf{H}$ into a linear combination of a set of rank-one positive semi-definite (PSD) matrices, which shares a similar formulation with the well-known Multiple Kernel

Learning (MKL) problem [72]. Therefore, the global solution can be obtained by using the existing MKL solvers.

Moreover, we further extend our reformulated HFA to semi-supervised HFA (or SHFA in short) by additionally utilizing the unlabeled data in the target domain. While learning the transformation metric $\mathbf{H}$, we also infer the labels for the unlabeled target samples. Considering we need to solve a non-trivial mixed integer programming problem when inferring the labels of unlabeled target training data, we first relax the objective of SHFA into a problem of finding the optimal linear combination of all possible label candidates. Then we also use the linear combination of these rank-one PSD matrices to replace $\mathbf{H}$ as in HFA. Finally, we further rewrite the problem as a convex MKL problem which can be readily solved by existing MKL solvers.

5.2 Heterogeneous Feature Augmentation

In this work, we assume there are only one source domain and one target domain. We are provided with a set of labeled training samples $\{(\mathbf{x}_i^s, y_i^s)|_{i=1}^{n_s}\}$ from the source domain as well as a limited number of labeled samples $\{(\mathbf{x}_i^t, y_i^t)|_{i=1}^{n_t}\}$ from the target domain, where y_i^s and y_i^t are the labels of the samples $\mathbf{x}_i^s$ and $\mathbf{x}_i^t$, respectively, and $y_i^s, y_i^t \in \{1, -1\}$. The dimensions of $\mathbf{x}_i^s$ and $\mathbf{x}_i^t$ are d_s and d_t , respectively. Note that in the HDA problem, $d_s \neq d_t$. We also define $\mathbf{X}_s = [\mathbf{x}_1^s, \dots, \mathbf{x}_{n_s}^s] \in \mathbb{R}^{d_s \times n_s}$ and $\mathbf{X}_t = [\mathbf{x}_1^t, \dots, \mathbf{x}_{n_t}^t] \in \mathbb{R}^{d_t \times n_t}$ as the data matrices for the source and target domains, respectively.

5.2.1 A Review of Heterogeneous Feature Augmentation

Daume III [91] proposed Feature Replication (FR) to augment the original feature space $\mathbb{R}^d$ into a larger space $\mathbb{R}^{3d}$ by replicating the source and target data for homogeneous domain adaptation. Specifically, for any data point $\mathbf{x} \in \mathbb{R}^d$, the feature mapping functions φ_s and φ_t for the source and target domains are defined as $\varphi_s(\mathbf{x}) = [\mathbf{x}', \mathbf{x}', \mathbf{0}_d]'$ and $\varphi_t(\mathbf{x}) = [\mathbf{x}', \mathbf{0}_d, \mathbf{x}']'$. By incorporating the original features into the augmented features φ_s and φ_t , the similarities between data from the same domain are enhanced, which is beneficial for the domain adaptation task.

When the source and target domains are represented with different types of features, it is not meaningful to directly use the method in [91] for the HDA task by simply

padding zeros to make the dimensions of the data from two domains become the same, because there would be no correspondences between the heterogeneous features in this case. Duan *et al.* [24] extended the FR method for heterogeneous domain adaptation, which is referred to as heterogeneous feature augmentation (HFA). Specifically, they respectively introduce two projection matrices $\mathbf{P} \in \mathbb{R}^{d_c \times d_s}$ and $\mathbf{Q} \in \mathbb{R}^{d_c \times d_t}$ for the source and target domains, such that any source sample $\mathbf{x}^s$ and target sample $\mathbf{x}^t$ can be projected onto a common subspace $\mathbb{R}^{d_c}$. Then they define the feature mapping functions φ_s and φ_t as follows:

$$\varphi_s(\mathbf{x}^s) = \begin{bmatrix} \mathbf{P}\mathbf{x}^s \\ \mathbf{x}^s \\ \mathbf{0}_{d_t} \end{bmatrix} \text{ and } \varphi_t(\mathbf{x}^t) = \begin{bmatrix} \mathbf{Q}\mathbf{x}^t \\ \mathbf{0}_{d_s} \\ \mathbf{x}^t \end{bmatrix}. \quad (5.1)$$

With the above feature mapping functions, one can learn an SVM classifier $f(\mathbf{x}^t) = \mathbf{w}\varphi_t(\mathbf{x}_i^t) + b$ by using the labeled training samples from both domains as follows,

$$\min_{\mathbf{P}, \mathbf{Q}} \min_{\mathbf{w}, b, \xi_i^s, \xi_i^t} \frac{1}{2} \|\mathbf{w}\|^2 + C \left(\sum_{i=1}^{n_s} \xi_i^s + \sum_{i=1}^{n_t} \xi_i^t \right), \quad (5.2)$$

$$\text{s.t.} \quad y_i^s(\mathbf{w}'\varphi_s(\mathbf{x}_i^s) + b) \geq 1 - \xi_i^s, \quad \xi_i^s \geq 0; \quad (5.3)$$

$$y_i^t(\mathbf{w}'\varphi_t(\mathbf{x}_i^t) + b) \geq 1 - \xi_i^t, \quad \xi_i^t \geq 0; \quad (5.4)$$

$$\|\mathbf{P}\|_F^2 \leq \lambda_p, \quad \|\mathbf{Q}\|_F^2 \leq \lambda_q,$$

where $\mathbf{w} \in \mathbb{R}^{d_c + d_s + d_t}$ is the weight vector for the SVM classifier, $C > 0$ is a tradeoff parameter which balances the model complexity and the empirical losses on the training samples from two domains, and $\lambda_p, \lambda_q > 0$ are predefined parameters to control the complexities of $\mathbf{P}$ and $\mathbf{Q}$, respectively.

Let us define an intermediate variable $\mathbf{H} = [\mathbf{P}, \mathbf{Q}]'[\mathbf{P}, \mathbf{Q}] \in \mathbb{R}^{(d_s + d_t) \times (d_s + d_t)}$. By introducing dual variables $\{\alpha_i^s\}_{i=1}^{n_s}$ and $\{\alpha_i^t\}_{i=1}^{n_t}$ for the constraints in (5.3) and (5.4), respectively, one can derive the dual form of (5.2) as follows,

$$\begin{aligned} \min_{\mathbf{H} \succeq 0} \max_{\boldsymbol{\alpha}} \quad & \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})'\mathbf{K}_{\mathbf{H}}(\boldsymbol{\alpha} \circ \mathbf{y}), \\ \text{s.t.} \quad & \mathbf{y}'\boldsymbol{\alpha} = 0, \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \\ & \text{tr}(\mathbf{H}) \leq \lambda, \end{aligned} \quad (5.5)$$

where $\boldsymbol{\alpha} = [\alpha_1^s, \dots, \alpha_{n_s}^s, \alpha_1^t, \dots, \alpha_{n_t}^t]'$ is a vector of the dual variables, $\mathbf{y} = [\mathbf{y}'_s, \mathbf{y}'_t]'$ is the label vector of all training samples, $\mathbf{y}_s = [y_1^s, \dots, y_{n_s}^s]'$ is the label vector of source samples, and $\mathbf{y}_t = [y_1^t, \dots, y_{n_t}^t]'$ is the label vector of target samples.

$\{+1, -1\}^{n_s}$ is the label vector of the samples from source domain, $\mathbf{y}_t = [y_1^t, \dots, y_{n_t}^t]'$ $\in \{+1, -1\}^{n_t}$ is the label vector of the samples from target domain, $\mathbf{K}_H = \mathbf{X}'(\mathbf{H} + \mathbf{I})\mathbf{X}$, $\mathbf{X} = \begin{bmatrix} \mathbf{X}_s & \mathbf{O}_{d_s \times n_t} \\ \mathbf{O}_{d_t \times n_s} & \mathbf{X}_t \end{bmatrix} \in \mathbb{R}^{(d_s+d_t) \times (n_s+n_t)}$ and $\lambda = \lambda_p + \lambda_q$.

Nonlinear HFA: The nonlinear form of HFA shares a similar form with the linear HFA problem in (5.5). Let us denote the kernel on the source domain samples as $\mathbf{K}_s = \Phi_s' \Phi_s \in \mathbb{R}^{n_s \times n_s}$, where $\Phi_s = [\phi_s(\mathbf{x}_1^s), \dots, \phi_s(\mathbf{x}_{n_s}^s)]$ and $\phi_s(\cdot)$ is the nonlinear feature mapping function induced by $\mathbf{K}_s$. Similarly, we denote the kernel on the target domain samples as $\mathbf{K}_t = \Phi_t' \Phi_t \in \mathbb{R}^{n_t \times n_t}$, where $\Phi_t = [\phi_t(\mathbf{x}_1^t), \dots, \phi_t(\mathbf{x}_{n_t}^t)]$ and $\phi_t(\cdot)$ is the nonlinear feature mapping function induced by $\mathbf{K}_t$. As in the linear case, we can correspondingly define the augmented features $\varphi_s(\mathbf{x}^s)$ and $\varphi_t(\mathbf{x}^t)$ in (5.1) for the nonlinear features of two domains by replacing $\mathbf{x}^s$ and $\mathbf{x}^t$ with $\phi_s(\mathbf{x}^s)$ and $\phi_t(\mathbf{x}^t)$, respectively. Denoting the dimensions of the nonlinear features $\phi_s(\mathbf{x}^s)$ and $\phi_t(\mathbf{x}^t)$ as $\tilde{d}_s$ and $\tilde{d}_t$, we can also derive an optimization problem as in (5.5) to solve a transformation metric $\mathbf{H} \in \mathbb{R}^{(\tilde{d}_s+\tilde{d}_t) \times (\tilde{d}_s+\tilde{d}_t)}$ which maps the different nonlinear features from two domains into a common feature space. Correspondingly, the kernel can be written as $\mathbf{K}_H = \Phi'(\mathbf{H} + \mathbf{I})\Phi$ where $\Phi = \begin{bmatrix} \Phi_s & \mathbf{O}_{\tilde{d}_s \times n_t} \\ \mathbf{O}_{\tilde{d}_t \times n_s} & \Phi_t \end{bmatrix} \in \mathbb{R}^{(\tilde{d}_s+\tilde{d}_t) \times (n_s+n_t)}$. Then the nonlinear HFA problem can be formulated as:

$$\begin{aligned} \min_{\tilde{\mathbf{H}} \succeq 0} \max_{\boldsymbol{\alpha}} \quad & \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})'\mathbf{K}_{\tilde{\mathbf{H}}}(\boldsymbol{\alpha} \circ \mathbf{y}), \\ \text{s.t.} \quad & \mathbf{y}'\boldsymbol{\alpha} = 0, \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \\ & \text{tr}(\tilde{\mathbf{H}}) \leq \lambda. \end{aligned} \tag{5.6}$$

where $\tilde{\mathbf{H}} \in \mathbb{R}^{(n_s+n_t) \times (n_s+n_t)}$ is a nonlinear transformation matrix satisfies that $\mathbf{H} = \Phi\mathbf{K}^{-\frac{1}{2}}\tilde{\mathbf{H}}\mathbf{K}^{-\frac{1}{2}}\Phi'$ with $\mathbf{K} = \begin{bmatrix} \mathbf{K}_s & \mathbf{O}_{n_s \times n_t} \\ \mathbf{O}_{n_t \times n_s} & \mathbf{K}_t \end{bmatrix} \in \mathbb{R}^{(n_s+n_t) \times (n_s+n_t)}$ and $\mathbf{K}^{\frac{1}{2}}$ being the symmetric square root of $\mathbf{K}$. More details on the derivations of (5.6) can be found in [24, 128].

As the two formulations share the same form from the perspective of optimization, we will only discuss the nonlinear case in the remainder of this chapter while the linear case can be similarly derived by replacing $\mathbf{K}^{\frac{1}{2}} \in \mathbb{R}^{(n_s+n_t) \times (n_s+n_t)}$ and $\tilde{\mathbf{H}} \in \mathbb{R}^{(n_s+n_t) \times (n_s+n_t)}$ with $\mathbf{X} \in \mathbb{R}^{(d_s+d_t) \times (n_s+n_t)}$ and $\mathbf{H} \in \mathbb{R}^{(d_s+d_t) \times (d_s+d_t)}$, respectively. We also use $\mathbf{H}$ instead of $\tilde{\mathbf{H}}$ below for better presentation.

5.2.2 A Convex Formulation

To solve the optimization problem in (5.6), in the pervious work on HFA [24], an alternating optimization approach was adopted to iteratively solve an SVM problem with respect to α and a semi-definite programming (SDP) problem with respect to $\mathbf{H}$. However, the global convergence remains unclear and there may be pre-mature convergence. In this subsection, we show that (5.6) can be equivalently reformulated as a convex MKL problem so that the global solution can be guaranteed by using the existing MKL solvers [72].

As pointed in [72], the Ivanov regularization can be replaced with some Tikhonov regularization and vice versa with the appropriate choice of regularization parameter, which means we can write the trace norm regularization in (5.6) either as a constraint or as a regularizer term in the objective function. Formally, let us denote $\mu(\mathbf{H}) = \max_{\alpha \in \mathcal{A}} \mathbf{1}'\alpha - \frac{1}{2}(\alpha \circ \mathbf{y})'\mathbf{K}_\mathbf{H}(\alpha \circ \mathbf{y})$ where $\mathcal{A} = \{\alpha | \mathbf{y}'\alpha = 0, \mathbf{0} \leq \alpha \leq C\mathbf{1}\}$, then the problem in (5.6) can also be reformulated as:

$$\min_{\mathbf{H} \succeq \mathbf{0}} \quad \mu(\mathbf{H}) + \eta \text{tr}(\mathbf{H}), \quad (5.7)$$

where η is a tradeoff parameter. By properly setting η , the above optimization problem yields the same solution as the original problem in (5.6) [72].

To avoid solving the non-trivial SDP problem as in [24], we propose to decompose $\mathbf{H}$ as a linear combination of a set of positive semi-definite (PSD) matrices. Inspired by [156], in this work, we use the set of rank-one normalized PSD matrices which is defined as $\mathcal{M} = \{\mathbf{M}_r |_{r=1}^\infty\}$ where $\mathbf{M}_r = \mathbf{h}_r \mathbf{h}_r'$, $\mathbf{h}_r \in \mathbb{R}^{(n_s+n_t)}$ and $\mathbf{h}_r' \mathbf{h}_r = 1$. Then, any PSD matrix $\mathbf{H}$ in (5.7) can be represented as a linear combination of the rank-one PSD matrices in $\mathcal{M}$, *i.e.*, $\mathbf{H} = \mathbf{H}_\theta = \sum_{r=1}^\infty \theta_r \mathbf{M}_r$ where the linear combination coefficient vector $\theta = [\theta_1, \dots, \theta_\infty]'$, $\theta \geq \mathbf{0}$, $\mathbf{1}'\theta \leq \lambda$. Although there are an infinite number of matrices in $\mathcal{M}$ (*i.e.*, the index r goes from 1 to ∞), only considering the linear combination vector θ with a finite number of non-zero entries is sufficient to represent $\mathbf{H}$, since there exists a continuous linear map from these θ 's to the PSD matrices [156].

Note that we have $\text{tr}(\mathbf{H}) = \text{tr}(\sum_{r=1}^\infty \theta_r \mathbf{M}_r) = \sum_{r=1}^\infty \theta_r \text{tr}(\mathbf{M}_r) = \mathbf{1}'\theta$. Instead of directly solving for the optimal $\mathbf{H}$ in (5.7), we show in the following theorem that it is equivalent to solving for the optimal linear combination coefficient vector θ :

Theorem 5.2 *Given that $\boldsymbol{\theta}^*$ is the optimal solution to the following optimization problem,*

$$\min_{\boldsymbol{\theta} \geq \mathbf{0}} \quad \mu(\mathbf{H}_{\boldsymbol{\theta}}) + \eta \mathbf{1}'\boldsymbol{\theta}, \quad (5.8)$$

$\mathbf{H}_{\boldsymbol{\theta}^}$ is also the optimum to the optimization problem in (5.7).*

Proof: Let us denote the objective function in (5.7) as $F(\mathbf{H}) = \mu(\mathbf{H}) + \eta \text{tr}(\mathbf{H})$ and the objective function in (5.8) as $G(\boldsymbol{\theta}) = \mu(\mathbf{H}_{\boldsymbol{\theta}}) + \eta \mathbf{1}'\boldsymbol{\theta}$, and denote the optimum to (5.7) and (5.8) as $\mathbf{H}^* = \arg \min_{\mathbf{H} \succeq \mathbf{0}} F(\mathbf{H})$ and $\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta} \geq \mathbf{0}} G(\boldsymbol{\theta})$, respectively. To show $\mathbf{H}_{\boldsymbol{\theta}^*}$ is also the optimum of (5.7), we need to prove $F(\mathbf{H}_{\boldsymbol{\theta}^*}) = F(\mathbf{H}^*)$.

On one hand, we have $F(\mathbf{H}_{\boldsymbol{\theta}^*}) \geq F(\mathbf{H}^*)$, because $\mathbf{H}^*$ is the optimal solution to (5.7). On the other hand, we will prove it as $F(\mathbf{H}^*) \geq G(\boldsymbol{\theta}^*) = F(\mathbf{H}_{\boldsymbol{\theta}^*})$. Specifically, for any PSD matrix $\mathbf{H}$ and a vector $\boldsymbol{\theta}$ which satisfies $\mathbf{H} = \mathbf{H}_{\boldsymbol{\theta}} = \sum_{r=1}^{\infty} \theta_r \mathbf{M}_r$, we have $F(\mathbf{H}) = \mu(\mathbf{H}) + \eta \text{tr}(\mathbf{H}) = \mu(\mathbf{H}_{\boldsymbol{\theta}}) + \eta \mathbf{1}'\boldsymbol{\theta} = G(\boldsymbol{\theta}) \geq G(\boldsymbol{\theta}^*)$ in which $G(\boldsymbol{\theta}) \geq G(\boldsymbol{\theta}^*)$ is due to the fact that $\boldsymbol{\theta}^*$ is the optimal solution to (5.8). Thus we have $F(\mathbf{H}^*) \geq G(\boldsymbol{\theta}^*)$. Moreover, since $G(\boldsymbol{\theta}^*) = \mu(\mathbf{H}_{\boldsymbol{\theta}^*}) + \eta \mathbf{1}'\boldsymbol{\theta}^* = \mu(\mathbf{H}_{\boldsymbol{\theta}^*}) + \eta \text{tr}(\mathbf{H}_{\boldsymbol{\theta}^*}) = F(\mathbf{H}_{\boldsymbol{\theta}^*})$, we have $F(\mathbf{H}^*) \geq G(\boldsymbol{\theta}^*) = F(\mathbf{H}_{\boldsymbol{\theta}^*})$.

Finally, we conclude that $F(\mathbf{H}_{\boldsymbol{\theta}^*}) = F(\mathbf{H}^*)$, because we have proved $F(\mathbf{H}_{\boldsymbol{\theta}^*}) \geq F(\mathbf{H}^*)$ and $F(\mathbf{H}^*) \geq G(\boldsymbol{\theta}^*) = F(\mathbf{H}_{\boldsymbol{\theta}^*})$. This completes the proof.

By replacing the Tikhonov regularization (5.8) with the corresponding Ivanov regularization (*i.e.* the regularizer term $\mathbf{1}'\boldsymbol{\theta}$ is rewritten as the constraint), we reformulate the optimization problem of HFA as:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad & \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})' \mathbf{S}(\mathbf{H}_{\boldsymbol{\theta}} + \mathbf{I}) \mathbf{S}(\boldsymbol{\alpha} \circ \mathbf{y}), \\ \text{s.t.} \quad & \mathbf{H}_{\boldsymbol{\theta}} = \sum_{r=1}^{\infty} \theta_r \mathbf{M}_r, \quad \mathbf{M}_r \in \mathcal{M}, \\ & \mathbf{1}'\boldsymbol{\theta} \leq \lambda, \quad \boldsymbol{\theta} \geq \mathbf{0}. \end{aligned} \quad (5.9)$$

By setting $\boldsymbol{\theta} \leftarrow \frac{1}{\lambda} \boldsymbol{\theta}$, it can be further rewritten as:

$$\min_{\boldsymbol{\theta} \in \mathcal{D}_{\boldsymbol{\theta}}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})' \sum_{r=1}^{\infty} \theta_r \mathbf{K}_r (\boldsymbol{\alpha} \circ \mathbf{y}), \quad (5.10)$$

where $\mathbf{K}_r = \mathbf{S}(\lambda \mathbf{M}_r + \mathbf{I}) \mathbf{S}$ and $\mathcal{D}_{\boldsymbol{\theta}} = \{\boldsymbol{\theta} | \mathbf{1}'\boldsymbol{\theta} \leq 1, \boldsymbol{\theta} \geq \mathbf{0}\}$. It is an Infinite Kernel Learning (IKL) problem with each base kernel as $\mathbf{K}_r$, which can be readily solved with the existing MKL solver [72, 157].

5.2.3 Solution

One problem in (5.10) is that there are an infinite number of base kernels because the set $\mathcal{M}$ contains infinite rank-one matrices. However, a finite number of rank-one matrices are sufficient to represent the matrix $\mathbf{H}$ [156]. Inspired by [157], we solve (5.10) based on a small number of base kernels which are constructed by using the cutting-plane algorithm. Let us introduce a dual variable τ for $\boldsymbol{\theta}$ in (5.10) and write the dual form as:

$$\begin{aligned} \max_{\tau, \boldsymbol{\alpha} \in \mathcal{A}} \quad & \mathbf{1}'\boldsymbol{\alpha} - \tau, \\ \text{s.t.} \quad & \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})'\mathbf{K}_r(\boldsymbol{\alpha} \circ \mathbf{y}) \leq \tau, \quad \forall r, \end{aligned} \quad (5.11)$$

which has an infinite number of constraints. With the cutting-plane algorithm, we can approximate (5.11) by iteratively adding a kernel for which the corresponding constraint is violated according to the current solution. The kernel associated with this constraint is called an *active kernel*. To find the most active kernel, we need to maximize the left-hand side of the constraint in (5.11), which is given as:

$$\max_{\mathbf{M} \in \mathcal{M}} \quad \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})'\mathbf{K}_{\mathbf{M}}(\boldsymbol{\alpha} \circ \mathbf{y}), \quad (5.12)$$

where $\mathbf{K}_{\mathbf{M}} = \mathbf{S}(\lambda\mathbf{M} + \mathbf{I})\mathbf{S}$. It has a closed form solution as $\mathbf{M} = \mathbf{h}\mathbf{h}' \in \mathbb{R}^{(n_s+n_t) \times (n_s+n_t)}$ with $\mathbf{h} = \frac{\mathbf{S}(\boldsymbol{\alpha} \circ \mathbf{y})}{\|\mathbf{S}(\boldsymbol{\alpha} \circ \mathbf{y})\|}$.

We summarize the proposed algorithm in Algorithm 6. First, we initialize the set of rank-one PSD matrices $\mathcal{M}$ with $\mathbf{M}_1 = \mathbf{h}_1\mathbf{h}_1'$ where $\mathbf{h}_1$ is a unit vector. Based on the current $\mathcal{M}$, we solve the MKL problem in (5.10) to obtain the optimal $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$. After that, we find the most active kernel which is decided by a rank-one PSD matrix $\mathbf{M}$ as in (5.12). By using the closed form solution of (5.12), we obtain a new rank-one PSD matrix and add it into the current set $\mathcal{M}$. Then we solve the MKL problem again. The above steps are repeated until convergence. After obtaining the optimal solution $\boldsymbol{\alpha}$ and $\mathbf{H}$ to (5.10), we can predict any test sample $\mathbf{x}$ from the target domain by using the following target decision function:

$$f(\mathbf{x}) = (\boldsymbol{\alpha} \circ \mathbf{y})'\mathbf{K}^{\frac{1}{2}}(\mathbf{H} + \mathbf{I}) \begin{bmatrix} \mathbf{O}_{n_s \times n_t} \\ \mathbf{K}_t^{-\frac{1}{2}} \end{bmatrix} \mathbf{k}_t + b, \quad (5.13)$$

Algorithm 6 Heterogeneous Feature Augmentation

Require: Labeled source samples $\{(\mathbf{x}_i^s, y_i^s)|_{i=1}^{n_s}\}$ and labeled target samples $\{(\mathbf{x}_i^t, y_i^t)|_{i=1}^{n_t}\}$.

- 1: Set $r = 1$ and initialize $\mathcal{M}_1 = \{\mathbf{M}_1\}$ with $\mathbf{M}_1 = \mathbf{h}_1 \mathbf{h}_1'$ and $\mathbf{h}_1 = \frac{1}{\sqrt{n_s+n_t}} \mathbf{1}_{n_s+n_t}$.
- 2: **repeat**
- 3: Solve $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ in (5.10) based on $\mathcal{M}_r$ by using the existing MKL solver [72].
- 4: Obtain a rank-one PSD matrix $\mathbf{M}_{r+1}$ by solving (5.12).
- 5: Set $\mathcal{M}_{r+1} = \mathcal{M}_r \cup \{\mathbf{M}_{r+1}\}$, and $r = r + 1$.
- 6: **until** The objective converges.

Ensure: $\boldsymbol{\alpha}$ and $\mathbf{H} = \lambda \sum_r \theta_r \mathbf{M}_r$.

where $\mathbf{k}_t = [k(\mathbf{x}_1^t, \mathbf{x}), \dots, k(\mathbf{x}_{n_t}^t, \mathbf{x})]'$ and $k(\mathbf{x}_i, \mathbf{x}_j) = \phi_t(\mathbf{x}_i)' \phi_t(\mathbf{x}_j)$ is a predefined kernel function for two data samples $\mathbf{x}_i$ and $\mathbf{x}_j$ in the target domain.

Complexity Analysis: In our HFA, we first calculate $\mathbf{S}$ once at the beginning, which costs $O(n^3)$ time with $n = n_s + n_t$ being the total number of training samples¹. After that, we perform the cutting-plane algorithm (*i.e.*, Algorithm 6), in which we iteratively train an MKL classifier and find the most violated rank-one matrix as in (5.12). As we have an efficient closed form solution for solving (5.12), the major time cost of Algorithm 6 is from the training of MKL at each iteration. However, the time complexity of MKL has not been theoretically analyzed. Usually, the MKL solver needs to train an SVM classifier for a few iterations. The empirical analysis shows that optimizing the QP problem in SVM is about $O(n^{2.3})$ [25]. Therefore, the complexity of MKL is $O(Ln^{2.3})$ with L being the number of iterations in MKL. Thus, the total time complexity of our HFA is $O(n^3 + TLn^{2.3})$, where T is the number of iterations in Algorithm 6. In practice, both L and T are not very large.

5.2.4 Convergence Analysis

Let us represent the objective function in (5.10) as $F(\boldsymbol{\alpha}, \boldsymbol{\theta}) = \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})' \sum_{r=1}^{\infty} \theta_r \mathbf{K}_r(\boldsymbol{\alpha} \circ \mathbf{y})$, and also denote the optimal solution to (5.10) as $(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) = \arg \min_{\boldsymbol{\theta} \in \mathcal{D}_{\boldsymbol{\theta}}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} F(\boldsymbol{\alpha}, \boldsymbol{\theta})$.

We denote the optimal solution of the MKL problem at the r -th iteration as $(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r)$. Because there are at most r non-zero elements in $\boldsymbol{\theta}^r$, we assume these non-zero elements are the first r entries in $\boldsymbol{\theta}^r$ for ease of presentation. Then, we show in the following theorem that Algorithm 6 converges to the global optimal solution:

¹More accurately, the time complexity for solving $\mathbf{K}$ is $O(n_s^3 + n_t^3)$, because the kernel matrix $\mathbf{K}$ is a block-diagonal matrix.

Theorem 5.3 *With Algorithm 6, $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r)$ monotonically decreases as r increases, and the following inequality holds*

$$F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) \geq F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) \geq F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1}),$$

where $\mathbf{e}_{r+1} \in \mathcal{D}_{\boldsymbol{\theta}}$ is the vector with all zeros except the $(r+1)$ -th entry being 1. We also have $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) = F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) = F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1})$ when Algorithm 6 converges at the r -th iteration.

The theorem can be proved similarly as in [158]. We also give the proof in the Appendix. Moreover, as indicated in [159], the cutting-plane algorithm stops in a finite number of steps under some conditions. In our experiments, the algorithm usually takes less than 50 iterations to obtain a sufficient accurate solution.

5.3 Semi-supervised Heterogeneous Feature Augmentation

The unlabeled data has been demonstrated to be helpful for training a robust classifier in many applications [54]. For the traditional semi-supervised learning, readers can refer to [51] for a comprehensive survey. There are also many works on semi-supervised homogeneous domain adaptation, such as [92, 98, 107]. However, most existing heterogeneous domain adaptation works [125–127] were designed for the supervised setting, and cannot utilize the abundant unlabeled data in the target domain. Thus, we further propose semi-supervised HFA to utilize the unlabeled data in the target domain.

We still use $\{(\mathbf{x}_i^s, y_i^s)|_{i=1}^{n_s}\}$ and $\{(\mathbf{x}_i^t, y_i^t)|_{i=1}^{n_t}\}$ to represent the labeled data from the source domain and the target domain, respectively. Let us denote the unlabeled data in the target domain as $\{(\mathbf{x}_i^u, y_i^u)|_{i=1}^{n_u}\}$ where $\mathbf{x}_i^u \in \mathbb{R}^{d_t}$ is an unlabeled sample in the target domain, n_u is the number of unlabeled samples, and the label $y_i^u \in \{-1, +1\}$ is unknown. We also denote $\mathbf{y}_u = [y_1^u, \dots, y_{n_u}^u]'$ as the label vector of all the unlabeled data. Moreover, the total number of training samples is denoted as $n = n_s + n_t + n_u$.

5.3.1 Formulation

Since the labels of unlabeled data are unknown, we propose to infer the optimal labeling $\mathbf{y}_u$ for the unlabeled data in the target domain when learning the classifier. Based on the ρ -SVM with the squared hinge loss, we propose the objective for semi-supervised heterogeneous domain adaptation as follows:

$$\begin{aligned}
 \min_{\substack{\mathbf{y}_u, \mathbf{w}, b, \rho, \\ \mathbf{P}, \mathbf{Q}, \xi_i^s, \xi_i^t, \xi_i^u}} \quad & \frac{1}{2} (\|\mathbf{w}\|^2 + b^2) - \rho + \frac{C}{2} \left(\sum_{i=1}^{n_s} (\xi_i^s)^2 + \sum_{i=1}^{n_t} (\xi_i^t)^2 \right) + \frac{C_u}{2} \sum_{i=1}^{n_u} (\xi_i^u)^2 \\
 \text{s.t.} \quad & y_i^s (\mathbf{w}' \varphi_s(\mathbf{x}_i^s) + b) \geq \rho - \xi_i^s, \\
 & y_i^t (\mathbf{w}' \varphi_t(\mathbf{x}_i^t) + b) \geq \rho - \xi_i^t, \\
 & y_i^u (\mathbf{w}' \varphi_t(\mathbf{x}_i^u) + b) \geq \rho - \xi_i^u, \\
 & \mathbf{1}' \mathbf{y}_u = \delta, \quad \|\mathbf{P}\|_F^2 \leq \lambda_p, \quad \|\mathbf{Q}\|_F^2 \leq \lambda_q,
 \end{aligned}$$

where $\varphi_s(\cdot)$ and $\varphi_t(\cdot)$ are defined in (5.1) for generating the augmented features, and the constraint $\mathbf{1}' \mathbf{y}_u = \delta$ is used as the prior information on the unlabeled data similarly as in Transductive SVM (T-SVM) [54]. We refer to the above method as Semi-supervised Heterogeneous Feature Augmentation, or SHFA in short.

Similarly as in HFA, we only discuss the nonlinear case for SHFA here, and the linear case can be derived analogously. Let us define a kernel matrix

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_s & \mathbf{O}_{n_s \times (n_t + n_u)} \\ \mathbf{O}_{(n_t + n_u) \times n_s} & \mathbf{K}_t \end{bmatrix} \in \mathbb{R}^{n \times n}$$

where $\mathbf{K}_s \in \mathbb{R}^{n_s \times n_s}$ is the kernel of source domain samples and $\mathbf{K}_t \in \mathbb{R}^{(n_t + n_u) \times (n_t + n_u)}$ is the kernel of target domain samples. Then, by defining a nonlinear transformation metric $\mathbf{H} \in \mathbb{R}^{n \times n}$, we can derive the dual form of (5.14) as follows:

$$\begin{aligned}
 \min_{\mathbf{y} \in \mathcal{Y}, \mathbf{H} \succeq \mathbf{0}} \quad & \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad -\frac{1}{2} \boldsymbol{\alpha}' (\mathbf{Q}_{\mathbf{H}, \mathbf{y}} + \mathbf{D}) \boldsymbol{\alpha} \\
 \text{s.t.} \quad & \text{tr}(\mathbf{H}) \leq \lambda,
 \end{aligned} \tag{5.14}$$

where $\mathbf{Q}_{\mathbf{H}, \mathbf{y}} = (\mathbf{S}(\mathbf{H} + \mathbf{I})\mathbf{S} + \mathbf{1}\mathbf{1}') \circ (\mathbf{y}\mathbf{y}') \in \mathbb{R}^{n \times n}$, $\mathbf{y} = [\mathbf{y}'_s, \mathbf{y}'_t, \mathbf{y}'_u]'$ is the label vector in which $\mathbf{y}_s$ and $\mathbf{y}_t$ are given and $\mathbf{y}_u$ is unknown, $\mathcal{Y} = \{\mathbf{y} \in \{-1, +1\}^n | \mathbf{y} = [\mathbf{y}'_s, \mathbf{y}'_t, \mathbf{y}'_u]', \mathbf{1}' \mathbf{y}_u = \delta\}$ is the domain of $\mathbf{y}$, $\boldsymbol{\alpha} = [\alpha_1^s, \dots, \alpha_{n_s}^s, \alpha_1^t, \dots, \alpha_{n_t}^t, \alpha_1^u, \dots, \alpha_{n_u}^u]'$ is the dual variables corresponding to the constraints for source

samples, labeled target samples and unlabeled target samples, respectively, $\mathcal{A} = \{\boldsymbol{\alpha} | \boldsymbol{\alpha} \geq \mathbf{0}, \mathbf{1}'\boldsymbol{\alpha} = 1\}$ is the domain of $\boldsymbol{\alpha}$ and $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with the diagonal elements as $\frac{1}{C}$ for the labeled data from both domains and $\frac{1}{C_u}$ for the unlabeled target data.

5.3.2 Convex Relaxation

Compared with HFA, one major challenge in (5.14) is that we need to infer the optimal label vector $\mathbf{y}$, which is a mixed integer programming (MIP) problem. It is an NP problem and is computationally expensive to be solved [37, 46, 146] because there are possibly an exponential number of feasible labeling candidates $\mathbf{y}$'s. Inspired by [37, 46, 146], instead of directly finding the optimal labeling $\mathbf{y}$, we seek for an optimal linear combination of the feasible labeling candidates $\mathbf{y}$'s, which leads to a lower-bound of the original optimization problem as described below.

Proposition 1 *The objective of (5.14) is lower-bounded by the optimum of the following optimization problem:*

$$\begin{aligned} \min_{\boldsymbol{\gamma} \in \mathcal{D}_{\boldsymbol{\gamma}}, \mathbf{H} \succeq \mathbf{0}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} & -\frac{1}{2} \boldsymbol{\alpha}' \left(\sum_l \gamma_l \mathbf{Q}_{\mathbf{H}, \mathbf{y}_l} + \mathbf{D} \right) \boldsymbol{\alpha} \\ \text{s.t.} & \text{tr}(\mathbf{H}) \leq \lambda, \end{aligned} \quad (5.15)$$

where $\mathbf{y}_l$ is the l -th feasible labeling candidate, $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_{|\mathcal{Y}|}]'$ is the coefficient vector for the linear combination of all feasible labeling candidates and $\mathcal{D}_{\boldsymbol{\gamma}} = \{\boldsymbol{\gamma} | \boldsymbol{\gamma} \geq \mathbf{0}, \mathbf{1}'\boldsymbol{\gamma} \leq 1\}$ is the domain of $\boldsymbol{\gamma}$.

Proof: The proof is provided in the Appendix.

Another challenge in (5.14) or (5.15) is to solve the positive semi-definite matrix $\mathbf{H}$. We apply a similar strategy here as used in HFA to solve the optimization problem in (5.15). Specifically, we decompose $\mathbf{H}$ into a linear combination of a set of rank-one PSD matrices, *i.e.*, $\mathbf{H} = \sum_{r=1}^{\infty} \theta_r \mathbf{M}_r$ where $\mathbf{M}_r \in \mathbb{R}^{n \times n}$ is a rank-one PSD matrix and θ_r is the corresponding combination coefficient, which leads to the following optimization problem:

$$\min_{\boldsymbol{\gamma} \in \mathcal{D}_{\boldsymbol{\gamma}}} \min_{\boldsymbol{\theta} \in \mathcal{D}_{\boldsymbol{\theta}}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} -\frac{1}{2} \boldsymbol{\alpha}' \left(\sum_r \sum_l \theta_r \gamma_l \mathbf{Q}_{\mathbf{M}_r, \mathbf{y}_l} + \mathbf{D} \right) \boldsymbol{\alpha} \quad (5.16)$$

where $\mathbf{Q}_{\mathbf{M}_r, \mathbf{y}_l} = (\mathbf{S}(\lambda \mathbf{M}_r + \mathbf{I})\mathbf{S} + \mathbf{1}\mathbf{1}') \circ (\mathbf{y}_l \mathbf{y}_l')$ and $\mathcal{D}_\theta = \{\theta | \theta \geq 0, \mathbf{1}'\theta \leq 1\}$.

However, (5.16) contains three variables, θ, γ and α . To efficiently solve this problem, we propose a relaxation by combining θ and γ into one variable $\mathbf{d}$. Specifically, let us denote $d_k = \theta_r \gamma_l$ where d_k is the k -th entry of $\mathbf{d}$. After combining the two indices r and l into one index k , we assign $\mathbf{M}_k$ as $\mathbf{M}_r$ and $\mathbf{y}_k$ as $\mathbf{y}_l$, respectively. We have $\mathbf{1}'\mathbf{d} = \sum_k d_k = \sum_r \sum_l \theta_r \gamma_l = (\mathbf{1}'\theta)(\mathbf{1}'\gamma) \leq 1$. Then we reformulate the optimization problem in (5.16) as:

$$\min_{\mathbf{d} \in \mathcal{D}_\mathbf{d}} \max_{\alpha \in \mathcal{A}} -\frac{1}{2} \alpha' \left(\sum_k d_k \mathbf{Q}_{\mathbf{M}_k, \mathbf{y}_k} + \mathbf{D} \right) \alpha \quad (5.17)$$

where $\mathbf{Q}_{\mathbf{M}_k, \mathbf{y}_k} = (\mathbf{S}(\lambda \mathbf{M}_k + \mathbf{I})\mathbf{S} + \mathbf{1}\mathbf{1}') \circ (\mathbf{y}_k \mathbf{y}_k')$ and $\mathcal{D}_\mathbf{d} = \{\mathbf{d} | \mathbf{1}'\mathbf{d} \leq 1, \mathbf{d} \geq 0\}$.

Hence, we obtain an MKL problem as in (5.17) where each base kernel is $\mathbf{Q}_{\mathbf{M}_k, \mathbf{y}_k}$, and the primal form of (5.17) is as follows:

$$\begin{aligned} \min_{\mathbf{d}, \mathbf{w}_k, \rho, \xi_i} \quad & \frac{1}{2} \left(\sum_k \frac{\|\mathbf{w}_k\|^2}{d_k} + C \sum_{i=1}^n \nu_i (\xi_i)^2 \right) - \rho \\ \text{s.t.} \quad & \sum_k \mathbf{w}'_k \psi_k(\mathbf{x}_i) \geq \rho - \xi_i, \\ & \mathbf{1}'\mathbf{d} \leq 1, \quad \mathbf{d} \geq 0, \end{aligned} \quad (5.18)$$

where $\mathbf{d}$ is the coefficient vector, $\psi_k(\cdot)$ is the k -th feature mapping function induced by the kernel $\mathbf{Q}_{\mathbf{M}_k, \mathbf{y}_k} = (\mathbf{S}(\lambda \mathbf{M}_k + \mathbf{I})\mathbf{S} + \mathbf{1}\mathbf{1}') \circ (\mathbf{y}_k \mathbf{y}_k')$, and ν_i is the weight for the i -th sample which is 1 for labeled data from both domains and C_u/C for unlabeled target data.

5.3.3 Solution

Similar to HFA, there are also an infinite number of base kernels in (5.17). We therefore employ the cutting-plane algorithm to iteratively select a small set of active kernels. We first write the dual form of (5.18) as follows:

$$\begin{aligned} \max_{\tau, \alpha \in \mathcal{A}} \quad & -\tau \\ \text{s.t.} \quad & \frac{1}{2} \alpha' (\mathbf{Q}_{\mathbf{M}_k, \mathbf{y}_k} + \mathbf{D}) \alpha \leq \tau, \quad \forall k \end{aligned} \quad (5.19)$$

where we have an infinite number of constraints. The subproblem for selecting the most active kernel is:

$$\max_{\mathbf{y} \in \mathcal{Y}, \mathbf{M} \in \mathcal{M}} \frac{1}{2} \boldsymbol{\alpha}' \mathbf{Q}_{\mathbf{M}, \mathbf{y}} \boldsymbol{\alpha}, \quad (5.20)$$

where $\mathbf{Q}_{\mathbf{M}, \mathbf{y}} = (\mathbf{S}(\lambda \mathbf{M} + \mathbf{I})\mathbf{S} + \mathbf{1}\mathbf{1}') \circ (\mathbf{y}\mathbf{y}')$. Note that we do not need to consider the constant term $\boldsymbol{\alpha}' \mathbf{D} \boldsymbol{\alpha}$ in the above formulation when selecting the most active kernel.

Given any $\mathbf{y}$, finding the violated $\mathbf{M}$ is as the same as in HFA. It can be obtained by solving (5.12) with the closed form solution $\mathbf{M} = \mathbf{h}\mathbf{h}'$ where $\mathbf{h} = \frac{\mathbf{S}(\boldsymbol{\alpha} \circ \mathbf{y})}{\|\mathbf{S}(\boldsymbol{\alpha} \circ \mathbf{y})\|}$. Then we substitute $\mathbf{M}$ back into (5.20) and obtain

$$\begin{aligned} & \max_{\mathbf{y} \in \mathcal{Y}, \mathbf{M} \in \mathcal{M}} \frac{1}{2} \boldsymbol{\alpha}' \mathbf{Q}_{\mathbf{M}, \mathbf{y}} \boldsymbol{\alpha}, \\ &= \max_{\mathbf{y} \in \mathcal{Y}, \mathbf{M} \in \mathcal{M}} \frac{1}{2} (\boldsymbol{\alpha} \circ \mathbf{y})' (\mathbf{S}(\lambda \mathbf{M} + \mathbf{I})\mathbf{S} + \mathbf{1}\mathbf{1}') (\boldsymbol{\alpha} \circ \mathbf{y}), \\ &= \max_{\mathbf{y} \in \mathcal{Y}} \lambda \frac{(\boldsymbol{\alpha} \circ \mathbf{y})' \mathbf{K} (\boldsymbol{\alpha} \circ \mathbf{y}) (\boldsymbol{\alpha} \circ \mathbf{y})' \mathbf{K} (\boldsymbol{\alpha} \circ \mathbf{y})}{(\boldsymbol{\alpha} \circ \mathbf{y})' \mathbf{K} (\boldsymbol{\alpha} \circ \mathbf{y})} \\ & \quad + (\boldsymbol{\alpha} \circ \mathbf{y})' (\mathbf{K} + \mathbf{1}\mathbf{1}') (\boldsymbol{\alpha} \circ \mathbf{y}) \\ &= \max_{\mathbf{y} \in \mathcal{Y}} (\boldsymbol{\alpha} \circ \mathbf{y})' ((\lambda + 1)\mathbf{K} + \mathbf{1}\mathbf{1}') (\boldsymbol{\alpha} \circ \mathbf{y}), \end{aligned} \quad (5.21)$$

which indicates that we only need to solve an optimization problem on $\mathbf{y}$. However, it is another MIP problem, and is difficult to be solved. Similar to [37, 46], we employ an approximated solution to (5.21) for finding the most violated $\mathbf{y}$. Specifically, we first rewrite (5.21) as:

$$\max_{\mathbf{y} \in \mathcal{Y}} \mathbf{y}' \left(\tilde{\mathbf{K}} \circ (\boldsymbol{\alpha} \boldsymbol{\alpha}') \right) \mathbf{y} = \max_{\mathbf{y} \in \mathcal{Y}} \left\| \sum_i y_i \alpha_i \tilde{\phi}(\mathbf{x}_i) \right\|^2 \quad (5.22)$$

where $\tilde{\mathbf{K}} = (\lambda + 1)\mathbf{K} + \mathbf{1}\mathbf{1}'$ and $\tilde{\phi}(\cdot)$ is the feature mapping function induced by $\tilde{\mathbf{K}}$. Following [37, 46], we use the ℓ_∞ -norm to approximate the ℓ_2 -norm in (5.22), and the problem becomes

$$\begin{aligned} & \max_{\mathbf{y} \in \mathcal{Y}} \left\| \sum_i y_i \alpha_i \tilde{\phi}(\mathbf{x}_i) \right\|_\infty \\ &= \max_{\mathbf{y} \in \mathcal{Y}} \max_{j=1, \dots, \tilde{d}} \left\{ \sum_i y_i \alpha_i z_{ij}, - \sum_i y_i \alpha_i z_{ij} \right\} \\ &= \max_{j=1, \dots, \tilde{d}} \left\{ \max_{\mathbf{y} \in \mathcal{Y}} \sum_i y_i \alpha_i z_{ij}, \max_{\mathbf{y} \in \mathcal{Y}} - \sum_i y_i \alpha_i z_{ij} \right\} \end{aligned} \quad (5.23)$$

where z_{ij} is the j -th entry of the feature vector $\tilde{\phi}(\mathbf{x}_i) = [z_{i1}, \dots, z_{i\tilde{d}}]'$ with $\tilde{d}$ as the feature dimension.

To find the optimal $\mathbf{y}$, we first obtain $\tilde{\phi}(\mathbf{x})$ by using SVD decomposition on the kernel matrix $\tilde{\mathbf{K}}$, which is also known as the empirical kernel map [160]. Then we calculate $\alpha_i z_{ij}$ for each feature dimension and each sample. For the j -th dimension, we can respectively obtain two label vectors by a simple sorting operation to solve the two inner problems in (5.23). Specifically, we first sort the unlabeled samples in descending order according to $\alpha_i z_{ij}$. For $\max_{\mathbf{y} \in \mathcal{Y}} \sum_i y_i \alpha_i z_{ij}$, the optimal label vector can be obtained by setting the first $(\delta + n_u)/2$ unlabeled samples as positive and the remaining unlabeled samples as negative; similarly for $\max_{\mathbf{y} \in \mathcal{Y}} -\sum_i y_i \alpha_i z_{ij}$, the optimal label vector is obtained by setting the last $(\delta + n_u)/2$ unlabeled samples as positive and remaining unlabeled samples as negative. Finally, the most violated $\mathbf{y}$ is the label vector with the maximum objective value among these $2\tilde{d}$ label vectors.

We summarize the algorithm for solving SHFA in Algorithm 7. We first initialize the set of rank-one PSD matrices $\mathcal{M}$ with $\mathbf{M}_1 = \mathbf{h}_1 \mathbf{h}_1'$, and also initialize the labeling candidate set $\mathcal{Y}$ by using a feasible label vector $\mathbf{y}_1$. To obtain $\tilde{\mathbf{y}}_u$ in $\mathbf{y}_1$, we first sort the unlabeled training samples in descending order according to the prediction of the classifier trained on the labeled target samples. Then $\tilde{\mathbf{y}}_u$ is obtained by setting the first $(\delta + n_u)/2$ unlabeled samples as positive and the remaining samples as negative. Next, we solve the MKL problem in (5.17) based on $\mathcal{Y}$ and $\mathcal{M}$. After that, we find a violated $\mathbf{y}$ and calculate the corresponding $\mathbf{M} = \mathbf{h} \mathbf{h}'$ where $\mathbf{h} = \frac{\mathbf{S}(\boldsymbol{\alpha} \circ \mathbf{y})}{\|\mathbf{S}(\boldsymbol{\alpha} \circ \mathbf{y})\|}$. We respectively add $\mathbf{y}$ and $\mathbf{M}$ into $\mathcal{Y}$ and $\mathcal{M}$ and solve the MKL problem again. This process is repeated until convergence. The time complexity can be analyzed similarly as in HFA, which is $O(n^3 + TLn^{2.3})$ with $n = n_s + n_t + n_u$ being the total number of training samples².

After obtaining the optimal solution $\boldsymbol{\alpha}$, $\mathbf{d}$, $\mathcal{Y}$ and $\mathcal{M}$, we can predict any test sample $\mathbf{x}$ from the target domain by using the following target decision function:

$$f(\mathbf{x}) = \sum_k d_k (\boldsymbol{\alpha} \circ \mathbf{y}_k)' \mathbf{K}^{\frac{1}{2}} (\lambda \mathbf{M}_k + \mathbf{I}) \begin{bmatrix} \mathbf{O}_{n_s \times (n_t + n_u)} \\ \mathbf{K}_t^{-\frac{1}{2}} \end{bmatrix} \mathbf{k}_t + b, \quad (5.24)$$

where $\mathbf{k}_t = [k(\mathbf{x}_1^t, \mathbf{x}), \dots, k(\mathbf{x}_{n_t}^t, \mathbf{x}), k(\mathbf{x}_1^u, \mathbf{x}), \dots, k(\mathbf{x}_{n_u}^u, \mathbf{x})]'$ and $k(\mathbf{x}_i, \mathbf{x}_j) = \phi_t(\mathbf{x}_i)' \phi_t(\mathbf{x}_j)$ is a predefined kernel function for two data samples $\mathbf{x}_i$ and $\mathbf{x}_j$ in the target domain.

²The time complexity of the sorting operation for $\tilde{d}$ times in finding the optimal $\mathbf{y}$ is $\tilde{d} n_u \log(n_u)$, which is less than $n^2 \log(n)$. When the number of training samples (*i.e.*, n) is large as in our experiments, it can be ignored when compared with the time complexity $O(Ln^{2.3})$ for solving the MKL problem.

Algorithm 7 Semi-supervised Heterogeneous Feature Augmentation

Require: Labeled source samples $\{(\mathbf{x}_i^s, y_i^s)|_{i=1}^{n_s}\}$, labeled target samples $\{(\mathbf{x}_i^t, y_i^t)|_{i=1}^{n_t}\}$, and unlabeled target samples $\{(\mathbf{x}_i^u, y_i^u)|_{i=1}^{n_u}\}$ with the unknown y_i^u 's.

- 1: Train an SVM classifier f_0 by only using the labeled target samples.
- 2: Initialize the labeling candidate set $\mathcal{Y} = \{\mathbf{y}_1\}$ where $\mathbf{y}_1 = [\mathbf{y}'_s, \mathbf{y}'_t, \tilde{\mathbf{y}}'_u]'$ where $\tilde{\mathbf{y}}_u$ is a feasible label vector obtained by using the prediction of f_0 .
- 3: Initialize the rank-one matrices set $\mathcal{M} = \{\mathbf{M}_1\}$ with $\mathbf{M}_1 = \mathbf{h}_1 \mathbf{h}'_1$ and $\mathbf{h}_1 = \frac{1}{\sqrt{n}} \mathbf{1}_n$ and set $k = 1$.
- 4: **repeat**
- 5: Set $k = k + 1$.
- 6: Solve $\mathbf{d}$ and $\boldsymbol{\alpha}$ in (5.17) based on $\mathcal{Y}$ and $\mathcal{M}$ by using the existing MKL solver [72].
- 7: Find the violated $\mathbf{y}_k$ by solving (5.23) and obtain $\mathbf{M}_k = \mathbf{h} \mathbf{h}'$ where $\mathbf{h} = \frac{\mathbf{S}(\boldsymbol{\alpha} \odot \mathbf{y}_k)}{\|\mathbf{S}(\boldsymbol{\alpha} \odot \mathbf{y}_k)\|}$.
- 8: Set $\mathcal{M} = \mathcal{M} \cup \{\mathbf{M}_k\}$, $\mathcal{Y} = \mathcal{Y} \cup \{\mathbf{y}_k\}$.
- 9: **until** The objective converges.

Ensure: $\boldsymbol{\alpha}$, $\mathbf{d}$, $\mathcal{Y}$ and $\mathcal{M}$.

5.3.4 ℓ_p -MKL Extension

Recall that we have formulated our SHFA as an MKL problem in (5.18), in which ℓ_1 -norm constraint on the kernel coefficient vector $\mathbf{d}$ (*i.e.* $\|\mathbf{d}\|_1 \leq 1$) is adopted. However, the optimization problem in (5.18) can be extended to more general ℓ_p -MKL by using ℓ_p -norm on $\mathbf{d}$ (*i.e.* $\|\mathbf{d}\|_p \leq 1$) as follows:

$$\begin{aligned}
 \min_{\mathbf{d}, \mathbf{w}_k, \rho, \xi_i} \quad & \frac{1}{2} \left(\sum_k \frac{\|\mathbf{w}_k\|^2}{d_k} + C \sum_{i=1}^n \nu_i (\xi_i)^2 \right) - \rho \\
 \text{s.t.} \quad & \sum_k \mathbf{w}'_k \psi_k(\mathbf{x}_i) \geq \rho - \xi_i, \\
 & \|\mathbf{d}\|_p \leq 1, \quad \mathbf{d} \geq 0,
 \end{aligned} \tag{5.25}$$

where $\mathbf{d}$, $\psi_k(\mathbf{x}_i)$ and ν_i are as the same as defined in (5.18). Thus, the original SHFA is a special case of (5.25) when $p = 1$. The ℓ_p -MKL problem in (5.25) can also be solved by Algorithm 7. The only difference is that we solve an ℓ_p -MKL problem instead of ℓ_1 -MKL in Step 6.

Intuitively, setting a larger $p > 1$ will lead to a denser solution of $\mathbf{d}$, namely, there are possibly more non-zeros elements in $\mathbf{d}$ when setting $p > 1$ instead of $p = 1$. Considering that we have merge the two coefficient vector $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ into one vector $\mathbf{d}$, so using the ℓ_p -norm on $\mathbf{d}$ is approximately imposing ℓ_p -norm on $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$. We discuss the implications of using ℓ_p -norm on $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ as follows.

Note the trace norm discussed in (5.7) is also known as the Schatten- p norm when $p = 1$. It can be easily verified that Theorem 5.2 also holds, if we replace the ℓ_1 norm on $\boldsymbol{\theta}$ as ℓ_p norm, and also replace the trace norm on $\mathbf{H}$ in (5.7) as Schatten- p norm. Therefore, imposing ℓ_p -norm on $\boldsymbol{\theta}$ implies that we apply a different regularizer (*i.e.*, the Schatten- p norm) on the transformation metric $\mathbf{H}$. On the other hand, since that $\boldsymbol{\gamma}$ is the combination coefficient vector for combining different labeling candidates $\mathbf{y}_l$'s, using ℓ_p -norm with $p \geq 1$ indicates that the combination coefficients of different labeling candidates are more similar to each other when compared with using ℓ_1 -norm. So it is more favorable to set $p \geq 1$ when those labeling candidates are not very diverse.

5.4 Experiments

In this section, we evaluate our proposed HFA and SHFA methods for object recognition, multilingual text categorization and cross-lingual sentiment classification. We focus on the heterogeneous domain adaptation problem with only one source domain and one target domain. For the supervised heterogeneous domain adaptation setting, we only utilize a limited number of labeled training samples in the target domain; for the semi-supervised heterogeneous domain adaptation setting, we additionally employ a large number of unlabeled training samples in the target domain.

5.4.1 Setup

Object recognition: We employ a recently released Office dataset³ used in [127, 155] for this task. This dataset contains a total of 4106 images from 31 categories collected from three sources: **amazon** (object images downloaded from Amazon), **dslr** (high-resolution images taken from a digital SLR camera) and **webcam** (low-resolution images taken from a web camera). We follow the same protocols in the previous work [127]. Specifically, SURF features [161] are extracted for all the images. The images from **amazon** and **webcam** are clustered into 800 visual words by using k-means. After vector quantization, each image is represented as a 800 dimensional histogram feature. Similarly, we represent each image from **dslr** as a 600-dimensional histogram feature.

³<http://www.icsi.berkeley.edu/~saenko/projects.html>

Table 5.1: Summarization of the object dataset including 31 categories.

Domain	Source		Target
	amazon	webcam	dslr
# dimension	800	800	600
# total samples	2,813	795	498
# labeled samples per class	20	8	3
# unlabeled samples	—	—	405

In the experiments, **dslr** is used as the target domain, while **amazon** and **webcam** are considered as two individual source domains. We strictly follow the setting in [127, 155] and randomly select 20 (resp., 8) training images per category for the source domain **amazon** (resp., **webcam**). For the target domain **dslr**, 3 training images are randomly selected from each category, and the remaining **dslr** images are used for testing, which are also used as the unlabeled training samples in the semi-supervised setting. See Table 5.1 for a summarization of this dataset.

Text categorization: We use the Reuters multilingual dataset⁴ [162], which is collected by sampling parts of the Reuters RCV1 and RCV2 collections. It contains about 11K newswire articles from 6 classes (*i.e.*, C15, CCAT, E21, ECAT, GCAT and M11) in 5 languages (*i.e.*, **English**, **French**, **German**, **Italian** and **Spanish**). While each document was also translated into the other four languages in this dataset, we do not use the translated documents in this work. All documents are represented by using the TF-IDF feature.

We take **Spanish** as the target domain in the experiment and use each of the other four languages as an individual source domain. For each class, we randomly sample 100 training documents from the source domain and m training documents from the target domain, where $m = 5, 10, 15$ and 20 . And the remaining documents in the target domain are used as the test data, among which 3,000 documents are additionally sampled as the unlabeled training data in the semi-supervised setting. Note that the method in [126] cannot handle the original high dimensional TF-IDF features. In order to fairly compare our HFA method [126], for documents written in each language, we perform PCA based on the TF-IDF features with 60% energy preserved. We summarize this dataset in Table 5.2.

⁴<http://multilingreuters.iit.nrc.ca/ReutersMultiLingualMultiView.htm>

Table 5.2: Summarization of the Reuters multilingual dataset including 6 classes.

Domain	Source				Target
	English	French	German	Italian	Spanish
# dim after PCA	1,131	1,230	1,417	1,041	807
# total samples	18,758	25,468	29,953	24,039	11,547
# labeled samples per class	100	100	100	100	5/10/15/20
# unlabeled samples	—	—	—	—	3,000

Sentiment Classification: We use the Cross-Lingual Sentiment (CLS) dataset⁵ [121], which is an extended version of the Multi-Domain Sentiment Dataset [151] widely used for domain adaptation. It is collected from Amazon and contains about 800,000 reviews of three product categories: Books, DVDs and Music, and written in four languages: English, German, French, and Japanese. The English reviews were sampled from the Multi-Domain Sentiment Dataset and reviews in other languages are crawled from Amazon. For each category and each language, the dataset is officially partitioned into a training set, a test set and an unlabeled set, where the training set and test set consist of 2,000 reviews, and the numbers of unlabeled reviews vary from 9,000 to 170,000.

We take English as the source domain and each of the other three languages as an individual target domain in the experiment. We randomly sample 500 reviews from the training set of the source domain and 100 reviews from the training set of the target domain as the labeled data. The test set is the official test set for each category and each language. We also sample 1,000 reviews from the unlabeled set as the unlabeled target training data. Similarly as for text categorization, we extracted the TF-IDF features and perform PCA with 60% energy preserved. The complete information of this dataset is summarized in Table 5.3.

Baselines: To evaluate our proposed methods, HFA and SHFA, we compare them with a number of baselines under two settings. The first setting (*i.e.*, the supervised HDA setting) is as the same as [24], in which there are sufficient labeled source samples and a limited number of labeled target samples. As the source and target data have different dimensions, they cannot be directly combined to train any classifiers for the target domain. So the baseline algorithms in this setting are listed as follows:

⁵<http://www.uni-weimar.de/cms/medien/webis/research/corpora/corpus-webis-cls-10.html>

Table 5.3: Summarization of the Cross-Lingual Sentiment dataset including 3 categories and 2 classes.

Domain	Source	Target		
	English	French	German	Japanese
# dim after PCA	715	964	929	874
# labeled samples	500	100	100	100
# unlabeled samples	—	1,000	1,000	1,000

- **SVM-T**: It utilizes the labeled samples only from the target domain to train a standard SVM classifier for each category/class. This is a naive approach without considering the information from the source domain.
- **HeMap [125]**: It finds the projection matrices for a common feature subspace as well as learns the optimally projected data from both domains. We align the samples from different domains according to their labels. Since HeMap requires the same number of samples from the source and target domains, we randomly select $\min\{n_s, n_t\}$ samples from each domain for learning the subspace.
- **DAMA [126]**: It learns a common feature subspace by utilizing the class labels of the source and target training data for manifold alignment.
- **ARC-t [127]**: It uses the labeled training data from both domains to learn an asymmetric transformation metric between different feature spaces.

In the second setting (*i.e.* the semi-supervised HDA setting), we additionally employ the unlabeled samples in the target domain. To evaluate our SHFA, we report the results of one more baseline, transductive SVM (T-SVM) [54], which utilizes both the labeled data and unlabeled data to train the classifier. Note that the labeled samples in the source domain cannot be used in T-SVM because they have different features with the samples in the target domain. Moreover, all the above heterogeneous domain adaptation methods [125–127] were designed for the supervised heterogeneous domain adaptation scenario, so it is unclear how to utilize the unlabeled target data to learn the projection matrices or transformation metric for these methods.

For HeMap and DAMA, after learning the projection matrices, we apply SVM to train the final classifiers by using the projected training data from both domains for a

given category/class. For ARC-t, we construct the kernel matrix based on the learned asymmetric transformation metric, and then SVM is also applied to train its final classifier. The RBF kernel is used for all methods with the bandwidth parameter as the mean distance of all training samples. As we only have a very limited number of labeled training samples in the target domain, the cross-validation technique cannot be effectively employed to determine the optimal parameters. Therefore, we set the tradeoff parameter in SVM as the default value $C = 1$ for all methods. For our HFA and SHFA methods, we empirically fix the parameter λ as 100 in the vision application (*i.e.* the object recognition) and 1 in the text applications (*i.e.*, document classification and sentiment classification). And we also empirically set the weight of unlabeled data C_u in SHFA as 10^{-3} for all experiments. Moreover, we additionally report the results of our SHFA with the ℓ_p -MKL extension (see Section 5.3.4) where we empirically set $p = 1.5$ for all the datasets which generally achieves better results.

For other methods, we report their best results on the test data by varying their parameters in a wide range on each dataset. Specifically, we validate the parameters β in HeMap (see Equation (1) in [125]), μ in DAMA (see Theorem 1 in [126]) and λ in ARC-t (see Equation (1) in [127]) from $\{0.01, 0.1, 1, 10, 100\}$. For T-SVM, we validate the weight of unlabeled data C_u from $\{0.001, 0.01, 0.1, 1\}$ and the parameter s for the ramp loss from $[-0.9, 0]$ with the step size as 0.1. For both T-SVM and our SHFA, we set the parameter δ for the balance constraint on unlabeled samples using the prior information.

Evaluation metric: Following [127], for each method we measure the classification accuracy over all categories/classes on three datasets. We randomly sample the training data for a fixed number of times (*i.e.*, 20 for the Office dataset as in [127], and 10 for the Reuters dataset and the Cross-Lingual Sentiment dataset) and report the mean classification accuracies of all methods over all rounds of experiments.

5.4.2 Classification Results

Object recognition: We report the means and standard deviations of classification accuracies for all methods on the Office dataset [155] in Table 5.4. From the results, we have the following observations in terms of the mean classification accuracy. SVM-T outperforms HeMap by using only 3 labeled training samples from the target domain.

Table 5.4: Means and standard deviations of classification accuracies (%) of all methods on the object dataset by using SURF features. Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.

Methods	Source Domain	
	amazon	webcam
SVM_T	52.9 ± 3.1	
HeMap	42.8 ± 2.4	42.2 ± 2.6
DAMA	53.3 ± 2.3	53.2 ± 3.2
ARC-t	53.1 ± 2.4	53.0 ± 3.2
HFA	55.4 ± 2.9	54.3 ± 3.6
T-SVM	53.5 ± 2.0	
SHFA ($p = 1$)	56.1 ± 2.9	55.1 ± 3.4
SHFA ($p = 1.5$)	56.6 ± 2.4	55.9 ± 3.3

The explanation is that HeMap does not explicitly utilize the label information of the target training data to learn the feature mapping matrices. As a result, the learned common subspace cannot well preserve a similar data structure as in the original feature spaces of the source and target data, which results in poor classification performances. DAMA performs only slightly better than SVM_T, possibly due to the lack of strong manifold structure on this dataset. Both results of ARC-t implemented by ourselves and reported in [127] are only comparable with those of SVM_T, which shows that ARC-t is less effective for HDA on this dataset. Our HFA outperforms the other methods for both cases, which clearly demonstrate the effectiveness of our proposed method for HDA by learning with augmented features. Moreover, we also observe that it is beneficial to additionally use unlabeled data in the target domain to learn a more robust classifier. Specifically, when setting the parameter p in the ℓ_p -norm regularizer of ℓ_p -MKL as $p = 1$, our SHFA outperforms HFA on both cases when amazon and webcam are used as the source domain. When setting $p = 1.5$, the improvements of SHFA over HFA are 1.2% and 1.6%, respectively. SHFA also outperforms T-SVM which demonstrates we can train a better classifier by learning the transformation metric $\mathbf{H}$ to effectively exploit the source data in SHFA.

Object recognition using DeCAF features: Recently, deep learning features [163] have shown impressive performances on many computer vision tasks recently. We also conduct experiments by using the DeCAF features on object recognition task. We extract

Table 5.5: Means and standard deviations of classification accuracies (%) of all methods on the object dataset by using DeCAF features. Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.

Methods	Source Domain	
	amazon	webcam
SVM_T	78.2 ± 2.8	
HeMap	60.5 ± 4.2	64.1 ± 3.8
DAMA	79.0 ± 2.6	78.4 ± 2.7
ARC-t	78.8 ± 2.4	78.3 ± 2.7
HFA	81.4 ± 2.0	80.1 ± 2.4
T-SVM	79.9 ± 2.9	
SHFA ($p = 1$)	82.6 ± 2.4	81.9 ± 2.6
SHFA ($p = 1.5$)	83.6 ± 2.3	83.1 ± 2.5

DeCAF⁶ features for the source domains, and also extract DeCAF⁷ features for the target domain. Other settings are the same as when using the SURF features. The means and standard deviations of classification accuracies for all methods by using DeCAF features are summarized in Table 5.5. Similarly as using the SURF features, we observe that our HFA algorithm outperforms other baseline methods in the supervised heterogeneous domain adaptation setting, and our SHFA algorithm obtains the best results when using $p=1.5$.

Text categorization: Table 5.6 shows the means and standard deviations of classification accuracies for all methods on the Reuters multilingual dataset [162] by using $m = 10$ and $m = 20$ labeled training samples per class from the target domain. We have the following observations in terms of the mean classification accuracy. SVM_T still outperforms HeMap, and DAMA and ARC-t perform better than SVM_T for most cases. Our proposed HFA method is better than other supervised HDA methods on this dataset. For the semi-supervised setting, T-SVM is even worse than SVM_T although we have tuned all the parameters in a wide range, possibly because T-SVM cannot effectively utilize these target unlabeled data on this dataset. However, our SHFA can effectively handle the unlabeled data in the target domain and the performance improvements of SHFA ($p = 1.5$) over HFA are 3.5%, 3.2%, 3.1%, 3.1% and 1.1%, 1.1%, 1.0%, 1.1% on these four different source domains when $m = 10$ and $m = 20$, respectively .

We also plot the classification results of SVM_T, DAMA, ARC-t and our methods HFA and SHFA by using different numbers of target training samples per class (*i.e.*,

Table 5.6: Means and standard deviations of classification accuracies (%) of all methods on the Reuters multilingual dataset by using 10 and 20 labeled training samples per class from the target domain **Spanish**. Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.

Methods	#target labeled samples per class = 10				#target labeled samples per class = 20			
	English	French	German	Italian	English	French	German	Italian
SVM.T	66.6 $\pm$ 3.7				72.6 $\pm$ 2.3			
HeMap	54.7 $\pm$ 7.4	55.0 $\pm$ 9.4	58.0 $\pm$ 7.9	59.4 $\pm$ 3.7	65.7 $\pm$ 3.1	64.2 $\pm$ 4.2	64.6 $\pm$ 3.6	65.8 $\pm$ 2.3
DAMA	65.0 $\pm$ 2.9	66.9 $\pm$ 2.1	67.5 $\pm$ 2.1	68.5 $\pm$ 2.8	72.4 $\pm$ 2.4	72.8 $\pm$ 2.0	72.9 $\pm$ 2.3	73.3 $\pm$ 2.1
ARC-t	65.7 $\pm$ 2.7	66.9 $\pm$ 1.7	68.7 $\pm$ 2.9	67.9 $\pm$ 2.8	72.9 $\pm$ 2.0	73.5 $\pm$ 1.8	74.7 $\pm$ 1.6	74.0 $\pm$ 2.0
HFA	68.6 $\pm$ 2.3	69.5 $\pm$ 1.9	69.8 $\pm$ 2.7	69.8 $\pm$ 2.5	75.3 $\pm$ 1.7	75.7 $\pm$ 1.6	76.1 $\pm$ 1.5	75.8 $\pm$ 1.8
T-SVM	63.3 $\pm$ 3.8				69.2 $\pm$ 2.2			
SHFA ($p = 1$)	70.7 $\pm$ 2.3	71.6 $\pm$ 2.3	72.0 $\pm$ 2.9	71.8 $\pm$ 2.6	75.8 $\pm$ 1.7	76.4 $\pm$ 1.5	76.8 $\pm$ 1.4	76.4 $\pm$ 1.8
SHFA ($p = 1.5$)	72.1 $\pm$ 2.2	72.7 $\pm$ 2.1	72.9 $\pm$ 2.5	72.9 $\pm$ 2.2	76.4 $\pm$ 1.6	76.8 $\pm$ 1.4	77.1 $\pm$ 1.3	76.9 $\pm$ 1.6

$m = 5, 10, 15$ and 20) for each source domain in Figure 5.2. We do not report the results of HeMap, as they are much worse than the other methods. From the results, the performances of all methods increase when using a larger m . And the two HDA methods DAMA and ARC-t generally achieve better mean classification accuracies than SVM-T except for the setting using **English** as the source domain. Our HFA method generally outperforms all other baseline methods according to mean classification accuracy. When using the unlabeled data in the target domain, our SHFA ($p = 1$) outperforms all existing HDA methods and the performance can be further improved when setting $p = 1.5$. We also observe that SHFA has large improvements over HFA when the number of labeled data in the target domain is very small (see $m = 5$ in Figure 5.2). When the number of labeled data in the target domain increases, the unlabeled data in the target domain is less helpful, but SHFA is still better than HFA.

Sentiment classification: Table 5.7 summarizes the means and standard deviations of classification accuracies for all methods on the Cross-Lingual Sentiment dataset by using $m = 100$ labeled training samples in the target domain. As in each domain there are three categories (*i.e.*, Books, DVDs, Music), each mean accuracy in Table 5.7 is the mean accuracy over three categories and ten rounds. We have the following observations in terms of the mean classification accuracy. We observe that HeMap is worse than SVM-T which again indicates it cannot learn good feature mapping on this dataset. ARC-t is only comparable with SVM-T, and DAMA outperform SVM-T for all cases. Our HFA is better than other baseline methods, except one exceptional case that HFA is worse than DAMA when using **Japanese** as the target domain. A possible explanation is the reviews in **Japanese** have good manifold correspondence with that in **English**. However, our HFA is still comparable with DAMA in this case. Moreover, we also have the similar observation as on the Office dataset and Reuters dataset, our SHFA achieves better results than HFA by additionally exploiting the unlabeled data in the target domain. With setting $p = 1$, the performance improvements of SHFA over HFA are 3.7%, 3.6% and 3.6% when using **German**, **French** and **Japanese** as the target domain, respectively. With setting $p = 1.5$, the performance improvements of SHFA over HFA are further increased to 4.4%, 4.7% and 4.4%, respectively.

Figure 5.2: Classification accuracies of all methods with respect to different number of target training samples per class (i.e., $m = 5, 10, 15$ and 20) on the Reuters multilingual dataset. **Spanish** is considered as the target domain, and in each subfigure the results are obtained by using one language as the source domain.

Table 5.7: Means and standard deviations of classification accuracies (%) of all methods on the Cross-Lingual Sentiment dataset by using 100 labeled training samples from the target domain. Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.

Methods	Target Domain		
	German	French	Japanese
SVM-T	58.3 ± 2.8	60.4 ± 3.9	57.4 ± 2.0
HeMap	50.4 ± 0.6	49.8 ± 0.6	51.3 ± 1.0
DAMA	64.6 ± 1.9	65.7 ± 1.8	64.4 ± 1.8
ARC-t	58.3 ± 3.0	59.4 ± 4.3	57.5 ± 1.9
HFA	66.5 ± 2.2	66.9 ± 2.1	64.2 ± 2.5
T-SVM	65.6 ± 2.4	67.8 ± 2.3	63.9 ± 2.6
SHFA ($p = 1$)	70.2 ± 1.9	70.5 ± 1.1	67.8 ± 1.2
SHFA ($p = 1.5$)	70.9 ± 1.7	71.6 ± 1.2	68.6 ± 1.3

5.4.3 Augmented Features *v.s.* Common Features

We defined two augmented feature mapping functions $\varphi_s(\mathbf{x}^s) = [(\mathbf{P}\mathbf{x}^s)', \mathbf{x}^{s'}, \mathbf{0}'_{d_t}]'$ and $\varphi_t(\mathbf{x}^t) = [(\mathbf{Q}\mathbf{x}^t)', \mathbf{0}'_{d_s}, \mathbf{x}^t]'$ in (5.1) by concatenating the feature representation in the learnt common subspace (referred to as *common features* here) with the original features and zeros. However, our methods are also applicable by only using the common feature representations $\mathbf{P}\mathbf{x}^s$ and $\mathbf{Q}\mathbf{x}^t$ for the samples from source and target domains without using the original features and zeros. We take SHFA when setting $p = 1.5$ as an example to evaluate our work by only using the feature representation in the common space, which is referred as *SHFA_commFeat*. The results on the Reuters multilingual dataset are shown in Table 5.8, where we use the same settings as described in Section 5.4.1. We observe that SHFA_commFeat still outperforms the existing HDA methods on all settings in terms of mean accuracy, which clearly demonstrates the effectiveness of our proposed learning scheme. Moreover, SHFA using the augmented features are consistently better than SHFA_commFeat in terms of mean accuracy, which demonstrates it is beneficial to use our proposed new learning methods with the augmented features for HDA.

5.4.4 Performance Variations using Different Parameters

We conduct experiments on the Reuters multilingual dataset to evaluate the performance variations of our SHFA by using different parameters (*i.e.*, λ , p , and C_u). As described

Figure 5.3: The performances of our SHFA using different parameters on the Reuters multilingual dataset.

in Section 5.4.1, we still use 100 labeled samples per class from the source domain, as well as 20 labeled samples per class and 3000 unlabeled samples from the target domain. The results of our SHFA ($p = 1$) and SHFA ($p = 1.5$) by using the default values $\lambda = 1$ and $C_u = 0.001$ have been reported in Table 5.6. To evaluate the performance variations, at each time we vary one parameter and set the other parameters as the default values (*i.e.*, $\lambda = 1$, $C_u = 0.001$, and $p = 1.5$). The means of classification accuracies by varying different parameters on the four settings are plotted in Fig 5.3.

From Fig 5.3, we observe that our SHFA is quite stable to these parameters in certain ranges. Specifically, by changing λ in the range of $[0.01, 100]$, the performances of SHFA ($p = 1.5$) vary within 1% in terms of mean classification accuracy, which are still better than these baseline methods reported in Table 5.6. Also, by changing the parameter p of the ℓ_p -norm in the range of $\{1, 1.2, 1.5, 2\}$, we observe that with a larger p , SHFA can achieve better results. However, our initial experiments show that a larger p usually leads to a slower convergence. We empirically set $p = 1.5$ as the default value in all our experiments for a good tradeoff between the effectiveness and efficiency. Moreover, we also evaluate our SHFA ($p = 1.5$) by varying C_u in the range of $[10^{-5}, 10^{-1}]$. The parameter C_u controls the weights of unlabeled samples. Intuitively, it should not be too large because the inferred labels for the unlabeled samples are not accurate, which is also supported by our experiment as shown in Fig 5.3(c). While we empirically set $C_u = 10^{-3}$ in all our experiments, we observe that SHFA ($p = 1.5$) using a larger value (*i.e.*, $C_u = 10^{-2}$) can achieve better results on this dataset. However, the performances drop dramatically when setting it to a much larger value (say, $C_u = 10^{-1}$). Nevertheless, our SHFA is generally stable and better than these baseline methods reported in Table 5.6

Table 5.8: Means and standard deviations of classification accuracies (%) of our SHFA ($p = 1.5$) and SHFA_commFeat ($p = 1.5$) on the Reuters multilingual dataset.

	#target labeled samples per class = 10					#target labeled samples per class = 20				
	English	French	German	Italian	English	French	German	Italian	English	French
SHFA ($p = 1.5$)	72.1 $\pm$ 2.2	72.7 $\pm$ 2.1	72.9 $\pm$ 2.5	72.9 $\pm$ 2.2	76.4 $\pm$ 1.6	76.8 $\pm$ 1.4	77.1 $\pm$ 1.3	76.9 $\pm$ 1.6	76.4 $\pm$ 1.6	76.5 $\pm$ 1.7
SHFA_commFeat ($p = 1.5$)	70.8 $\pm$ 2.2	71.6 $\pm$ 2.2	72.0 $\pm$ 2.8	71.8 $\pm$ 2.4	75.9 $\pm$ 1.7	76.5 $\pm$ 1.5	76.8 $\pm$ 1.4	76.5 $\pm$ 1.9	76.4 $\pm$ 1.6	76.5 $\pm$ 1.7

100

Table 5.9: The average per class training time (in seconds) comparisons of all methods on the Cross-Lingual Sentiment dataset.

	SVM-T	HeMap	DAMA	ARC-t	HFA	T-SVM	SHFA ($p = 1$)	SHFA ($p = 1.5$)
Training Time	0.01	0.32	9.68	1.98	3.16	20.70	18.13	43.34

when setting $C_u \in [10^{-5}, 10^{-2}]$. For the domain adaptation problem, it is difficult to perform cross-validation to choose the optimal parameters, because we usually only have a limited number of labeled samples in the target domain. We would like to study how to automatically decide the optimal parameters in the future.

5.4.5 Time Analysis

We take the Cross-Lingual Sentiment dataset as an example to evaluate the running time of all methods. The experimental setting is as the same as described in Section 5.4.1. The average per class training times of all methods are reported in Table 5.9. All the experiments are performed on a workstation with Xeon 3.33 GHz CPU and 16 GB of RAM. From Table 5.9, we observe that the supervised methods (*i.e.*, SVM-T, HeMap, DAMA, ARC-t and HFA) are generally faster than the semi-supervised methods (*i.e.*, T-SVM and our SHFA), because additional unlabeled samples are used in the semi-supervised methods. SVM-T is very fast because it only utilizes the labeled training data from the target domain. HeMap is fast since it only needs to solve the eigen-decomposition problem in a very small size due to the limited number of labeled samples in the target domain. The training time of HFA is comparable to that of DAMA and ARC-t. For the semi-supervised methods, we observe that our SHFA ($p = 1$) is faster than T-SVM, and SHFA ($p = 1.5$) is slower than SHFA ($p = 1$). Moreover, the warm start strategy can be used to further accelerate our SHFA, which will be studied in the future.

5.5 Summary

We have proposed a new method called Semi-supervised Heterogeneous Feature Augmentation (SHFA) for heterogeneous domain adaptation. We build upon the recent work on heterogeneous feature augmentation, in which we also augment the heterogeneous features from the source and target domains by using two newly proposed feature mapping functions, respectively. Then we convert the learning problem into an MKL formulation which is convex and thus the global solution can be guaranteed. Moreover, to utilize the abundant unlabeled data in the target domain, we further extend our HFA method

to semi-supervised HFA (SHFA). Promising results have demonstrated the effectiveness of HFA and SHFA on three real-world datasets for object recognition, text classification and sentiment classification.

Chapter 6

Conclusions and Future Work

With the development of personal imaging devices and network technologies, we have witnessed an explosive growth of web images and videos on Internet. It can be foreseen that the number of web images and videos will continually grow in the future. The learning techniques from web data will attract more and more attentions from both academics and industries. In this thesis, we have investigated three major issues, label noise, multi-view representations, and domain adaptation, for learning from web data, and we have also developed new algorithms for each of those problems. In this chapter, we first conclude the contributions of our works, and then point out several potential directions of learning from web data in the future.

6.1 Conclusions

The contributions of this thesis are summarized as follows,

- We have proposed a new MIL algorithm called MIL-CPB to effectively exploit loosely labeled web images. In our MIL-CPB method, we first generalize the traditional MIL constraint on positive bags as that each positive bag contains at least a portion of true positive instances, considering the relevant images usually contain many true positive images. Based on the new MIL constraints, we present a tight relaxation of MIL, which share a similar form with MKL. We also propose a progressive scheme called PMIL-CPB to automatically and progressively select more

confident positive bags, which leads to more robust classifiers. We conduct comprehensive experiments on the challenging NUS-WIDE data set and Google data set, and the results clearly demonstrate the effectiveness of our proposed approach.

- We have proposed a new problem called multi-view weakly labeled learning for effectively utilizing various types of multi-view weakly labeled web data, and develop a novel co-labeling approach to solve the multi-view weakly labeled learning problem. Our co-labeling algorithm can be applied to many multi-view learning problems including multi-view semi-supervised learning and multi-view multi-instance learning. In our co-labeling, we model the learning problem on each view as a weakly labeled learning problem, which can be formulated as an MKL problem with each kernel being constructed from a pseudo-label vector. Then we use the classifiers trained on different views to help each other by using the predictions on training samples from one classifier to construct the pseudo-label vector set of another classifier. We have also presented a theoretical analysis for our co-labeling approach. Extensive experimental results for MIL and SSL on real-world multi-view datasets demonstrate that our proposed approach achieves state-of-the-art results.
- We have proposed the first work on semi-supervised heterogeneous domain adaptation problem called Semi-supervised Heterogeneous Feature Augmentation (SHFA). Specifically, we first build upon the recent HFA method by reformulating it as an MKL problem, which is convex and thus the global solution can be guaranteed. When there are additional unlabeled training samples in the target domain, we propose to infer the labels of those unlabeled data when learning the SVM classifier based on HFA, which can also be relaxed to an MKL problem. The experimental results have demonstrated the effectiveness of HFA and SHFA on three real-world datasets for object recognition, text classification and sentiment classification.

6.2 Future Work

We discuss the potential directions of learning from web data in the future as follows,

Large Scale Problem: Besides those three issues investigated in this thesis, another major problem when exploiting the web data is the large scale problem. As aforementioned, the number of web images and videos on Internet is spectacularly huge. On one hand, this allows us to easily crawl abundant training data from Internet. On the other hand, it also requires new machine learning techniques for dealing with the large scale training data. The training process of weakly labeled learning algorithms is usually more time-consuming than traditional supervised learning algorithms, as one needs to additionally deal with the label noise. Therefore, to speed up the training of weakly labeled learning algorithms is an important issue for learning from web data. While we have made some process by extending our MIL-CPB by dividing the training data into small batches, and using a quasi-online algorithm to accelerate the training process [37], we believe it needs more efforts in the future to make the weakly labeled learning algorithms more practical for large scale web data.

Complex Data Structure: We have discussed the problem of learning problem with multi-view weakly labeled web data in this thesis. However, Internet develops very fast, and the web data is also associated with various types of structures. For example, the social media becomes extremely popular in recent years. As a result, tremendous information of people’s daily lives is published via social media, which is valuable for building intelligent visual systems. As the information in social media usually transits and connects with the other via social networks, the social media data is not independent, and is naturally organized as a graph. Therefore, how to utilize such graph structure and other more complex structures associated web data for reducing the label noise is an interesting research topic in the future.

Other Issues in Domain Adaptation: While domain adaptation has attracted a lot of attentions from computer vision community, there are still many issues that have not been well exploited for the real-world visual recognition applications. For example, the recent works [164–166] investigated the issue of latent domains in visual domain adaptation problem, in which they discovered that one source domain might contain multiple latent domains. Considering the data crawled from Internet usually comes from many sources, thus the web data is very diverse and may also contain many latent domains, which is worth investigating in the future. Also, in our recent works [105, 167]

we studied the problem when the source domain have additional features. Moreover, the work in [168] studied the domain adaptation problem with evolving target domain, in which the distribution of target domain changes over time. The similar problem will occur if we apply our visual recognition system to a serious of photos captured by the customer, in which the backgrounds, illuminations, and viewpoints may gradually change as the customer travels through different places and times. Therefore, it is also an interesting research issue in the future for learning from web data.

Integrated System: As discussed above, web data is usually very diverse and complex, and learning from web data also contains many difficulties and research issues. It is important to jointly consider all those issues for effectively exploiting the web data. In our recent work [105], we have made some progress by considering to jointly handle the label noise, the domain adaptation problem, and the additional textual features in a single framework, which has demonstrated its advantages compared with the previous works on each issue. Therefore, it is also important to build an integrated scheme to exploit more issues in learning from web data in the future.

References

- [1] V. Sindhwani and P. Niyogi, “A co-regularized approach to semi-supervised learning with multiple views,” in *Proceedings of the ICML Workshop on Learning with Multiple Views*, 2005.
- [2] S. A. Nene, S. K. Nayar, and H. Murase, “Columbia object image library (coil-20),” Tech. Rep. CUCS-005-96, Columbia University, 1996.
- [3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [4] L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,” *T-PAMI*, vol. 28, no. 4, pp. 594–611, 2006.
- [5] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” Tech. Rep. CNS-TR-2007-001, California Institute of Technology, 2007.
- [6] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *CVPR*, 2006.
- [7] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “Sun database: Large-scale scene recognition from abbey to zoo,” in *CVPR*, 2010.
- [8] F. Schroff, A. Criminisi, and A. Zisserman, “Harvesting image databases from the web,” *T-PAMI*, vol. 33, no. 4, pp. 754–766, 2011.
- [9] A. Torralba, R. Fergus, and W. T. Freeman, “80 million tiny images: A large data set for nonparametric object and scene recognition,” *T-PAMI*, vol. 30, no. 11, pp. 1958–1970, 2008.

- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *CVPR*, 2009.
- [11] S. Andrews, I. Tsochantaridis, and T. Hofmann, “Support vector machines for multiple-instance learning,” in *NIPS*, 2002.
- [12] T. Joachims, “Transductive inference for text classification using support vector machines,” in *ICML*, 1999.
- [13] Y.-F. Li, J. T. Kwok, and Z.-H. Zhou, “Semi-supervised learning using label mean,” in *ICML*, 2009.
- [14] Y.-F. Li, I. W. Tsang, J. T. Kwok, and Z.-H. Zhou, “Convex and scalable weakly labeled svms,” *JMLR*, vol. 14, pp. 1391–1445, 2013.
- [15] A. Blum and T. M. Mitchell, “Combining labeled and unlabeled data with co-training,” in *COLT*, pp. 92–100, 1998.
- [16] K. Nigam and R. Ghani, “Analyzing the effectiveness and applicability of co-training,” in *CIKM*, 2000.
- [17] B. Krishnapuram, D. Williams, Y. Xue, A. Hartemink, L. Carin, and M. Figueiredo, “On semi-supervised classification,” in *NIPS*, 2004.
- [18] V. Sindhwani and D. Rosenberg, “An rkhs for multi-view learning and manifold co-regularization,” in *ICML*, 2008.
- [19] S. Yu, B. Krishnapuram, R. Rosales, H. Steck, and R. B. Rao, “Bayesian co-training,” in *NIPS*, 2007.
- [20] A. Kumar and H. D. III, “A co-training approach for multi-view spectral clustering,” in *ICML*, pp. 393–400, 2011.
- [21] A. Kumar, P. Rai, and H. D. III, “Co-regularized multi-view spectral clustering,” in *NIPS*, pp. 1413–1421, 2011.
- [22] S. Dasgupta, M. L. Littman, and D. McAllester, “PAC generalization bounds for co-training,” in *NIPS*, 2002.

- [23] U. Brefeld and T. Scheffer, “Co-EM support vector learning,” in *ICML*, 2004.
- [24] L. Duan, D. Xu, and I. W. Tsang, “Learning with augmented features for heterogeneous domain adaptation,” in *ICML*, 2012.
- [25] J. C. Platt, “Fast training of support vector machines using sequential minimal optimization,” in *Advances in kernel methods*, pp. 185–208, MIT Press, 1999.
- [26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [27] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, “Solving the multiple instance problem with axis-parallel rectangles,” *AI*, vol. 89, no. 1-2, pp. 31–71, 1997.
- [28] O. Maron and T. Lozano-Pérez, “A framework for multiple-instance learning,” in *NIPS*, 1998.
- [29] J. Wang and J.-D. Zucker, “Solving the multiple-instance problem: A lazy learning approach,” in *ICML*, 2000.
- [30] Q. Zhang and S. A. Goldman, “EM-DD: An improved multiple-instance learning technique,” in *NIPS*, 2001.
- [31] O. Maron and A. L. Ratan, “Multiple-instance learning for natural scene classification,” in *ICML*, 1998.
- [32] Y. Chen and J. Z. Wang, “Image categorization by learning and reasoning with regions,” *JMLR*, vol. 5, pp. 913–939, 2004.
- [33] Y. Chen, J. Bi, and J. Z. Wang, “Miles: Multiple-instance learning via embedded instance selection,” *T-PAMI*, vol. 28, no. 12, pp. 1931–1947, 2006.
- [34] P. Viola, J. C. Platt, and C. Zhang, “Multiple instance boosting for object detection,” in *NIPS*, 2005.
- [35] B. Babenko, M. Yang, and S. Belongie, “Visual tracking with online multiple instance learning,” in *CVPR*, 2009.

- [36] M. Li, J.-T. Kwok, and B.-L. Lu, “Online multiple instance learning with no regret,” in *CVPR*, 2010.
- [37] W. Li, L. Duan, I. W.-H. Tsang, and D. Xu, “Batch mode adaptive multiple instance learning for computer vision tasks,” in *CVPR*, pp. 2368–2375, 2012.
- [38] S. Vijayanarasimhan and K. Grauman, “Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization,” in *CVPR*, 2008.
- [39] R. C. Bunescu and R. J. Mooney, “Multiple instance learning for sparse positive bags,” in *ICML*, 2007.
- [40] T. Leung, Y. Song, and J. Zhang, “Handling label noise in video classification via multiple instance learning,” in *ICCV*, 2011.
- [41] W. Li, L. Duan, D. Xu, and I. W.-H. Tsang, “Text-based image retrieval using progressive multi-instance learning,” in *ICCV*, pp. 2049–2055, 2011.
- [42] Z. Fu, A. Robles-Kelly, and J. Zhou, “MILIS: Multiple instance learning with instance selection,” *T-PAMI*, vol. 33, no. 5, pp. 958–977, 2011.
- [43] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola, “Multi-instance kernels,” in *ICML*, 2002.
- [44] R. C. Bunescu and R. J. Mooney, “Multiple instance learning for sparse positive bags,” in *ICML*, 2007.
- [45] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, “Multi-instance learning by treating instances as non-i.i.d. samples,” in *ICML*, 2009.
- [46] Y.-F. Li, I. W. Tsang, J. T.-Y. Kwok, and Z.-H. Zhou, “Tighter and convex maximum margin clustering,” *AISTATS*, 2009.
- [47] Y.-F. Li, J. T. Kwok, I. W. Tsang, and Z.-H. Zhou, “A convex method for locating regions of interest with multi-instance learning,” in *ECML/PKDD (2)*, 2009.
- [48] Z. Zhou, *Multi-instance learning: A survey*, 2004. Technical report, AI Lab, Department of Computer Science & Technology, Nanjing University, Nanjing, China.

- [49] J. Foulds and E. Frank, “A review of multi-instance assumptions,” *Knowledge Engineering Review*, vol. 25, no. 1, pp. 1–25, 2010.
- [50] J. Amores, “Multiple instance classification: Review, taxonomy and comparative study,” *Artificial Intelligence*, vol. 201, pp. 81–105, 2013.
- [51] X. Zhu, “Semi-supervised learning literature survey,” tech. rep., University of Wisconsin-Madison, 2005.
- [52] S. Melacci and M. Belkin, “Laplacian support vector machines trained in the primal,” *JMLR*, vol. 12, pp. 1149–1184, 2011.
- [53] O. Chapelle and A. Zien, “Semi-supervised classification by low density separation,” in *AISTATS*, 2005.
- [54] R. Collobert, F. H. Sinz, J. Weston, and L. Bottou, “Large scale transductive svms,” *JMLR*, vol. 7, pp. 1687–1712, 2006.
- [55] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples,” *JMLR*, vol. 7, pp. 2399–2434, 2006.
- [56] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, “Maximum margin clustering,” in *NIPS*, 2004.
- [57] K. Zhang, I. W. Tsang, and J. T. Kwok, “Maximum margin clustering made practical,” *IEEE Transactions on Neural Networks*, vol. 20, no. 4, pp. 583–596, 2009.
- [58] A. Joulin and F. Bach, “A convex relaxation for weakly supervised classifiers,” in *ICML*, 2012.
- [59] A. J. Smola, L. Song, and C. H. Teo, “Relative novelty detection,” *JMLR - Proceedings Track*, vol. 5, pp. 536–543, 2009.
- [60] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, “Inlier-based outlier detection via direct density ratio estimation,” in *ICDM*, pp. 223–232, 2008.

- [61] T. Kanamori, S. Hido, and M. Sugiyama, “A least-squares approach to direct importance estimation,” *JMLR*, vol. 10, pp. 1391–1445, 2009.
- [62] C. Elkan and K. Noto, “Learning classifiers from only positive and unlabeled data,” in *KDD*, 2008.
- [63] R. Rahmani and S. A. Goldman, “Missl: Multiple-instance semi-supervised learning,” in *ICML*, pp. 705–712, 2006.
- [64] C. Xu, D. Tao, and C. Xu, “A survey on multi-view learning,” *CoRR*, vol. abs/1304.5634, 2013.
- [65] H. Hotelling, “Relations between two sets of variates,” *Biometrika*, vol. 28, pp. 312–337, 1936.
- [66] D. R. Hardoon, S. R. Szedmak, and J. R. Shawe-taylor, “Canonical correlation analysis: An overview with application to learning methods,” *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2004.
- [67] S. Bickel and T. Scheffer, “Multi-view clustering,” in *ICDM*, 2004.
- [68] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, “Multi-view clustering via canonical correlation analysis,” in *ICML*, 2009.
- [69] J. D. R. Farquhar, H. Meng, S. Szedmak, D. R. Hardoon, and J. Shawe-taylor, “Two view learning:SVM-2K, theory and practice,” in *NIPS*, 2006.
- [70] G. R. G. Lanckriet, N. Cristianini, P. L. Bartlett, L. E. Ghaoui, and M. I. Jordan, “Learning the kernel matrix with semidefinite programming,” *JMLR*, vol. 5, pp. 27–72, 2004.
- [71] A. Rakotomamonjy, F. R. Bach, and Y. Grandvalet, “SimpleMKL,” *JMLR*, vol. 9, pp. 2491–2521, 2008.
- [72] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, “ ℓ_p -norm multiple kernel learning,” *JMLR*, vol. 12, pp. 953–997, 2011.

- [73] X. Xu, I. W. Tsang, and D. Xu, “Soft margin multiple kernel learning,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 24, no. 5, pp. 749–761, 2013.
- [74] Z.-H. Zhou and M. Li, “Tri-training: Exploiting unlabeled data using three classifiers,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.
- [75] M. Li and Z.-H. Zhou, “Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples,” *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 37, pp. 1088–1098, 2007.
- [76] W. Wang and Z.-H. Zhou, “Analyzing co-training style algorithms,” in *ECML*, pp. 454–465, 2007.
- [77] S. P. Abney, “Bootstrapping,” in *ACL*, pp. 360–367, 2002.
- [78] M.-F. Balcan, A. Blum, and K. Yang, “Co-training and expansion: Towards bridging theory and practice,” in *NIPS*, 2004.
- [79] W. Wang and Z.-H. Zhou, “A new analysis of co-training,” in *ICML*, 2010.
- [80] W. Wang and Z.-H. Zhou, “Co-training with insufficient views,” in *ACML*, pp. 467–482, 2013.
- [81] W. Li, L. Duan, I. W.-H. Tsang, and D. Xu, “Co-labeling: A new multi-view learning approach for ambiguous problems,” in *ICDM*, pp. 419–428, 2012.
- [82] S. J. Pan and Q. Yang, “A survey on transfer learning,” *TKDE*, vol. 22, pp. 1345–1359, October 2010.
- [83] J. Jiang, “A literature survey on domain adaptation of statistical classifiers,” tech. rep., University of Illinois at Urbana-Champaign, 2008.
- [84] A. Margolis, “A literature review of domain adaptation with unlabeled data,” tech. rep., University of Washington, 2011.
- [85] O. Beijbom, “Domain adaptations for computer vision applications,” *CoRR*, vol. abs/1211.4860, 2012.

- [86] Q. Li, “Literature survey: Domain adaptation algorithms for natural language processing,” 2012.
- [87] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, “Visual domain adaptation: An overview of recent advances,” 2014.
- [88] L. Shao, F. Zhu, and X. Li, “Transfer learning for visual categorization: A survey,” *T-NNLS*, 2014.
- [89] J. Yang, R. Yan, and A. Hauptmann, “Cross-domain video concept detection using adaptive svms,” in *ACM MM*, 2007.
- [90] L. Duan, I. W. Tsang, and D. Xu, “Domain transfer multiple kernel learning,” *T-PAMI*, vol. 34, pp. 465–479, March 2012.
- [91] H. Daumé III, “Frustratingly easy domain adaptation,” in *ACL*, 2007.
- [92] H. Daumé III, A. Kumar, and A. Saha, “Co-regularization based semi-supervised domain adaptation,” in *NIPS*, 2010.
- [93] B. S. David Lopez-paz, Jose M. Hernández-lobato, “Semi-supervised domain adaptation with non-parametric copulas,” in *NIPS*, 2012.
- [94] M. Xiao and Y. Guo, “Semi-supervised kernel matching for domain adaptation,” 2012.
- [95] T. D. Jeff Donahue, Judy Hoffman, Erik Rodner, Kate Saenko, “Semi-supervised domain adaptation with instance constraints,” in *CVPR*, 2013.
- [96] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Scholkopf, “Correcting sample selection bias by unlabeled data,” in *NIPS*, 2007.
- [97] A. Gretton, K. M. Gorgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A kernel two-sample test,” *JMLR*, vol. 23, pp. 723–773, 2012.
- [98] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” *T-NNLS*, vol. 22, pp. 199–210, Feb 2011.

- [99] R. Gopalan, R. Li, and R. Chellappa, “Domain adaptation for object recognition: An unsupervised approach,” in *ICCV*, 2011.
- [100] B. Gong, Y. Shi, F. Sha, and K. Grauman, “Geodesic flow kernel for unsupervised domain adaptation,” in *CVPR*, 2012.
- [101] M. Baktashmotlagh, M. Harandi, and M. S. Brian Lovell, “Unsupervised domain adaptation by domain invariant projection,” in *ICCV*, 2013.
- [102] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, “Unsupervised visual domain adaptation using subspace alignment,” in *ICCV*, 2013.
- [103] L. Bruzzone and M. Marconcini, “Domain adaptation problems: A DASVM classification technique and a circular validation strategy,” *T-PAMI*, vol. 32, no. 5, pp. 770–787, 2010.
- [104] W.-S. Chu, F. DelaTorre, and J. Cohn, “Selective transfer machine for personalized facial action unit detection,” in *CVPR*, 2013.
- [105] W. Li, L. Niu, and D. Xu, “Exploiting privileged information from web data for image categorization,” in *ECCV*, 2014.
- [106] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua, “Domain adaptation from multiple sources via auxiliary classifiers,” in *International Conference on Machine Learning*, 2009.
- [107] L. Duan, D. Xu, and I. W. Tsang, “Domain adaptation from multiple sources: A domain-dependent regularization approach,” *T-NNLS*, vol. 23, pp. 504–518, March 2012.
- [108] L. Chen, L. Duan, and D. Xu, “Event recognition in videos by learning from heterogeneous web sources,” in *CVPR*, pp. 2666–2673, 2013.
- [109] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, “Analysis of representations for domain adaptation,” in *NIPS*, 2006.

- [110] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, “Learning bounds for domain adaptation,” in *NIPS*, 2007.
- [111] Y. Mansour, M. Mohri, and A. Rostamizadeh, “Domain adaptation with multiple sources,” in *NIPS*, 2008.
- [112] K. Crammer, M. Kearns, and J. Wortman, “Learning from multiple sources,” *JMLR*, vol. 9, pp. 1757–1774, 2008.
- [113] Y. Mansour, M. Mohri, and A. Rostamizadeh, “Domain adaptation: Learning bounds and algorithms,” in *COLT*, 2009.
- [114] C. Cortes, Y. Mansour, and M. Mohri, “Learning bounds for importance weighting,” in *NIPS*, 2010.
- [115] S. Ben-David, T. Lu, T. Luu, and D. Pal, “Impossibility theorems for domain adaptation,” in *AISTATS*, 2010.
- [116] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” *ML*, vol. 79, pp. 151–175, 2010.
- [117] C. Zhang, L. Zhang, and J. Ye, “Generalization bounds for domain adaptation,” in *NIPS*, 2012.
- [118] S. Ben-David, S. Shalev-Shwartz, and R. Urner, “Domain adaptation can quantity compensate for quality?,” in *ISAIM*, 2012.
- [119] S. Ben-David and R. Urner, “On the hardness of domain adaptation and the utility of unlabeled target samples,” in *ALT*, 2012.
- [120] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, “Translated learning: Transfer learning across different feature spaces,” in *NIPS*, 2009.
- [121] P. Prettenhofer and B. Stein, “Cross-Language Text Classification using Structural Correspondence Learning,” in *ACL*, 2010.
- [122] B. Wei and C. Pal, “Cross-lingual adaptation: An experiment on sentiment classifications,” in *ACL*, 2010.

- [123] Q. Yang, Y. Chen, G.-R. Xue, W. Dai, and Y. Yu, “Heterogeneous transfer learning for image clustering via the social web,” in *ACL/IJCNLP*, 2009.
- [124] Y. Zhu, Y. Chen, Z. Lu, S. J. Pan, G.-R. Xue, Y. Yu, and Q. Yang, “Heterogeneous transfer learning for image classification,” in *AAAI*, 2011.
- [125] X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu, “Transfer learning on heterogenous feature spaces via spectral transformation,” in *ICDM*, 2010.
- [126] C. Wang and S. Mahadevan, “Heterogeneous domain adaptation using manifold alignment,” in *IJCAI*, 2011.
- [127] B. Kulis, K. Saenko, and T. Darrell, “What you saw is not what you get: Domain adaptation using asymmetric kernel transforms,” in *CVPR*, 2011.
- [128] W. Li, L. Duan, D. Xu, and I. W. Tsang, “Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation,” vol. 36, pp. 1134–1148, June 2014.
- [129] J. Hoffman, E. Rodner, J. Donahue, K. Saenko, and T. Darrell, “Efficient learning of domain-invariant image representations,” in *International Conference on Learning Representations*, 2013.
- [130] R. Datta, D. Joshi, J. Li, and J. Z. Wang, “Image retrieval: Ideas, influences, and trends of the new age,” *ACM Computer Surveys*, vol. 40, no. 2, pp. 1–60, 2008.
- [131] L. Chen, D. Xu, I. W. Tsang, and J. Luo, “Tag-based web photo retrieval improved by batch mode re-tagging,” in *CVPR*, 2010.
- [132] X.-J. Wang, L. Zhang, X. Li, and W.-Y. Ma, “Annotating images by mining image search results,” *T-PAMI*, vol. 30, no. 11, pp. 1919–1932, 2008.
- [133] L. Duan, W. Li, I. W. Tsang, and D. Xu, “Improving web image search by bag-based re-ranking,” *T-IP*, 2011.
- [134] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, “Learning object categories from google’s image search,” in *ICCV*, 2005.

- [135] M. Fritz and B. Schiele, “Decomposition, discovery and detection of visual categories using topic models,” in *CVPR*, 2008.
- [136] X. Li, C. G. M. Snoek, and M. Worring, “Learning social tag relevance by neighbor voting,” *T-MM*, vol. 11, no. 7, pp. 1310–1322, 2009.
- [137] P. V. Gehler and O. Chapelle, “Deterministic annealing for multiple-instance learning,” in *AISTATS*, 2007.
- [138] Y. Liu, D. Xu, I. W. Tsang, and J. Luo, “Textual query of personal photos facilitated by large-scale web data,” *T-PAMI*, vol. 33, no. 5, pp. 1022–1036, 2011.
- [139] F. Li and C. Sminchisescu, “Convex multiple-instance learning by estimating likelihood ratio,” in *NIPS*, 2010.
- [140] J. E. Kelley, “The cutting plane method for solving convex programs,” *SIAM Journal on Applied Mathematics*, vol. 8, no. 4, pp. 703–712, 1960.
- [141] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, “Nus-wide: a real-world web image database from national university of singapore,” in *CIVR*, 2009.
- [142] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, vol. 60, pp. 91–110, November 2004.
- [143] C.-W. Seah, I. W.-H. Tsang, and Y.-S. Ong, “Healing sample selection bias by source classifier selection,” in *ICDM*, 2011.
- [144] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
- [145] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in *CVPR*, 2010.
- [146] W. Li, L. Duan, D. Xu, and I. W. Tsang, “Text-based image retrieval using progressive multi-instance learning,” in *ICCV*, 2011.
- [147] D. Greene and P. Cunningham, “Practical solutions to the problem of diagonal dominance in kernel document clustering,” in *ICML*, pp. 377–384, 2006.

- [148] G. Li, S. C. H. Hoi, and K. Chang, “Two-view transductive support vector machines,” in *SDM*, pp. 235–244, 2010.
- [149] M. Chen, K. Q. Weinberger, and Y. Chen, “Automatic feature decomposition for single view co-training,” in *ICML*, 2011.
- [150] J. Blitzer, R. McDonald, and F. Pereira, “Domain adaptation with structural correspondence learning,” in *EMNLP*, 2006.
- [151] J. Blitzer, M. Dredze, and F. Pereira, “Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification,” in *ACL*, 2007.
- [152] L. Duan, D. Xu, I. W. Tsang, and J. Luo, “Visual event recognition in videos by learning from web data,” *T-PAMI*, vol. 34, pp. 1667–1680, September 2012.
- [153] L. Duan, D. Xu, and S.-F. Chang, “Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach,” in *CVPR*, pp. 1338–1345, 2012.
- [154] M. Harel and S. Mannor, “Learning from multiple outlooks,” in *ICML*, 2011.
- [155] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in *ECCV*, 2010.
- [156] M. Dudik, Z. Harchaoui, and J. Malick, “Lifted coordinate descent for learning with trace-norm regularization,” in *AISTATS*, 2012.
- [157] P. V. Gehler and S. Nowozin, “Infinite kernel learning,” tech. rep., Max Planck Institute for Biological Cybernetics, 2008.
- [158] M. Tan, L. Wang, and I. W. Tsang, “Learning sparse svm for feature selection on very high dimensional datasets,” in *ICML*, 2010.
- [159] A. Mutapcic and S. Boyd, “Cutting-set methods for robust convex optimization with pessimizing oracles,” *Optimization Methods & Software*, vol. 24, pp. 381–406, June 2009.

- [160] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola, “Input space versus feature space in kernel-based methods,” *T-NN*, vol. 10, pp. 1000–1017, September 1999.
- [161] H. Bay, T. Tuytelaars, and L. V. Gool, “Surf: Speeded up robust features,” in *ECCV*, 2006.
- [162] M. Amini, N. Usunier, and C. Goutte, “Learning from multiple partially observed views – an application to multilingual text categorization,” in *NIPS*, 2009.
- [163] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “DeCAF: A deep convolutional activation feature for generic visual recognition,” in *ICML*, 2014.
- [164] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko, “Discovering latent domains for multisource domain adaptation,” in *ECCV*, 2012.
- [165] B. Gong, K. Grauman, and F. Sha, “Reshaping visual datasets for domain adaptation,” in *NIPS*, 2013.
- [166] Z. Xu, W. Li, L. Niu, and D. Xu, “Exploiting low-rank structure from latent domains for domain generalization,” in *ECCV*, 2014.
- [167] L. Chen, W. Li, and D. Xu, “Recognizing rgb images by learning from rgb-d data,” in *CVPR*, 2014.
- [168] J. Hoffman, T. Darrell, and K. Saenko, “Continuous manifold based adaptation for evolving visual domains,” in *CVPR*, 2014.

Appendix A

Appendix

A.1 Proof of Theorem 4.1

In this section, we present the proofs for the generalization bounds of the weakly labeled learning method and the co-labeling algorithm, respectively.

A.1.1 Generalization Bound for Weakly Labeled Learning

We start from the generalization bound for supervised learning. In supervised learning, the training samples can be represented as $\{(\mathbf{x}_i, g_i)\}_{i=1}^n$, where $\mathbf{x}_i$ is the i -th training sample, g_i is its label, and n is the total number of training samples. In kernel based algorithms such as SVM, one usually maps the training samples into a nonlinear feature space with a kernel function. Let us denote the nonlinear feature as $\phi(\mathbf{x})$, then the target classifier can be represented as $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i g_i \phi(\mathbf{x}_i)' \phi(\mathbf{x})$. As shown in [144], for the soft margin SVM with the squared hinge loss, the generalization bound can be described as

Proposition 2 *For fixed $B, \rho > 0$ and $0 < \delta < 1$, we define $\mathcal{F} = \{f \mid \|f\|_{\mathcal{H}}^2 \leq B^2\}$, then, with probability at least $1 - \delta$, every function $f \in \mathcal{F}$ satisfies,*

$$\text{err}(f) \leq \frac{1}{\rho^2} \ell(f, \mathbf{g}) + \frac{4}{\rho} \hat{R}(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2n}}, \quad (\text{A.1})$$

where $\text{err}(f)$ is the expectation error of the testing samples drawn from the same distribution of the training samples, $\ell(f, \mathbf{g}) = \frac{1}{n} \sum_{i=1}^n (\max(0, \rho - g_i f(\mathbf{x}_i)))^2$ is the squared hinge loss on training samples, $\hat{R}(\mathcal{F}) = \frac{2B}{n} \sqrt{\text{tr}(\mathbf{K})}$ is the empirical Rademacher complexity of the feasible function set $\mathcal{F}$, and $\text{tr}(\mathbf{K})$ is the trace of the kernel matrix $\mathbf{K}$.

The proof can be found in Chapter 4 and Chapter 7 of [144]. We give a brief explanation on the bound. The first term on the right hand side of the inequality (A.1) is the empirical loss on the training data measured by the squared hinge loss. The second term is based on the empirical Rademacher complexity of a class of functions, which models the capacity of the feasible set $\mathcal{F}$. The third term is a constant term induced in the derivations. In more brief words, the generalization bound of SVM mainly depends on the empirical loss on the training samples and the capacity of the feasible set of target classifiers.

Based on the above generalization bound for supervised learning, we first give the learning bound for the weakly labeled learning on the v -th view in our co-labeling algorithm. Let us rewrite the objective of the weakly labeled learning problem (Eq. (4.7) in the main text) as follows,

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

The final classifier can be represented as $f(\mathbf{x}) = \sum_{m=1}^M d_m f_m(\mathbf{x})$ with $f_m(\mathbf{x}) = \mathbf{w}_m' \phi(\mathbf{x})$ being the m -th base classifier.

Let us denote the empirical loss for the weakly labeled learning problem in (A.2) as $\hat{\ell}(f) = \frac{1}{n} \sum_{i=1}^n \xi_i^2$ with ξ_i defined in (A.2), i.e., $\xi_i = \max(0, \rho - \sum_{m=1}^M d_m y_i^m \mathbf{w}_m' \phi(\mathbf{x}_i))$. We also denote the squared hinge loss calculated by using the ground truth labeling as $\ell(f, \mathbf{g}) = \frac{1}{n} \sum_{i=1}^n \zeta_i^2$ with $\zeta_i = \max(0, \rho - g_i f(\mathbf{x}_i))$. We assume that both the decision value of each base classifier and the squared hinge loss can be bounded as $|f_m(\mathbf{x}_i)|^2 \leq U$ and $\xi_i^2 \leq U$ for some fixed U . Then, we have the following proposition,

Proposition 3 *The squared hinge loss based on the ground truth labeling $\mathbf{g}$ can be bounded as,*

$$\ell(f, \mathbf{g}) \leq \hat{\ell}(f) + \frac{8U \|\mathbf{d}\|_1}{n} \sum_{m=1}^M d_m h(\mathbf{g}, \mathbf{y}_m), \quad (\text{A.3})$$

where $h(\mathbf{g}, \mathbf{y}_m)$ is the hamming distance between $\mathbf{g}$ and $\mathbf{y}_m$, i.e., $h(\mathbf{g}, \mathbf{y}_m) = \sum_{i=1}^n I(g_i \neq y_i^m)$ with $I(\cdot)$ being the indicator function.

Proof: We first bound the hinge loss on each sample as follows,

$$\begin{aligned}
 \zeta_i &= \max(0, \rho - g_i f(\mathbf{x}_i)) \\
 &= \max\left(0, \rho - g_i \sum_{m=1}^M d_m f_m(\mathbf{x}_i)\right) \\
 &= \max\left(0, \rho - \sum_{m=1}^M d_m (g_i - y_i^m + y_i^m) f_m(\mathbf{x}_i)\right) \\
 &= \max\left(0, \rho - \sum_{m=1}^M d_m y_i^m f_m(\mathbf{x}_i) + \sum_{m=1}^M d_m (y_i^m - g_i) f_m(\mathbf{x}_i)\right) \\
 &\leq \max\left(0, \rho - \sum_{m=1}^M d_m y_i^m f_m(\mathbf{x}_i)\right) \\
 &\quad + \max\left(0, \sum_{m=1}^M d_m (y_i^m - g_i) f_m(\mathbf{x}_i)\right) \\
 &\leq \xi_i + \left| \sum_{m=1}^M d_m (y_i^m - g_i) f_m(\mathbf{x}_i) \right| \\
 &\leq \xi_i + 2\sqrt{U} \sum_{m=1}^M d_m I(g_i \neq y_i^m),
 \end{aligned}$$

where the last inequality is from $|y_i^m - g_i| \leq 2I(g_i \neq y_i^m)$ and $|f_m(\mathbf{x}_i)| \leq \sqrt{U}$.

With the above inequality, we obtain

$$\begin{aligned}
 \zeta_i^2 &\leq \left(\xi_i + 2\sqrt{U} \sum_{m=1}^M d_m I(g_i \neq y_i^m) \right)^2 \\
 &= \xi_i^2 + 4\xi_i \sqrt{U} \sum_{m=1}^M d_m I(g_i \neq y_i^m) \\
 &\quad + 4U \left(\sum_{m=1}^M d_m I(g_i \neq y_i^m) \right)^2 \\
 &\leq \xi_i^2 + 4U \sum_{m=1}^M d_m I(g_i \neq y_i^m) \\
 &\quad + 4U \left(\sum_{m=1}^M d_m I(g_i \neq y_i^m) \right)^2 \\
 &\leq \xi_i^2 + 8U \|\mathbf{d}\|_1 \sum_{m=1}^M d_m I(g_i \neq y_i^m),
 \end{aligned}$$

where the second inequality is from $|\xi_i| \leq \sqrt{U}$, and the third inequality is obtained by multiplying $\|\mathbf{d}\|_1$ to the second term (note we have $\|\mathbf{d}\|_1 \geq 1$ in this work¹), and using the inequality $(\sum_{m=1}^M d_m I(g_i \neq y_i^m))^2 = \left(\sum_{m=1}^M d_m I(g_i \neq y_i^m)\right) \left(\sum_{m=1}^M d_m I(g_i \neq y_i^m)\right) \leq (\sum_{m=1}^M d_m)(\sum_{m=1}^M d_m I(g_i \neq y_i^m)) = \|\mathbf{d}\|_1 \sum_{m=1}^M d_m I(g_i \neq y_i^m)$. Then, based on the definition of $\ell(f, \mathbf{g})$,

$$\begin{aligned} \ell(f, \mathbf{g}) &= \frac{1}{n} \sum_{i=1}^n \xi_i^2 \\ &\leq \frac{1}{n} \sum_{i=1}^n \left(\xi_i^2 + 8U \|\mathbf{d}\|_1 \sum_{m=1}^M d_m I(g_i \neq y_i^m) \right) \\ &= \hat{\ell}(f) + \frac{8U \|\mathbf{d}\|_1}{n} \sum_{m=1}^M d_m h(\mathbf{g}, \mathbf{y}_m). \end{aligned}$$

This completes the proof.

Now, we prove the generalization bound of our weakly labeled learning method as follows:

Theorem A.4 *For fixed $B, U \geq 0$ and $0 < \delta < 1$, we define $\mathcal{F} = \{f \mid \|f\|_{\mathcal{H}}^2 \leq B^2\}$, then, with probability at least $1 - \delta$, every function $f \in \mathcal{F}$ satisfies*

$$\begin{aligned} \text{err}(f) &\leq \frac{1}{\rho^2} \hat{\ell}(f) + \frac{8U \|\mathbf{d}\|_1}{n\rho^2} \sum_{m=1}^M d_m h(\mathbf{g}, \mathbf{y}_m) \\ &\quad + \frac{8B}{n\rho} \sqrt{\text{tr}(\mathbf{K})} + 3\sqrt{\frac{\ln(2/\delta)}{2n}}, \end{aligned} \tag{A.4}$$

where $\text{err}(f)$ is the expectation error on the testing samples drawn from the same distribution of the training samples, $\hat{\ell}(f) = \frac{1}{n} \sum_{i=1}^n \xi_i^2$ is the empirical squared hinge loss on the training samples with ξ_i defined as in (4.4) of the main text, and $\text{tr}(\mathbf{K})$ is the trace of $\mathbf{K}$.

Proof: The proof can be easily obtained by substituting the inequality (A.3) into the generalization bound in (A.1).

¹At the optimum, the constraint $\|\mathbf{d}\|_p \leq 1$ of ℓ_p -MKL with $p \geq 1$ can always be satisfied at the upper bound, i.e., $\|\mathbf{d}\|_p = 1$ (see [72]). Then we have $\sum_m d_m \geq \sum_m (d_m)^p = 1$, where the inequality holds when $0 \leq d_m \leq 1$ and $p \geq 1$. It can be verified that it also holds for our multi-layer MKL with a different norm on each layer, considering in our work $p_1 \geq 1$, $p_2 \geq 1$, and $p_3 \geq 1$.

When $M = 1$ and $\mathbf{y}_m = \mathbf{g}$, we have $d_m = 1$, $h(\mathbf{y}_m, \mathbf{g}) = 0$ and $\zeta_i = \xi_i$, then the bound in the above theorem is reduced to the generalization bound of SVM as in [144] (see Lemma 2). Now we discuss the general case when $M > 1$. As the last two terms on the right hand side of (A.4) are similar to those in the generalization bound of SVM, we discuss the first two terms only.

The first term is the empirical loss of the MKL problem in (A.2), which should not be large as we minimize it in the objective function of (A.2). The second term is the linear combination of the hamming distance between the ground truth labeling $\mathbf{g}$ and the m -th pseudo-label vector $\mathbf{y}_m$, where the combination coefficients are d_m for $m = 1, \dots, M$. Recall that in the MKL problem, we aim to find an optimal combinations of $\sum_m d_m(\mathbf{y}_m \mathbf{y}_m')$ to approximate the output labeling matrix $\mathbf{g} \mathbf{g}'$ such that the regularized empirical risk is minimized.

On one hand, if the decision boundary of $\mathbf{y}_m$ can separate the two opposite clusters well, which means the corresponding decision boundary lies in low density regions. So it leads to a low regularized empirical risk, and should be assigned a large d_m in MKL. Moreover, based on the cluster assumption [12], the decision boundary of the ground truth labeling $\mathbf{g}$ lies in low density regions. In this case, $\mathbf{y}_m$ should be similar to $\mathbf{g}$, which means $h(\mathbf{g}, \mathbf{y}_m)$ should be reasonably small and $d_m h(\mathbf{g}, \mathbf{y}_m)$ will not be large. On the other hand, if the decision boundary of $\mathbf{y}_m$ does not lie in low density regions, then d_m should be small, which means $d_m h(\mathbf{g}, \mathbf{y}_m)$ will not be large as well. Thus, $\sum_{m=1}^M d_m h(\mathbf{g}, \mathbf{y}_m)$ should be small. In our co-labeling approach, by varying different biases to generate different pseudo-label vectors, we expect there should be some $\mathbf{y}_m$'s that lie in low density regions. Therefore, the generalization bound in Theorem A.4 is reasonably tight.

A.1.2 Generalization Bound for Co-Labeling

The final classifier in our co-labeling algorithm is obtained by fusing the MKL classifiers from all the V views. To distinguishing the classifiers from different views, we add the superscript v for the variables on the v^{th} view. Then, the final classifier can be represented as $\tilde{f}(\mathbf{x}) = \frac{1}{V} \sum_{v=1}^V \frac{1}{\rho^v} f^v(\mathbf{x}^v)$, where f^v is the classifier on the v -th view for $v = 1, \dots, V$. Let us denote $\zeta_i = \max(0, 1 - g_i \tilde{f}(\mathbf{x}_i))$ as the hinge loss of the i -th sample by using $\tilde{f}$,

and $\zeta_i^v = \max(0, \rho^v - g_i f^v(\mathbf{x}_i^v))$ as the hinge loss of the i -th sample by using f^v , then we have

$$\begin{aligned}
 \zeta_i &= \max\left(0, 1 - g_i \tilde{f}(\mathbf{x})\right) \\
 &= \max\left(0, 1 - g_i \frac{1}{V} \sum_{v=1}^V \frac{1}{\rho^v} f^v(\mathbf{x}_i^v)\right) \\
 &= \max\left(0, \sum_{v=1}^V \frac{1}{V \rho^v} (\rho^v - g_i f^v(\mathbf{x}_i^v))\right) \\
 &\leq \sum_{v=1}^V \frac{1}{V \rho^v} \max(0, \rho^v - g_i f^v(\mathbf{x}_i^v)) \\
 &= \sum_{v=1}^V \frac{1}{V \rho^v} \zeta_i^v.
 \end{aligned}$$

By using Cauchy–Schwarz inequality, we have $\left(\sum_{v=1}^V \frac{1}{V \rho^v} \zeta_i^v\right)^2 \leq \left(\sum_{v=1}^V \frac{1}{V^2}\right) \left(\sum_{v=1}^V \left(\frac{\zeta_i^v}{\rho^v}\right)^2\right) = \frac{1}{V} \sum_{v=1}^V \frac{1}{(\rho^v)^2} (\zeta_i^v)^2$. Then, we bound the loss function as follows:

$$\begin{aligned}
 \ell(\tilde{f}, \mathbf{g}) &= \frac{1}{n} \sum_{i=1}^n \zeta_i^2 \\
 &\leq \frac{1}{n} \sum_{i=1}^n \frac{1}{V} \sum_{v=1}^V \frac{1}{(\rho^v)^2} (\zeta_i^v)^2 \\
 &= \frac{1}{V} \sum_{v=1}^V \frac{1}{(\rho^v)^2} \ell(f^v, \mathbf{g}) \\
 &\leq \frac{1}{V} \sum_{v=1}^V \frac{1}{(\rho^v)^2} \left(\hat{\ell}(f^v) + \frac{8U \|\mathbf{d}^v\|_1}{n} \sum_{m=1}^M d_m^v h(\mathbf{g}, \mathbf{y}_m^v) \right).
 \end{aligned}$$

Moreover, from the property of the empirical Rademacher complexity, we have $\hat{R}(\mathcal{F}) = \hat{R}(\frac{1}{V} \sum_{v=1}^V \frac{1}{\rho^v} \mathcal{F}^v) \leq \frac{1}{V} \sum_{v=1}^V \frac{1}{\rho^v} \hat{R}(\mathcal{F}^v)$ (see Theorem 4.15 in Chapter 4 of [144], page 101). Substituting $\ell(\tilde{f}, \mathbf{g})$ and $\hat{R}(\mathcal{F})$ into the inequality in (A.1), we thus obtain the generalization bound of $\tilde{f}(\mathbf{x})$ as follows

$$\begin{aligned}
 \text{err}(\tilde{f}) &\leq \frac{1}{V} \sum_{v=1}^V \frac{1}{(\rho^v)^2} \left(\hat{\ell}(f^v) + \frac{8U \|\mathbf{d}^v\|_1}{n} \sum_{m=1}^M d_m^v h(\mathbf{g}, \mathbf{y}_m^v) \right) \\
 &\quad + \frac{8B}{nV} \sum_{v=1}^V \frac{1}{\rho^v} \sqrt{\text{tr}(\mathbf{K}^v)} + 3 \sqrt{\frac{\ln(2/\delta)}{2n}},
 \end{aligned} \tag{A.5}$$

where the ρ in (A.1) is set to 1, because the loss function here is $\zeta_i = \max(0, 1 - g_i \tilde{f}(\mathbf{x}_i))$. Thus, we complete the proof.

A.2 Proof of Theorem 5.3

In the main text, we have formulated our HFA as an infinite kernel learning problem as follows (*i.e.*, the Eq. (5.10) in the main text),

$$\min_{\boldsymbol{\theta} \in \mathcal{D}_{\boldsymbol{\theta}}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})' \sum_{r=1}^{\infty} \theta_r \mathbf{K}_r(\boldsymbol{\alpha} \circ \mathbf{y}), \quad (\text{A.6})$$

where $\mathbf{K}_r = \mathbf{S}(\lambda \mathbf{M}_r + \mathbf{I})\mathbf{S}$, $\mathcal{A} = \{\boldsymbol{\alpha} | \mathbf{y}'\boldsymbol{\alpha} = 0, \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}\}$ and $\mathcal{D}_{\boldsymbol{\theta}} = \{\boldsymbol{\theta} | \mathbf{1}'\boldsymbol{\theta} \leq 1, \boldsymbol{\theta} \geq \mathbf{0}\}$. As shown in the main text, the dual form of (A.6) can be written as follows by introducing a dual variable τ for $\boldsymbol{\theta}$:

$$\begin{aligned} \max_{\tau, \boldsymbol{\alpha} \in \mathcal{A}} \quad & \mathbf{1}'\boldsymbol{\alpha} - \tau, \\ \text{s.t.} \quad & \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})' \mathbf{K}_r(\boldsymbol{\alpha} \circ \mathbf{y}) \leq \tau, \quad \forall r. \end{aligned} \quad (\text{A.7})$$

Actually, the problem in (A.6) can also be deemed as the dual form of (A.7) by considering θ_r as the dual variable for the r -th constraint in (A.7).

In the main text, we have defined $F(\boldsymbol{\alpha}, \boldsymbol{\theta}) = \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})' \sum_{r=1}^{\infty} \theta_r \mathbf{K}_r(\boldsymbol{\alpha} \circ \mathbf{y})$, and denoted the optimal solution to (A.6) as $(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) = \arg \min_{\boldsymbol{\theta} \in \mathcal{D}_{\boldsymbol{\theta}}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} F(\boldsymbol{\alpha}, \boldsymbol{\theta})$. Let us denote $G(\boldsymbol{\alpha}, \tau) = \mathbf{1}'\boldsymbol{\alpha} - \tau$, and the optimal solution to (A.7) as $(\boldsymbol{\alpha}^*, \tau^*)$. We first give the following lemma which will be used in the proof of Theorem 2:

Lemma A.1 *For the infinite kernel learning problem in (A.6) and (A.7), we have $F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) = G(\boldsymbol{\alpha}^*, \tau^*)$, and $\frac{1}{2}(\boldsymbol{\alpha}^* \circ \mathbf{y})' \sum_{r=1}^{\infty} \theta_r^* \mathbf{K}_r(\boldsymbol{\alpha}^* \circ \mathbf{y}) = \tau^*$.*

Proof: The lemma can be proved by using the KKT condition. For any $\theta_r^* > 0$, we have $\frac{1}{2}(\boldsymbol{\alpha}^* \circ \mathbf{y})' \mathbf{K}_r(\boldsymbol{\alpha}^* \circ \mathbf{y}) = \tau^*$. Since $\sum_{r=1}^{\infty} \theta_r^* = 1$, then we can obtain $\frac{1}{2}(\boldsymbol{\alpha}^* \circ \mathbf{y})' \sum_{r=1}^{\infty} \theta_r^* \mathbf{K}_r(\boldsymbol{\alpha}^* \circ \mathbf{y}) = \tau^*$. Therefore we have

$$\begin{aligned} F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) &= \mathbf{1}'\boldsymbol{\alpha}^* - \frac{1}{2}(\boldsymbol{\alpha}^* \circ \mathbf{y})' \sum_{r=1}^{\infty} \theta_r^* \mathbf{K}_r(\boldsymbol{\alpha}^* \circ \mathbf{y}) \\ &= \mathbf{1}'\boldsymbol{\alpha}^* - \tau^* \\ &= G(\boldsymbol{\alpha}^*, \tau^*). \end{aligned}$$

This completes the proof.

It can be observed that Lemma A.1 is also satisfied for the multiple kernel learning problem with a finite number of kernels.

Recall that we have denoted the optimal solution of the MKL problem at the r -th iteration as $(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r)$. Because there are at most r non-zero elements in $\boldsymbol{\theta}^r$, we assume these non-zero elements are the first r entries in $\boldsymbol{\theta}^r$ for ease of presentation. Then we can represent $(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) = \arg \min_{\boldsymbol{\theta} \in \mathcal{E}_r} \max_{\boldsymbol{\alpha} \in \mathcal{A}} F(\boldsymbol{\alpha}, \boldsymbol{\theta})$, where $\mathcal{E}_r = \{\boldsymbol{\theta} | \boldsymbol{\theta} \in \mathcal{D}_{\boldsymbol{\theta}}, \theta_i = 0, \forall i > r\}$. We rewrite the Theorem 5.3 in the main text as follows:

Theorem A.6 *With Algorithm 6 in the main text, $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r)$ monotonically decreases as r increases, and the following inequality holds*

$$F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) \geq F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) \geq F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1}), \quad (\text{A.8})$$

where $\mathbf{e}_{r+1} \in \mathcal{D}_{\boldsymbol{\theta}}$ is the vector with all zeros except the $(r+1)$ -th entry being 1. We also have $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) = F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) = F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1})$ when Algorithm 1 converges at the r -th iteration.

Proof: We prove it in three steps. First, we prove that $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r)$ monotonically decreases as r increases. Then, we show that $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) \geq F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) \geq F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1})$ holds. Finally, we prove that $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) = F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) = F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1})$ when Algorithm 1 converges at the r -th iteration.

To show that $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r)$ monotonically decreases as r increases, we only need to prove $F(\boldsymbol{\alpha}^{r+1}, \boldsymbol{\theta}^{r+1}) \leq F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r)$. Note $F(\boldsymbol{\alpha}, \boldsymbol{\theta})$ is convex in $\boldsymbol{\theta}$ and concave in $\boldsymbol{\alpha}$, so we have

$$F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) = \min_{\boldsymbol{\theta} \in \mathcal{E}_r} \max_{\boldsymbol{\alpha} \in \mathcal{A}} F(\boldsymbol{\alpha}, \boldsymbol{\theta}) = \max_{\boldsymbol{\alpha} \in \mathcal{A}} \min_{\boldsymbol{\theta} \in \mathcal{E}_r} F(\boldsymbol{\alpha}, \boldsymbol{\theta}).$$

Similarly, we have $F(\boldsymbol{\alpha}^{r+1}, \boldsymbol{\theta}^{r+1}) = \max_{\boldsymbol{\alpha} \in \mathcal{A}} \min_{\boldsymbol{\theta} \in \mathcal{E}_{r+1}} F(\boldsymbol{\alpha}, \boldsymbol{\theta})$. Moreover, for each $\boldsymbol{\alpha}$, we have $\min_{\boldsymbol{\theta} \in \mathcal{E}_{r+1}} F(\boldsymbol{\alpha}, \boldsymbol{\theta}) \leq \min_{\boldsymbol{\theta} \in \mathcal{E}_r} F(\boldsymbol{\alpha}, \boldsymbol{\theta})$, because the feasible set $\mathcal{E}_{r+1} \supset \mathcal{E}_r$. Then, we can derive

$$\begin{aligned} F(\boldsymbol{\alpha}^{r+1}, \boldsymbol{\theta}^{r+1}) &= \max_{\boldsymbol{\alpha} \in \mathcal{A}} \min_{\boldsymbol{\theta} \in \mathcal{E}_{r+1}} F(\boldsymbol{\alpha}, \boldsymbol{\theta}) \\ &= \min_{\boldsymbol{\theta} \in \mathcal{E}_{r+1}} F(\boldsymbol{\alpha}^{r+1}, \boldsymbol{\theta}) \\ &\leq \min_{\boldsymbol{\theta} \in \mathcal{E}_r} F(\boldsymbol{\alpha}^{r+1}, \boldsymbol{\theta}) \\ &\leq \max_{\boldsymbol{\alpha} \in \mathcal{A}} \min_{\boldsymbol{\theta} \in \mathcal{E}_r} F(\boldsymbol{\alpha}, \boldsymbol{\theta}) \\ &= F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r). \end{aligned}$$

Thus, we have proved that $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r)$ monotonically decreases as r increases.

Now we show $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) \geq F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) \geq F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1})$ holds. The left part of (A.8) is obvious, since $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r)$ monotonically decreases. We only need to prove the right part, $F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) \geq F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1})$. Let us denote $J(\boldsymbol{\alpha}, \mathbf{K}) = \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})' \mathbf{K} (\boldsymbol{\alpha} \circ \mathbf{y})$. Recall $\mathbf{K}_{r+1}$ is obtained by using the most violated constraint in (A.7), so we have $\mathbf{K}_{r+1} = \arg \max_{\mathbf{K}_{\mathbf{M}}} J(\boldsymbol{\alpha}^r, \mathbf{K}_{\mathbf{M}})$ for any $\mathbf{M} \in \mathcal{M}$. Let us denote $\tilde{\tau} = J(\boldsymbol{\alpha}^r, \mathbf{K}_{r+1})$, then we have $J(\boldsymbol{\alpha}^r, \mathbf{K}_{\mathbf{M}}) \leq J(\boldsymbol{\alpha}^r, \mathbf{K}_{r+1}) = \tilde{\tau}$ for all $\mathbf{M} \in \mathcal{M}$, which means $(\boldsymbol{\alpha}^r, \tilde{\tau})$ is a feasible solution to (A.7). Because $(\boldsymbol{\alpha}^*, \tau^*)$ is the optimal solution to (A.7), we can derive

$$\begin{aligned}
 G(\boldsymbol{\alpha}^*, \tau^*) &\geq G(\boldsymbol{\alpha}^r, \tilde{\tau}) \\
 &= \mathbf{1}' \boldsymbol{\alpha}^r - \tilde{\tau} \\
 &= \mathbf{1}' \boldsymbol{\alpha}^r - J(\boldsymbol{\alpha}^r, \mathbf{K}_{r+1}) \\
 &= \mathbf{1}' \boldsymbol{\alpha}^r - \frac{1}{2}(\boldsymbol{\alpha}^r \circ \mathbf{y})' \mathbf{K}_{r+1} (\boldsymbol{\alpha}^r \circ \mathbf{y}) \\
 &= F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1})
 \end{aligned}$$

Since $F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) = G(\boldsymbol{\alpha}^*, \tau^*)$, we then have $F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) \geq F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1})$. Thus, we have proved that $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) \geq F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) \geq F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1})$ holds.

When Algorithm 1 converges at the r -th iteration, it means that we cannot find a feasible $\mathbf{M}$ which violates the constraint in (A.7). In other words, we have $(\boldsymbol{\alpha}^r \circ \mathbf{y})' \mathbf{K}_{\mathbf{M}} (\boldsymbol{\alpha}^r \circ \mathbf{y}) \leq \tau^r$ for any $\mathbf{M} \in \mathcal{M}$. Moreover, because $(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r)$ is the optimal solution to the MKL problem at the r -th iteration, we have $\frac{1}{2}(\boldsymbol{\alpha}^r \circ \mathbf{y})' \sum_r^\infty \theta_r \mathbf{K}_r (\boldsymbol{\alpha}^r \circ \mathbf{y}) = \tau^r$. Therefore, we have

$$\begin{aligned}
 F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1}) &= \mathbf{1}' \boldsymbol{\alpha}^r - (\boldsymbol{\alpha}^r \circ \mathbf{y})' \mathbf{K}_{r+1} (\boldsymbol{\alpha}^r \circ \mathbf{y}) \\
 &\geq \mathbf{1}' \boldsymbol{\alpha}^r - \tau^r \\
 &= \mathbf{1}' \boldsymbol{\alpha}^r - \frac{1}{2}(\boldsymbol{\alpha}^r \circ \mathbf{y})' \sum_r^\infty \theta_r \mathbf{K}_r (\boldsymbol{\alpha}^r \circ \mathbf{y}) \\
 &= F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r).
 \end{aligned}$$

Recall we have proved that $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) \geq F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) \geq F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1})$, so we conclude that $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) = F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) = F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1})$ when Algorithm 1 converges. This completes the proof.

A.3 Proof of Proposition 1

In the main text, we have shown that the dual form of our SHFA can be written as follows:

$$\begin{aligned} \min_{\mathbf{y} \in \mathcal{Y}, \mathbf{H} \succeq \mathbf{0}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad & -\frac{1}{2} \boldsymbol{\alpha}' (\mathbf{Q}_{\mathbf{H}, \mathbf{y}} + \mathbf{D}) \boldsymbol{\alpha} \\ \text{s.t.} \quad & \text{tr}(\mathbf{H}) \leq \lambda, \end{aligned} \quad (\text{A.9})$$

where $\mathbf{Q}_{\mathbf{H}, \mathbf{y}} = (\mathbf{S}(\mathbf{H} + \mathbf{I})\mathbf{S} + \mathbf{1}\mathbf{1}') \circ (\mathbf{y}\mathbf{y}') \in \mathbb{R}^{n \times n}$, $\mathbf{y} = [\mathbf{y}'_s, \mathbf{y}'_t, \mathbf{y}'_u]'$ is the label vector in which $\mathbf{y}_s$ and $\mathbf{y}_t$ are given and $\mathbf{y}_u$ is unknown, $\mathcal{Y} = \{\mathbf{y} \in \{-1, +1\}^n | \mathbf{y} = [\mathbf{y}'_s, \mathbf{y}'_t, \mathbf{y}'_u]', \mathbf{1}'\mathbf{y}_u = \delta\}$ is the domain of $\mathbf{y}$, $\boldsymbol{\alpha} = [\alpha_1^s, \dots, \alpha_{n_s}^s, \alpha_1^t, \dots, \alpha_{n_t}^t, \alpha_1^u, \dots, \alpha_{n_u}^u]' \in \mathbb{R}^n$ with α_i^s 's, α_i^t 's and α_i^u 's are the dual variables corresponding to the constraints for source samples, labeled target samples and unlabeled target samples, respectively, $\mathcal{A} = \{\boldsymbol{\alpha} | \boldsymbol{\alpha} \geq \mathbf{0}, \mathbf{1}'\boldsymbol{\alpha} = 1\}$ is the domain of $\boldsymbol{\alpha}$ and $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with the diagonal elements as $\frac{1}{C}$ for the labeled data from both domains and $\frac{1}{C_u}$ for the unlabeled target data.

Now we rewrite and prove Proposition 1 as follows:

Proposition 4 *The objective of (A.9) is lower-bounded by the optimum of the following optimization problem:*

$$\begin{aligned} \min_{\boldsymbol{\gamma} \in \mathcal{D}_{\boldsymbol{\gamma}}, \mathbf{H} \succeq \mathbf{0}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad & -\frac{1}{2} \boldsymbol{\alpha}' \left(\sum_l \gamma_l \mathbf{Q}_{\mathbf{H}, \mathbf{y}_l} + \mathbf{D} \right) \boldsymbol{\alpha} \\ \text{s.t.} \quad & \text{tr}(\mathbf{H}) \leq \lambda, \end{aligned} \quad (\text{A.10})$$

where $\mathbf{y}_l$ is the l -th feasible labeling candidates, $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_{|\mathcal{Y}|}]'$, is the coefficient vector for the linear combination of all feasible labeling candidates and $\mathcal{D}_{\boldsymbol{\gamma}} = \{\boldsymbol{\gamma} | \boldsymbol{\gamma} \geq \mathbf{0}, \mathbf{1}'\boldsymbol{\gamma} \leq 1\}$ is the domain of $\boldsymbol{\gamma}$.

Proof: For ease of presentation, let us define

$$\begin{aligned} F(\mathbf{y}) &= \min_{\mathbf{H} \succeq \mathbf{0}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} -\frac{1}{2} \boldsymbol{\alpha}' (\mathbf{Q}_{\mathbf{H}, \mathbf{y}} + \mathbf{D}) \boldsymbol{\alpha}, \quad \text{s.t.} \quad \text{tr}(\mathbf{H}) \leq \lambda, \\ G(\boldsymbol{\gamma}) &= \min_{\mathbf{H} \succeq \mathbf{0}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} -\frac{1}{2} \boldsymbol{\alpha}' \left(\sum_l \gamma_l \mathbf{Q}_{\mathbf{H}, \mathbf{y}_l} + \mathbf{D} \right) \boldsymbol{\alpha}, \quad \text{s.t.} \quad \text{tr}(\mathbf{H}) \leq \lambda, \end{aligned}$$

then the optimal solutions to (A.9) and (A.10) can be represented as $\mathbf{y}^* = \arg \min_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{y})$ and $\boldsymbol{\gamma}^* = \arg \min_{\boldsymbol{\gamma} \in \mathcal{D}_{\boldsymbol{\gamma}}} G(\boldsymbol{\gamma})$, respectively. Intuitively, we can construct a feasible solution $\boldsymbol{\gamma}_{\mathbf{y}^*} = [0, \dots, 0, 1, 0, \dots, 0]$ where the only non-zero entry corresponds to the optimal feasible labeling $\mathbf{y}^*$, and we have $G(\boldsymbol{\gamma}_{\mathbf{y}^*}) = F(\mathbf{y}^*)$. Since $\boldsymbol{\gamma}^*$ is the optimal solution to (A.10), we also have $G(\boldsymbol{\gamma}^*) \leq G(\boldsymbol{\gamma}_{\mathbf{y}^*}) = F(\mathbf{y}^*)$, which means the optimal objective of (A.10) is always lower than that of (A.9).

Publication

Journal Publications

- **Wen Li**, Lixin Duan, Dong Xu, and Ivor W.H. Tsang, “Learning with Augmented Features for Supervised and Semi-supervised Heterogeneous Domain Adaptation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 36 no. 6, pp. 1134-1148, June 2014.
- Zhen Cui, **Wen Li**, Dong Xu, Shiguang Shan, Xilin Chen, and Xuelong Li, “Flowing on Riemannian Manifold: Domain Adaptation by Shifting Covariance,” *IEEE Transactions on Cybernetics*, March 2014.
- Meina Kan, Dong Xu, Shiguang Shan, **Wen Li**, and Xilin Chen,, “Learning Prototype Hyperplanes for Face Recognition in the Wild,” *IEEE Transactions on Image Processing (T-IP)*, vol. 22, no. 8, pp. 3310-3316, August 2013.
- Lixin Duan, **Wen Li**, Ivor W.H. Tsang and Dong Xu, “Improving Web Image Search by Bag-based Re-ranking,” *IEEE Transactions on Image Processing (T-IP)*, vol. 20, no. 11, pp. 3280-3290, November 2011.

Conference Publications

- **Wen Li**^{*}, Li Niu^{*}, and Dong Xu, “Exploiting Privileged Information from Web Data for Image Categorization,” in *Proceedings of European Conference on Computer Vision (ECCV)*, 2014. (*indicates equal contribution)
- Zheng Xu, **Wen Li**, Li Niu, and Dong Xu, “Exploiting Low-rank Structure from Latent Domains for Domain Generalization,” in *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

- Lin Chen, **Wen Li**, and Dong Xu, “Recognizing RGB Images by Learning from RGB-D Data,” in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- Lixin Duan, Yanwu Xu, **Wen Li**, Lin Chen, Damon W. Wong, Tien Y. Wong, and Jiang Liu, “Incorporating Privileged Genetic Information for Fundus Image Based Glaucoma Detection,” *accepted by Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2014.
- Huiying Liu, Dong Xu, Qingming Huang, **Wen Li**, Min Xu, and Stephen Lin, “Semantically-based Human Scanpath Estimation with HMMs,” in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2013.
- Hoang Anh B. Nguyen, and **Wen Li**, “Pose-Robust Representation for Face Verification in Unconstrained Videos,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2013.
- Zhen Cui, **Wen Li**, Dong Xu, Shiguang Shan, and Xilin Chen, “Fusing Robust Face Region Descriptors via Multiple Metric Learning for Face Recognition in the Wild,” in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- **Wen Li**, Lixin Duan, Ivor W. Tsang, and Dong Xu, “Co-Labeling: A New Multi-view Learning Approach for Ambiguous Problems,” in *Proceedings of IEEE International Conference on Data Mining (ICDM)*, 2012.
- **Wen Li**, Lixin Duan, Ivor W. Tsang, and Dong Xu, “Batch Mode Adaptive Multiple Instance Learning for Computer Vision Tasks,” in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 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 (ICCV)*, pp. 2049–2055, 2011.