Transfer Learning for Cross-domain Visual Recognition

Jing Zhang
*University of Wollongong*

Follow this and additional works at: [https://ro.uow.edu.au/theses1](https://ro.uow.edu.au/theses1)

*University of Wollongong*

Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following: This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author. Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

Unless otherwise indicated, the views expressed in this thesis are those of the author and do not necessarily represent the views of the University of Wollongong.

**Recommended Citation**


Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
Transfer Learning for Cross-domain Visual Recognition

Jing Zhang

This thesis is presented as part of the requirements for the conferral of the degree:

Doctor of Philosophy

Supervisors:
A/PR. Wanqing Li & PR. Philip Ogunbona

The University of Wollongong
School of Computing and Information Technology

April 11, 2019
Declaration

I, Jing Zhang, declare that this thesis is submitted in partial fulfilment of the requirements for the conferral of the degree Doctor of Philosophy, from the University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

______________________________

Jing Zhang

April 11, 2019
Abstract

Machine learning algorithms usually require a huge amount of training examples to learn a new model from scratch and often fail to apply the learned model to test data acquired from the scenarios different from those of the training data mainly due to domain divergence and task divergence. Transfer learning tries to use previously available data, models or knowledge effectively for a new domain or task with scarce data. This thesis focuses on addressing the cross-domain visual recognition using transfer learning.

First, a comprehensive literature review of transfer learning methods for cross-dataset visual recognition is presented by taking a problem-oriented perspective. Second, though there has been extensive research on unsupervised domain adaptation, the performance on the target domain is still far from comparable to that without distribution shift. Hence, a novel feature transformation-based method on unsupervised domain adaptation is proposed by taking both geometrical and statistical shift into consideration, and the performance is improved compared to previous methods. Third, a novel classifier-based unsupervised domain adaptation method is proposed by presenting a new perspective that the unsupervised domain adaptation can be formulated as a multi-task learning problem. This formulation removes the commonly used shared classifier assumption in previous methods and proposes unshared classifiers for the source and target domains for exploiting more domain specific features. Fourth, an important weighted adversarial nets-based method for partial domain adaptation is proposed, where the target domain has less number of classes compared to the source domain. Different from previous domain adaptation methods that generally assume the identical label spaces, a more realistic scenario that requires adaptation from a larger and more diverse source domain to a smaller target domain with less number of classes is considered. Last, a new research topic called multi-source domain expansion (MSDE) is introduced, which expands the source domains to include the new domain, such that the learned model is capable to perform well on both the new target domain and the old source domains. The MSDE is different from traditional domain adaptation whose target domain is defined only as the new domain excluding the source domains. MSDE is also different from multi-task learning, lifelong Learning, and incremental learning all of which assume no domain shift among different tasks. Specifically, a novel method is proposed for unsupervised MSDE without source data.
Acknowledgments

First, I would like to express my utmost gratitude to my supervisors Associate Professor Wanqing Li and Professor Philip Ogunbona for their invaluable guidance and advice of my Ph.D. study and research. During my Ph.D. studies, my principal supervisor, Associate Professor Wanqing Li, gives me freedom and confidence to explore my research interest. He not only provides the guidance on my research but also teaches me the attitude towards research, critical, concrete and precise. My co-supervisor Professor Philip Ogunbona gives me insightful advice on my research and the invaluable skills for writing papers. Without their patience, motivation, guidance, and encouragement, this thesis would not have been possible.

I would like to thank my colleagues and friends Dr. Pichao Wang, Zewei Ding, Dr. Lijuan Zhou, Shuai Li, Yuyao Zhang, Chang Tang, Song Liu, Dr. Hongda Tian, Chuankun Li, Huogen Wang, Dr. Zhimin Gao, and Dr. Jianjia Zhang for their constructive comments and suggestions, as well as the accompany during my Ph.D. study and living in Australia.

My gratitude is also expressed to my parents for their everlasting and unconditional love, care, and support throughout my life. My sincere thanks also go towards my husband, Xiaoyu Yu for his care, comfort, and encouragement.

Last, I would like to express my grateful acknowledgement to the China Scholarship Council (CSC) and the University of Wollongong for their financial support to my Ph.D. study.
# Contents

Abstract iv

1 Introduction 1
  1.1 Background ............................................................... 1
  1.2 Research Questions .................................................... 3
  1.3 Contributions ............................................................. 4
  1.4 Publications ............................................................... 6
    1.4.1 Published & Accepted Papers ...................................... 6
    1.4.2 Submitted Papers .................................................... 6
  1.5 Organization of the thesis ............................................. 7

2 Literature Review 8
  2.1 Introduction ............................................................. 8
  2.2 Overview ................................................................. 9
    2.2.1 Problem-oriented Taxonomy of Cross-dataset Recognition .... 9
    2.2.2 Approaches .......................................................... 10
  2.3 Homogeneous Feature Spaces and Label Spaces ..................... 18
    2.3.1 Labelled Target Dataset ........................................... 18
    2.3.2 Labelled plus Unlabelled Target Dataset ......................... 19
    2.3.3 Unlabelled Target Dataset ........................................ 20
    2.3.4 Imbalanced Unlabelled Target Dataset ............................ 23
    2.3.5 Sequential Labelled Target Data ................................ 24
    2.3.6 Sequential Unlabelled Target Data ............................... 24
    2.3.7 Unavailable Target Data .......................................... 25
  2.4 Heterogeneous Feature Spaces .......................................... 26
    2.4.1 Labelled Target Dataset ........................................... 26
    2.4.2 Labelled plus Unlabelled Target Dataset ......................... 27
    2.4.3 Unlabelled Target Dataset ........................................ 27
  2.5 Heterogeneous Label Spaces ............................................ 29
    2.5.1 Labelled Target Dataset ........................................... 29
    2.5.2 Unlabelled Target Dataset ........................................ 31
### 2.5.3 Sequential Labelled Target Data
31

### 2.5.4 Unavailable Target Data
32

### 2.5.5 Unlabelled Source Dataset
33

### 2.6 Heterogeneous Feature Spaces and Label Spaces
34

#### 2.6.1 Labelled Target Dataset
34

#### 2.6.2 Sequential Labelled Target Data
34

### 2.7 Datasets
35

#### 2.7.1 Object Recognition
35

#### 2.7.2 Hand-Written Digit and Character Recognition
36

#### 2.7.3 Face Recognition
37

#### 2.7.4 Person Re-identification
37

#### 2.7.5 Scene Categorization
38

#### 2.7.6 Action Recognition
38

#### 2.7.7 Video Event Detection
39

### 3 Joint Geometrical and Statistical Alignment for Visual Domain Adaptation
41

#### 3.1 Motivation
41

#### 3.2 Joint Geometrical and Statistical Alignment
42

##### 3.2.1 Problem Definition
43

##### 3.2.2 Formulation
43

##### 3.2.3 Optimization
47

##### 3.2.4 Kernelization Analysis
47

#### 3.3 Experiments
49

##### 3.3.1 Synthetic Data
49

##### 3.3.2 Real World Datasets
50

#### 3.4 Summary
55

### 4 Unsupervised Domain Adaptation: A Multi-task Learning-based Method
56

#### 4.1 Motivation
56

#### 4.2 Proposed Method
57

##### 4.2.1 Formulation
58

##### 4.2.2 Regularized Least Squares Algorithm
61

##### 4.2.3 Support Vector Machines Algorithm
62

#### 4.3 Experiments
63

##### 4.3.1 Results
64

##### 4.3.2 Parameter Sensitivity
68

#### 4.4 Summary
68
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Importance Weighted Adversarial Nets for Partial Domain Adaptation</td>
<td>70</td>
</tr>
<tr>
<td>5.1 Motivation</td>
<td>70</td>
</tr>
<tr>
<td>5.2 Proposed Method</td>
<td>72</td>
</tr>
<tr>
<td>5.2.1 Adversarial Nets-based Domain Adaptation</td>
<td>73</td>
</tr>
<tr>
<td>5.2.2 Importance Weighted Adversarial Nets-based Domain Adaptation</td>
<td>74</td>
</tr>
<tr>
<td>5.3 Experiments</td>
<td>77</td>
</tr>
<tr>
<td>5.3.1 Set-ups</td>
<td>77</td>
</tr>
<tr>
<td>5.3.2 Results and Analysis</td>
<td>79</td>
</tr>
<tr>
<td>5.4 Summary</td>
<td>82</td>
</tr>
<tr>
<td>6 Domain Expansion from Multiple Sources</td>
<td>87</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>87</td>
</tr>
<tr>
<td>6.2 Unsupervised MSDE without Source Data</td>
<td>89</td>
</tr>
<tr>
<td>6.2.1 Multi-source and New Domain Alignment</td>
<td>89</td>
</tr>
<tr>
<td>6.2.2 Learning of Source Model Weight</td>
<td>90</td>
</tr>
<tr>
<td>6.2.3 Source Domain Information Preservation</td>
<td>92</td>
</tr>
<tr>
<td>6.2.4 Overall Model</td>
<td>92</td>
</tr>
<tr>
<td>6.3 Experiments</td>
<td>93</td>
</tr>
<tr>
<td>6.3.1 Datasets and Implementation Details</td>
<td>93</td>
</tr>
<tr>
<td>6.3.2 Results</td>
<td>95</td>
</tr>
<tr>
<td>6.3.3 Discussion and Analysis</td>
<td>96</td>
</tr>
<tr>
<td>6.3.4 Ablation Studies and Parameter Sensitivities</td>
<td>98</td>
</tr>
<tr>
<td>6.4 Summary</td>
<td>101</td>
</tr>
<tr>
<td>7 Conclusions and Future Work</td>
<td>102</td>
</tr>
<tr>
<td>7.1 Conclusions</td>
<td>102</td>
</tr>
<tr>
<td>7.2 Challenges and Future Directions</td>
<td>104</td>
</tr>
<tr>
<td>7.2.1 Deep Transfer Learning</td>
<td>104</td>
</tr>
<tr>
<td>7.2.2 Partial Domain Adaptation</td>
<td>105</td>
</tr>
<tr>
<td>7.2.3 Transfer Learning from Multiple Sources</td>
<td>105</td>
</tr>
<tr>
<td>7.2.4 Domain Expansion</td>
<td>106</td>
</tr>
<tr>
<td>7.2.5 Online Transfer Learning</td>
<td>107</td>
</tr>
<tr>
<td>7.2.6 Data Imbalance</td>
<td>107</td>
</tr>
<tr>
<td>7.2.7 Few-shot and Zero-shot Learning</td>
<td>107</td>
</tr>
<tr>
<td>7.2.8 Cross-modal Recognition</td>
<td>108</td>
</tr>
<tr>
<td>7.2.9 Transfer Learning from Weakly Labelled Web Data</td>
<td>108</td>
</tr>
<tr>
<td>7.2.10 Self-taught Learning</td>
<td>109</td>
</tr>
<tr>
<td>7.2.11 Large Scale and Versatile Datasets for Transfer Learning</td>
<td>109</td>
</tr>
</tbody>
</table>

Bibliography 110
Chapter 1

Introduction

1.1 Background

Humans have exceptional ability to transfer learning from one context to another con-
text [283, 209]. Machine learning algorithms mostly inspired by human brains, how-
ever, usually require a huge amount of training examples to learn a new model from
scratch and often fail to apply the learned model to test data acquired from the scenar-
ios different from those of the training data mainly due to domain divergence and task
divergence [200]. This is particularly true in visual recognition [258] where the external
factors such as environments, lighting, background, sensor types, view angles, and post-
processing can cause the distribution shift or even feature space divergence of the same
task in two datasets or even the tasks, e.g. categories of the objects, are different.

To use previously available data effectively for current tasks with scarce data, models
or knowledge learned from one domain have to be transferred to a new domain for the
current task. Transfer learning has been actively researched in the past decade and one
of its topics, domain adaptation, has been especially extensively researched, where the
previous and current tasks are the same.

We begin with the definitions of terminologies used in the context of this thesis.

**Definition 1 (Domain [200])** “A domain is defined as \( D = \{ \mathcal{X}, P(x) \} \), which is com-
posed of two components: a feature space \( \mathcal{X} \) and a marginal probability distribution
\( P(x) \), where \( x \in \mathcal{X} \).”

**Definition 2 (Task [200])** “Given a specific domain, a task is defined as \( T = \{ \mathcal{Y}, f(x) \} \),
which is composed of two components: a label space \( \mathcal{Y} \) and a predictive function \( f(x) \),
where \( f(x) \) can be seen as a conditional distribution \( P(y|x) \) and \( y \in \mathcal{Y} \).”

**Definition 3 (Dataset)** A dataset is defined as \( S = \{ N, \mathcal{X}, P(x), \mathcal{Y}, f(x) \} \), which is a
collection of \( N \) data that belong to a specific domain \( D = \{ \mathcal{X}, P(x) \} \) with a specific
task \( T = \{ \mathcal{Y}, f(x) \} \).
Often $P(x)$ and $f(x)$ are unknown and need to be estimated and learned respectively. If for each sample in the dataset $\mathcal{N}$ its label $y \in \mathcal{Y}$ is given, $S$ is labelled, Otherwise, $S$ is unlabelled.

**Definition 4 (Transfer Learning [200])** “In general, given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.” Note that a special topic where $\mathcal{T}_S = \mathcal{T}_T$ and $\mathcal{D}_S \neq \mathcal{D}_T$ is known as Domain Adaptation. Specifically, in the context of cross-dataset recognition, the aim of transfer learning is to learn a robust classifier $f(x)$ from a dataset (i.e. target dataset $\mathcal{T}_T$) by effectively utilising the knowledge offered through other datasets (i.e. source datasets $\mathcal{T}_S$).

Many other general terms, such as domain adaptation, domain generalization, inductive learning, transductive learning, semi-supervised learning and unsupervised learning, are also related to transfer learning.

In domain adaptation, as mentioned in Definition 4, the tasks between the source and the target domains are the same, while the only difference between the two domains is the distributions of the data. Domain adaptation aims at employing previous labelled source domain data to boost the task in the new target domain, where the distribution shift exists between domains. Based on the availability of target labelled data, domain adaptation can be generally divided into supervised, semi-supervised, and unsupervised domain adaptation. The supervised domain adaptation requires a certain amount, usually insufficient, of labelled training sample in the target domain, semi-supervised domain adaptation requires a small amount of labelled plus a certain amount of unlabelled target data, and the unsupervised domain adaptation requires only unlabelled target domain data. Note that unsupervised domain adaptation is different from traditional unsupervised learning. In unsupervised learning, both training data and test data are from the same distribution and there is no labelled data in the training stage. Differently, in unsupervised domain adaptation the training data consist of both labelled source domain data and unlabelled target domain data and hence the “unsupervised” here is with regard to the target data.

Similar to domain adaptation, domain generalization also assumes the same tasks but different distributions between the source and the target domains. However, the objective of domain generalization is to generalize multiple related source domains to an unseen target domain. Hence, it assumes multiple related source domains are available and there is no training data in the target domain.

Inductive learning is traditional supervised learning. It learns a model from labelled data, and tries to predict the labels of the test data that have not been seen or known about. By contrast, transductive learning [127] refers to the situation where all test data are required to be seen at training time, and that the learned model cannot be reused
for future data. Similar to transductive learning, Semi-supervised learning requires both labelled and unlabelled data in the training stage. It is closely related to transductive learning. However, the goal of transductive learning is to classify the given test data of interest while the goal of semi-supervised learning is to find the optimal function that minimizes the risk functional. In other words, transductive learning can only classify the given test data while semi-supervised learning can generalize to unseen test data. Similar to domain adaptation, transductive learning and semi-supervised learning also requires both labelled and unlabelled data. However, the difference is that in domain adaptation, the labelled and unlabelled data are from different domains with different distributions. By contrast, the labelled and unlabelled data in transductive learning and semi-supervised learning are from the same distribution.

Over the past decade, domain adaptation (one of the topics in transfer learning) has attracted increasing attention. It deals with the cases where the source and target domains are different while the source and target tasks are identical. The key challenge of domain adaptation is how to effectively reduce the domain shift, such that the previously labelled source domain data can be employed to boost the task in the new target domain without the need of relabelling a large amount of data in the target domain. Based on the availability of target labelled data, domain adaptation can be generally divided into supervised, semi-supervised, and unsupervised domain adaptation. The supervised and semi-supervised approach requires a certain amount of labelled training samples in the target domain and the unsupervised one requires no labelled data. In this thesis, we focus on unsupervised domain adaptation where only unlabelled target domain data are available.

1.2 Research Questions

This thesis focuses on three challenging research problems in the context of unsupervised domain adaptation.

1. Despite the extensive research on unsupervised domain adaptation, using either traditional shallow learning-based methods [13, 200, 177, 178, 93, 74, 313] or deep learning-based methods [263, 174, 180, 175, 310, 246, 264, 82, 262, 23, 165, 186], the performance obtained using these domain adaptation methods on the target domain data is still far from comparable to the performance obtained using the model trained on sufficient target domain labelled training data. Hence, the first research question is how to design more advanced domain adaptation methods to boost the performance.

2. Current domain adaptation methods mainly assume the identical label spaces between the source and target domains. However, in the real world applications, it is more realistic that the target domain only contains a subset of source domain
classes because the source domain is generally assumed to be larger and more diverse compared to the target domain. In addition, when the target domain classes are unknown, a larger source domain dataset with more classes has more potential to cover the target domain classes for effective transferring of knowledge. Thus, the second research question is how to effectively transfer from a source domain with more classes to a target domain with fewer classes, which is also named partial domain adaptation (or partial transfer learning [27]). Compared to traditional domain adaptation, this partial domain adaptation has rarely been addressed to date.

3. The third research question is inspired by the observation that domain adaptation methods are concerned with the new domain performance regardless of the source domain performances. For example, when the adapted model is faced with the unseen data from both source and new domains, the domain adaptation methods may fail if the source domain performance is not preserved. Hence, the third research question is how to expand the source domains (rather than simply adapt the source domains), such that the learned model is capable to perform well on both the new domain and the source domains. We name this problem as domain expansion.

1.3 Contributions

The main contributions of this thesis are summarized as follows.

1. This thesis takes a problem-oriented perspective and presents a comprehensive literature review of transfer learning, both shallow and deep, for cross-dataset visual recognition. Specifically, it categorises the cross-dataset recognition into seventeen problems based on a set of carefully chosen data and label attributes. Such a problem-oriented taxonomy has allowed us to examine how different transfer learning approaches tackle each problem and how well each problem has been researched to date. The comprehensive problem-oriented review of the advances in transfer learning with respect to the problems has not only revealed the challenges in transfer learning for visual recognition, but also the problems (e.g. eight of the seventeen problems) that have been scarcely studied. This literature review not only presents an up-to-date technical review for researchers, but also a systematic approach and a reference for a machine learning practitioner to categorise a real problem and to look up for a possible solution accordingly.

2. A unified framework that reduces the shift between domains both statistically and geometrically is proposed, referred to as Joint Geometrical and Statistical Alignment (JGSA), for unsupervised domain adaptation. Specifically, we learn two coupled projections that project the source domain and target domain data into low-
dimensional subspaces where the geometrical shift and distribution shift are reduced simultaneously. The objective function can be solved efficiently in a closed form. Extensive experiments have verified that the proposed method significantly outperforms several state-of-the-art domain adaptation methods on a synthetic dataset and three different real-world cross-domain visual recognition tasks.

3. A new perspective to formulate unsupervised domain adaptation as a multi-task learning problem is developed. This formulation removes the commonly used assumption in the classifier-based adaptation approach that a shared classifier exists for the same task in different domains. Two novel classifier-based adaptation algorithms are proposed upon the formulation using Regularized Least Squares and Support Vector Machines respectively, in which unshared classifiers between the source and target domains are assumed and jointly learned to effectively deal with large domain shift. Experiments on both synthetic and real-world cross-domain recognition tasks have shown that the proposed methods outperform several state-of-the-art unsupervised domain adaptation methods.

4. An importance weighted adversarial nets-based method for partial domain adaptation where the target domain has less number of classes compared to the source domain is developed. Previous domain adaptation methods generally assume the identical label spaces, such that reducing the distribution divergence leads to feasible knowledge transfer. However, such an assumption is no longer valid in a more realistic scenario that requires adaptation from a larger and more diverse source domain to a smaller target domain with less number of classes. A novel adversarial nets-based partial domain adaptation method is proposed to identify the source samples that are potentially from the outlier classes and, at the same time, reduce the shift of shared classes between domains.

5. The thesis introduces a new research topic called multi-source domain expansion (MSDE). Unlike traditional domain adaptation in which the target domain is the domain defined by new data, in MSDE the target domain is formed jointly by the source domains and the new domain (hence, domain expansion) and the label function to be learned must work for the expanded domain. Specifically, a method for unsupervised MSDE is proposed where only models of source domains and unlabelled new domain data are available. The proposed method is based on the observation that by feeding the unlabelled data in the new domain into different source models, the biases among domains and the discriminative information in the new domain can be revealed. The method adapts the source models to the new domain while preserving their performance in the source domains simultaneously. Experimental results on the VLCS, ImageCLEF_DA and PACS datasets have verified the effectiveness of the proposed method.
CHAPTER 1. INTRODUCTION

1.4 Publications

1.4.1 Published & Accepted Papers


1.4.2 Submitted Papers


[2] Jing Zhang, Wanqing Li, Philip Ogunbona, Domain Expansion from Multiple Sources. Submitted to AAAI Conference on Artificial Intelligence (AAAI), 2019
1.5 Organization of the thesis

The rest of this thesis is organized as follows:

Chapter 2 takes a problem-oriented perspective and presents a comprehensive literature review of transfer learning for cross-dataset visual recognition.

Chapter 3 proposes a novel feature transformation-based method for unsupervised domain adaptation problem by jointly taking the geometrical and statistical shift between source and target domains into consideration.

Chapter 4 proposes a novel multi-task learning-based method for unsupervised domain adaptation problem by learning unshared classifiers between the source and target domains to deal with the large shift.

Chapter 5 proposes an importance weighted adversarial nets method for unsupervised partial domain adaptation, where the source domain is larger and more diverse than the target domain (i.e. the target domain only contains a subset of classes of the source domain).

Chapter 6 introduces a new problem and concept called multi-source domain expansion, which focuses on the expansion of source domains using the pre-learned source models and the unlabelled target domain data such that the learned model on the expanded domain performs well on both target domain and the source domains. An effective solution to the new problem is proposed.

Chapter 7 concludes this thesis and discusses the future works.
Chapter 2

Literature Review

2.1 Introduction

This chapter takes a new problem-oriented perspective and presents a comprehensive review of transfer learning methods for cross-dataset visual recognition. Specifically,

- It defines a set of data and label attributes, categorizes in a fine-grained way the cross-dataset recognition into seventeen problems based on these attributes, and presents a comprehensive review of the transfer learning methods, both shallow and deep, developed to date for each problem.

- An assessment of the suitability of widely used datasets for transfer learning in evaluating algorithms for each of the seventeen problems is provided.

- The problem-oriented taxonomy has allowed us to examine how different transfer learning approaches tackle each problem, how well each problem has been studied to date and the available solutions to each problem.

- This chapter not only presents an up-to-date technical review for researchers, but also a systematic approach and a reference for a machine learning practitioner to categorize a real problem and to look up for a possible solution accordingly.

The rest of the chapter is organized as follows. Section 2.2 defines the problem-oriented taxonomy of cross-dataset recognition, and summarises the transfer learning approaches to cross-dataset recognition. The seventeen problems identified in the taxonomy are categorized into four scenarios: homogeneous feature and label spaces, heterogeneous feature spaces, heterogeneous label spaces and heterogeneous feature and label spaces. Sections 2.3 through 2.6 review and analyze respectively the advances of techniques in addressing the problems under the four scenarios. Section 2.7 discusses and examines the suitability of the most commonly used datasets for cross-dataset transfer learning for all the problems.
2.2 Overview

This section introduces a problem-oriented taxonomy of cross-dataset recognition and provides a summary of the approaches that have been developed for transfer learning.

2.2.1 Problem-oriented Taxonomy of Cross-dataset Recognition

In cross-dataset recognition, there are often two datasets. One, referred to as a source dataset, is used for training and the other, referred to as a target dataset, is to be recognized. Their domains and/or tasks are different and their characteristics determine what methods can or should be used. In this chapter, we define a set of attributes to characterize the source or target datasets. These attributes have led to a comprehensive taxonomy of cross-dataset recognition problems that provides a unique perspective for this survey.

- **Attributes on data:**
  - Feature space: the consistency of feature spaces (i.e. different feature extraction methods or different data modalities) between the source and target datasets.
  - Data availability: the availability and sufficiency of target data in the training stage.
  - Balanced data: whether the numbers of data samples in each class are balanced.
  - Sequential data: whether the data are sequential and evolving over time.

- **Attributes on label:**
  - Label availability: the availability of labels in source and target datasets.
  - Label space: whether the data categories of the two datasets are identical.

Based on these attributes, the following four scenarios are defined as the first layer of the problem taxonomy to guide the survey.

- **Homogeneous feature spaces and label spaces:** The feature spaces and label spaces of the source and target datasets are identical. But domain divergence (i.e. different data distributions) exists across the source and target datasets.

- **Heterogeneous feature spaces:** the feature spaces of the source and target datasets are different (i.e. domain divergence occurs), but their label spaces are the same.

- **Heterogeneous label spaces:** the label spaces of the source and target datasets are different (i.e. task divergence occurs), but their feature spaces are the same.
• Heterogeneous feature spaces and label spaces: both the feature spaces and the label spaces of the source and target datasets are different (i.e. both domain and task divergence occurs).

The problems corresponding to the four scenarios are further divided into sub-problems using other data attributes such as the data being balanced and/or sequential. Fig. 2.1 shows the problem-oriented taxonomy for cross-dataset recognition, which shows seventeen different problems.

2.2.2 Approaches

Many approaches have been developed for transfer learning across datasets [200] at the instance level, i.e. re-weighting some source samples based on their divergence from the target domain, at the feature level, i.e. learning “good” feature representations that have minimum domain shift, and at the classifier level, i.e. learn an optimal target classification model by using the data from both source and target domains as well as the source model. This section summarises several most typical approaches to transfer learning for cross-dataset recognition, including Statistical approach, Geometric approach, Higher-level Representation, Correspondence approach, Class-based approach, Self Labelling, and Hybrid approach. These approaches have been reported explicitly or implicitly in the literature. In particular, the basic assumptions of each approach are analyzed and presented in this section. Moreover, several commonly used methods are illustrated under each approach.

In the following, \( \{ X_s, Y_s \} \) represents a source dataset drawn from distribution \( P_s(X_s, Y_s) \) and \( \{ X_t, Y_t \} \) refers to a target dataset drawn from distribution \( P_t(X_t, Y_t) \).

**Statistical Approach** is employed in transferring the knowledge at the levels of instances, features and classifiers by measuring the statistical distribution shift between the source and target datasets. This approach generally assumes sufficient data in each dataset to approximate the respective statistic distributions. The typical methods are,

1. Instance reweighting [115]

Reweight the source domain samples to direct a sample selection de-biasing procedure using unlabelled data in source and target domains:

\[
E_{x \sim P_t}[l(x, y, \theta)] = E_{x \sim P_s}[\beta(x)l(x, y, \theta)] = E_{x \sim P_s}\frac{P_s(x)}{P_t(x)}l(x, y, \theta)
\]  
(2.1)

provided that the support of \( P_t \) is contained in the support of \( P_s \), where \( \beta(x) \) is a reweighting factor for source samples, \( l(x, y, \theta) \) is a loss functions with parameter \( \theta \).
**Figure 2.1:** A problem-oriented taxonomy for cross-dataset recognition including the number of papers that are found to address the problems.
CHAPTER 2. LITERATURE REVIEW

2. Feature space mapping [202]
Find a domain invariant feature map \( \phi \) such that the marginal distributions between domains in the mapped feature space is small using Maximum Mean Discrepancy (MMD) metric [202]:

\[
D_{\text{MMD}}(P_s, P_t) = \left\| \frac{1}{n_s} \sum_{x_i \in X_s} \phi(x_i) - \frac{1}{n_t} \sum_{x_j \in X_t} \phi(x_j) \right\|_F^2 \tag{2.2}
\]

Except for the MMD metric, there are also several other commonly used measurements for comparing two distributions as presented below.

3. Classifier parameter mapping [214]
Align the two domains in the classifier space using MMD metric [202].

As mentioned, there are several commonly used metrics for measuring distribution shift, which are summarised as follows.

- **Kullback-Leibler divergence** [243]
  \[
  D_{\text{KL}}(P_s || P_t) = \int_X P_s(x) \log \frac{P_s(x)}{P_t(x)} dx \tag{2.3}
  \]

- **Jensen-Shannon divergence** [96]
  \[
  D_{\text{JS}}(P_s || P_t) = \frac{1}{2}(D_{\text{KL}}(P_s || P) + D_{\text{KL}}(P_t || P)) \tag{2.4}
  \]
  where \( P = 1/2(P_s + P_t) \).

- **Quadratic divergence** [237]
  \[
  D_{\text{Breg}}(P_s || P_t) = \int_X (P_s(x) - P_t(x))^2 dx \tag{2.5}
  \]

- **Hellinger distance** [9]
  \[
  D_{\text{Helli}}(P_s, P_t) = \int_X (\sqrt{P_s(x)} - \sqrt{P_t(x)})^2 dx \tag{2.6}
  \]

- **Mutual information** [233]
  \[
  I_{s,t}(X; Q) = H[\hat{q}_0] - \frac{1}{n} \sum_i H[\hat{q}_i] \tag{2.7}
  \]
  where \( X \) represents all the data from source and target domain, \( Q \) denotes the domain label (i.e. 0 for source domain, and 1 for target domain), \( \hat{q}_i \) is the two-dimensional posterior probability vector of assigning \( x_i \) to either the source or the
target, given all other data points from the two domains, and \( \hat{q}_0 \) is the estimated prior distribution of domain (i.e. \( \hat{q}_0 = 1/n \sum_i \hat{q}_i \)). By minimizing the mutual information between the data instance \( X \) and its (binary) domain label \( Q \), the domain shift is reduced.

- \( \mathcal{H} \) divergence [14, 13]

\[
D_{\mathcal{H}}(P_s, P_t) = 2 \sup_{h \in \mathcal{H}} |P_s[I(h)] - P_t[I(h)]|
\]

(2.8)

where \( \mathcal{H} \) is a hypothesis class on \( \mathcal{X} \), and \( I(h) \) is the set for which \( h \in \mathcal{H} \) is the characteristic function; that is \( x \in I(h) \iff h(x) = 1 \). \( \mathcal{H} \) divergence measures the distance between distributions in a hypothesis class \( \mathcal{H} \), which can be approximated by the empirical risk of a classifier that discriminates between instances drawn from \( P_s \) and instances drawn from \( P_t \).

- Maximum Mean Discrepancy (MMD) [202]

\[
D_{\text{MMD}}(P_s, P_t) = \| \frac{1}{n_s} \sum_{x_i \in \mathcal{X}_s} \phi(x_i) - \frac{1}{n_t} \sum_{x_j \in \mathcal{X}_t} \phi(x_j) \|^2_F
\]

, where \( \phi \) represents the kernel function that maps the original data to a reproducing kernel Hilbert space (RKHS). The MMD compares the statistic moments of distributions. If the \( \phi \) is a characteristic kernel (i.e. Gaussian kernels, or Laplace kernels), MMD compares all the orders of statistic moments, making MMD a metric on distributions.

**Geometric Approach** bridges datasets according to their geometrical properties. It assumes domain shift can be reduced using the relationship of geometric structures between the source and target datasets. Typical methods include

1. Subspace alignment [74]

   Learn a linear mapping \( M \) to align the source subspace to the target one.

\[
M = \arg \min_M \| A_s M - A_t \|^2_F
\]

(2.10)

where the subspaces \( A_s \) and \( A_t \) are pre-learned using PCA on the source and target domain respectively.

2. Intermediate subspaces [98, 93]

   Identify intermediate subspaces between the source and target, and then learn the information from these subspaces to convey the domain changes. The subspaces are identified with the Grassmann manifold \( \mathbb{G}_{N,d} \), and source subspace \( A_s \) and target
subspace $A_t$ are points on $G_{N,d}$. To bridge $A_s$ and $A_t$, the points on the geodesic paths between them (which are constant velocity curves on a manifold) are sampled to form the intermediate subspaces. Then both source and target data are projected to the obtained intermediate subspaces (either by sampling along the geodesic [98] or all of them [93]) to augment the data for help finding the correlations between domains.

3. Manifold alignment (without correspondence) [48]
Align the manifolds defined by source and target datasets without the correspondence between the samples in the source and target domains. Hence, the two manifolds can be aligned geometrically:

$$<A_s, A_t, F> = \arg\min_{A_s, A_t, F} \|K_s - FK_t F^T\|_F^2 + \|A_s^T X_s - A_t^T X_t F^T\|_F^2 + J(A_s, X_s) + J(A_t, X_t)$$

(2.11)

where the $K_s$ and $K_t$ are full adjacency matrix (i.e. $[K]_{ij} = d(X_i, X_j)$), $F \in \{0, 1\}^{n_s \times n_t}$ denotes the correspondence matrix that needs to be learned, $A_s$ and $A_t$ are linear projections of from two datasets that also need to be learned, and $J(A_s, X_s)$ and $J(A_t, X_t)$ are geometry preserving terms to preserve the manifold structures of respective domains.

**Higher-level Representation Approach** aims at finding higher-level representations that are representative, compact, and invariant between datasets. This approach does not assume the existence of labelled data, or the existence of correspondence set, but assumes that there exist the domain invariant higher-level representations between datasets. Note that this approach is commonly used together with other approaches for better transfer, but it is also used independently without any mechanism to reduce the domain divergence explicitly.

1. Sparse coding [215]
A dictionary is learned based on the source data, and then apply the learned dictionary to the target data to obtain the sparse codes of the target data.

$$D = \arg\min_{D, Z_s} \|X_s - DZ_s\| \quad \text{s.t.} \forall i, \|z_i\|_0 \leq T$$

(2.12)

$$Z_t^* = \arg\min_{Z_t} \|X_t - DZ_t\| \quad \text{s.t.} \forall j, \|z_j\|_0 \leq T$$

(2.13)

where $D$ is the learned dictionary on source data, $Z_s$ and $Z_t$ are the sparse codes.

2. Low-rank representation [230]
Find a subspace $A$ where each datum in the target domain can be linearly repre-
sent by the corresponding subspace in the source domain.

\[\langle A, Z \rangle = \arg \min_{A, Z} F(A, X_s) + \text{rank}(Z)\]

\[\text{s.t. } A^T X_s Z = A^T X_t\]  

(2.14)

Because \(Z\) is constrained to be low rank, each target sample is linearly represented by some subspace from a subspace union in the source domain. Hence, the structure information in the source and target domain is considered.

3. Deep Neural Networks [58, 219, 307]

Deep neural networks tends to learn transferable features by disentangling explanatory factors of variations underlying data samples. The pre-trained models on the source data can be used either as a feature extractor, or an initialization for the target task. Deep neural networks are also commonly used together with other transfer learning approaches to form hybrid ones.

4. Stacked Denoising Auto-encoders (SDAs) [89, 36]

Train the SDAs [266] to reconstruct the data from all the domains. It has been shown that SDAs can disentangle hidden factors by assuming the existence of generic concepts that are invariant to different domains [89, 36].

5. Attribute space [147, 2]

Use the human-specified high-level description (attributes) of target objects instead of training images to detect object in an image. Suppose \(\alpha^c = (\alpha_{c1}^1, ..., \alpha_{cm}^m)\) is the attribute representation for class \(c\) with \(m\) attributes in all the classes. In [147], the \(p(\alpha_m|x_s)\) is learned first by learning probabilistic classifiers for each attribute \(\alpha_m\) from the source dataset. In the test stage, the independent classifiers allow the prediction of attribute values of test samples, from which the test class label are inferred. Another strategy [2] is recognition by assuming a fixed transformation between the attributes and the class labels.

**Correspondence Approach** uses paired correspondence samples from different domains to construct the relationship between domains. A set of corresponding samples (i.e. the same object captured from different view angles, or by different sensors) are required. The typical methods are as follows.

1. Sparse coding with correspondence [329]

   The corresponding samples between domains are force to share the same sparse
codes:

\[
< D_s, D_t, Z > = \arg \min_{D_s, D_t, Z} \| X_t - D_t Z \|_F^2 + \| X_s - D_s Z \|_F^2
\]

s.t. \( \forall i, \| z_i \|_0 \leq T \)  

(2.15)

where \( D_s \) and \( D_t \) are the dictionaries, \( Z \) is the sparse codes.

2. Manifold alignment (with correspondence) [312]

Given a set of correspondence samples set \( C \) between domains, learn mapping matrices \( A_s \) and \( A_t \) for source and target set respectively to preserve the correspondence relationships after mapping:

\[
< A_s, A_t > = \arg \min_{A_s, A_t} \sum_{(i,j) \in C} \| A_s^T x_s^i - A_t^T x_t^j \|_2^2 + J(A_s, X_s) + J(A_t, X_t)
\]

(2.16)

where \( J(A_s, X_s) \) and \( J(A_t, X_t) \) are the manifold regularization terms which are used to preserve the intrinsic manifold structures of source and target domains.

**Class-based Approach** uses class label information as a guidance for connecting different datasets. Hence, the labelled samples from each dataset are assumed to be available, no matter sufficient or not. The commonly used methods include the following.

1. Feature augmentation [52]
   Each feature is augmented into three versions of it: a general version, a source-specific version and a target-specific version. The augmented source data will contain only general and source-specific versions. The augmented target data contains general and target-specific versions. The rest of the dimensions are appended with zeros. The augmented features are as follows,

\[
\Phi_s(x) = [x_g, x_s, 0]; \quad \Phi_t(x) = [x_g, 0, x_t]
\]

(2.17)

where \( x_g \) is the general version, \( x_s \) is the source specific version, and \( x_t \) is the target specific version.

2. Metric learning [224]
   Learn a metric such that the distance of samples from different domains with same labels are close while the distance of samples from different domains with different labels are far away:

\[
\min R(W) \quad s.t. \quad d_W(x_s, x_t) < u \quad \text{if} \quad y_s = y_t
\]

\[
d_W(x_s, x_t) > l \quad \text{if} \quad y_s \neq y_t
\]

(2.18)
where \( u, l \in \mathbb{R} \) are threshold parameters, \( d_W = (x_s - x_t)^T W (x_s - x_t) \) is the pairwise distance for some \( W \), and \( W \) is the matrix that needs to be learned.

3. Linear Discriminative Model [301]

Use the source classifier parameters to regularize the target classifier parameters:

\[
L = \min \sum_{x_s \in X_s} l(x_s, y_s, w_s) + \sum_{x_t \in X_t} l(x_t, y_t, w_t) + R(w_s, w_t) \tag{2.19}
\]

where \( l \) denotes some loss functions over data and labels, and \( R \) denotes some regularization between source and target classifiers.

4. Bayesian Model [71]

Present the source knowledge as a prior probability in the space of target model parameters. Specifically, the “general knowledge” from source domain categories are extracted and then represented in the form of a prior probability density function in the space of model parameters. Given a small training set in the target domain, this knowledge can be updated and produce a posterior density.

**Self Labelling** uses the source domain samples to train an initial model, which is used to obtain the pseudo labels of target domain samples. Then the target samples are incorporated to retrain the model. The procedure is carried iteratively until convergence.

1. Self-training [49, 251]

Initialize the target model parameters \( \theta_t \) using the source data (or re-weighted source data), i.e. \( \theta_t = \arg \max_{\theta} \sum_{x_s \in X_s} \log p(x_s, y_s | \theta) \). The pseudo labels of target samples can be obtained using the initial model, then use EM algorithm to iteratively refine the target model by incorporating some mechanisms (i.e. assign small weight to source samples that are dissimilar to target samples):

\[
\text{E step : } \hat{y}_t = \arg \max_c p(y_t = c | x_t, \theta_t) \tag{2.20}
\]

\[
\text{M step : } \theta_{t+1} = \arg \max_{\theta} \sum_{x_t \in X_t} w(x_t) \log p(x_t, \hat{y}_t | \theta) + \sum_{x_s \in X_s} w(x_s) \log p(x_s, y_s | \theta) \tag{2.21}
\]

where \( w(x_t) \) and \( w(x_s) \) are assigned weights to each target and source samples using some mechanisms.

**Hybrid Approach** combines two or more above approaches for better transferring of knowledge. The combination is generally used for feature representation transfer, classifier transfer, and hybrid knowledge transfer. Several example combinations are

1. Correspondence and Higher-level representation [113]
2. Higher-level representation and Statistic [179, 174, 281]

3. Statistic and Geometric [313]

4. Statistic and Self labelling [50]

5. Correspondence and Class-based [54]

6. Statistic and Class-based [60]

7. Higher-level representation and Class-based [332]

In the following sections, we present a comprehensive review of what approaches have been or can be used for the cross-dataset recognition problems shown in Figure 2.1.

2.3 Homogeneous Feature Spaces and Label Spaces

In this scenario, $\mathcal{X}_S = \mathcal{X}_T$ and $\mathcal{Y}_S = \mathcal{Y}_T$. Hence, the $\mathcal{S}$ and $\mathcal{T}$ are generally different in their distributions ($P(X, Y)$). Sufficiently labelled source domain data are generally assumed available and different assumptions are made on the target domain, leading to different sub-problems.

2.3.1 Labelled Target Dataset

In this problem, a small number of labelled data in target domain are available. However, the labelled target data are generally insufficient for learning an effective classifier. This is also called supervised domain adaptation or few-shot domain adaptation in the literature.

Class-based Approach The most commonly used approach in supervised domain adaptation is class-based one since the labelled data from both domains are available in the training stage. For example, Daumé III [52] propose a feature augmentation based method where each feature is replicated into a high-dimensional space containing the general and domain-specific version.

The idea of supervised metric learning has also been used [323, 210]. The core idea is to exploit the task relationships between domains to boost the target task. Another group of methods [301, 124, 292] transfer the parameters of discriminative classifiers (e.g. SVM) across datasets. Recently, Motiian et al. [189] propose to create pairs of source and target instances to handle the scarce target labelled data. In addition, they extend adversarial learning [96] to align the semantic information of classes.

A more realistic setting is that target samples from only a subset of classes are available. Then the adapted features are generalized to unseen categories (which are unseen in the target dataset, but have seen in the source dataset). Since it still assumes the same
label spaces between domains though some of the categories are not presented in the target training set, we discuss these methods under the problem of homogeneous label spaces setting. Generally, these methods assume the shift between domains is category-independent. For example, Saenko et al. [224] present a supervised metric learning-based method to learn a metric that minimizes the distribution shift using target labelled data from a subset of categories. Then the transformation is applied to unseen target test data that may come from different categories from the target training data. Similarly, some recent methods learn to recognize unseen target categories (but have seen in the source domain) under the deep learning frameworks by exploiting the semantic structure either via soft labels (which is the averaged softmax activations over all source samples in each category) [264] or by a Siamese architecture [190].

**Self Labelling**  Dai et al. [49] propose TrAdaBoost to extend boosting-based methods by decreasing the weights of the instances that are most dissimilar to the target distribution in order to weaken their impacts.

**Hybrid Approach**  The higher-level representation approach and class-based approach have been used together for better cross-dataset representation. For example, the discriminative dictionary can be learned such that the same class samples from different domains have similar sparse codes. [334, 232]. Except for the discriminative dictionary learning, the label information can also be used for guiding the deep neural networks to reduce domain shift. For example, Koniusz et al. [137] fuse the source and target CNN streams at the classifier level, where the scatters of the two network streams of the same class are aligned while the between-class are separated.

**2.3.2 Labelled plus Unlabelled Target Dataset**

Compared to the scenario where only limited labelled target data are presented, additional redundant unlabelled target data are also presented in training in this problem (often known as semi-supervised domain adaptation in the literature) to provide additional structural information. This setting is realistic in real-world applications because unlabelled data are easy to obtain.

**Correspondence Approach**  Zhai et al. [312] assume in addition to a set of labelled correspondence pairs between the source and target datasets, some unlabelled data from both datasets are also available. They proposed a manifold alignment method to learn explicit corresponding mappings from different manifolds to the underlying common embeddings, where the common embeddings should be consistent with the labelled corresponding pairs and also should preserve the local geometric structures of respective datasets.
CHAPTER 2. LITERATURE REVIEW

Class-based Approach  Duan et al. [61] extend SVM-based supervised classifier transfer methods with unlabelled target data. They proposed a regularizer which enforces that the learned target classifiers and the pre-learned source classifiers should have the similar decision values on the unlabelled target instances.

Self Labelling  Some researches extend distance-based classifiers, such as the k-Nearest Neighbour [254] and Nearest Class Mean [46] classifiers, to learn the domain invariant metric iteratively. Specifically, Tommasi and Caputo [254] present a method that learns a metric per class based on the NBNN algorithm, by progressively selecting target instances and combining it with a subset of the source data while imposing a large margin separation hyperplanes among classes. Similarly, Csurka et al. [46] extend the NCM classifier to a Domain Specific Class Means (DSCM) classifier and iteratively add high confidence unlabelled target samples to the training set.

A co-training-based method is proposed by [35] to facilitate the gradual inclusion of target features and instances in training. This method iteratively learns feature views and a target predictor upon the views.

Hybrid Approach  A group of methods for semi-supervised domain adaptation combines class-based and statistical approach to make use of both labelled and unlabelled target data. The key idea is that the statistical criteria (e.g. MMD metric between source data and unlabelled target data) are used as an additional constraint in discriminative learning methods (e.g. multiple kernel learning (MKL) [64, 60], or least square method [304]).

Yamada et al. [298] generalize the EASYADAPT method [52] to semi-supervised setting. They proposed to project input features into a higher dimensional space as well as estimate weights for the training samples based on the ratio of test and training marginal distributions in that space using unlabelled target samples.

Wu and Ji [284] introduce a constrained deep transfer feature learning method by incorporating correspondence approach into high-level representation approach. Specifically, the paired source and target samples are used for capturing the joint distribution to bridge the two domains. Then a large amount of additional unpaired source samples are transferred to the target domain through pseudo labelling for further target domain feature learning.

2.3.3 Unlabelled Target Dataset

In this problem, no labelled target domain data are available but sufficient unlabelled target domain data are observable for transfer learning. This problem is also named unsupervised domain adaptation. The unsupervised domain adaptation has attracted increasing attention nowadays, which is certainly more realistic and challenging.
Statistical Approach The MMD criterion is commonly used in unsupervised domain adaptation. Generally, the MMD distance between domains is reduced by re-weighting the samples [115, 248, 90], or mapping to another feature space [202, 10, 177, 313], or regularizing the source domain classifier using target domain unlabelled data [214, 176].

Except for MMD, other statistic criteria, such as Kullback-Leibler divergence [243], Hellinger distance [9], Quadratic divergence [237], and mutual information [233], are also used for comparing two distributions. Sun et al. [245] propose the CORrelation ALignment (CORAL) to minimize distribution divergence by mapping the covariance of data.

Instead of learning a global transformation, Optimal Transport [45] introduces a local transformation such that each source datum is mapped to target data and, at the same time, to preserve the marginal distribution.

Rather than assuming single domain in a dataset, some methods assume a dataset may contain several distinctive sub-domains due to the large variations in visual data. For example, Gong et al. [92] automatically discover latent domains from multi-source domains to characterize the inter-domain variations and, hence, to construct discriminative models.

Geometric Approach Gopalan et al. [98] proposed a Sampling Geodesic Flow (SGF) method by sampling intermediate subspace representations between the source and target generative subspaces. The two generative subspaces are viewed as two points on a manifold. Then they sample the intermediate subspaces on the geodesic flow between the two subspaces. Lastly, all the data are mapped to the concatenation of all the subspaces to obtain the final representation. Gong et al. [93] extend SGF to a geodesic flow kernel (GFK) method by proposing a kernel method, such that an infinite number of subspaces are integrated to represent the incremental changes. The methods in [98, 97] and [93, 91] open the opportunity for researches to construct intermediate representations to characterize the domain changes. For example, Zhang et al. [324] bridge the source and target domains by inserting virtual views along a virtual path for cross-view recognition. Rather than manipulating on the subspaces, Cui et al. [47] represent source and target domains as covariance matrices and interpolate some intermediate covariance matrices to bridge the two domains. Some methods [194, 290] are proposed to generate several intermediate domains by learning the domain-adaptive dictionaries between domains. The idea of intermediate domains is also employed in the deep learning framework [42].

Instead of modeling intermediate domains, some methods align the two domains directly [74, 4, 48, 181]. For instance, Fernando et al. [74] propose to align the source subspace to the target subspace directly by learning a linear transformation function.

Higher-level Representation The low-rank criterion is commonly used to learn the domain invariant representations [121, 229, 56]. Generally, these methods assume that
the data from different domains lie in a shared low-rank structure.

Bengio [16] argue that more transferable features can be learned by deep networks since they are able to extract the unknown factors of variation that are intrinsic to the data. Donahue et al. [58] propose the deep convolutional representations named DeCAF, where a deep CNN model is pre-trained using the source dataset (generally large-scale) in a fully supervised fashion. Then they transfer the features (defined by the pre-learned source convolutional network weights) to the target data. The deep auto-encoders are also used for the cross-dataset tasks by exploiting more transferable features by reconstruction [89, 130, 36, 125, 85]. For instance, Ghifary et al. [85] propose a Deep Reconstruction-Classification Network (DRCN) to learn a shared deep CNN model for both classification task of the source samples and reconstruction task of the target samples.

**Self Labelling** Recently, Panareda Busto and Gall [203] propose an open set domain adaptation problem, where only some of the classes are shared between source and target datasets. The task is to label all the target samples either by one of the classes shared between domains or as unknown. They solve this problem by first assigning some of the target data with the labels of the known classes in the source dataset and then reducing the shift between the shared classes in the source and target datasets by a subspace alignment method (similar to [74]). The two procedures are learned iteratively.

**Hybrid Approach** Combining different approaches generally trigger better transferring of knowledge. Some methods [329, 113] learn two dictionaries on pairs of correspondence samples and encourage the sparse representation of each sample pair to be similar. Some methods use both geometric and statistical approach [247, 313]. For example, Zhang et al. [313] propose to learn two projections for the source and target domain respectively to reduce the geometrical shift and statistical shift. Differently, Gholami et al. [88] jointly learn a low dimensional subspace and a classifier through a Bayesian learning framework.

Though deep networks can generally learn more transferable features [16, 58], the higher level features computed by the last few layers are usually task-specific and are not transferable to new target tasks [307]. Hence, some recent work imposes statistical approach into the deep learning framework (high-level representation approach) to further reduce domain bias. For instance, the MMD loss is incorporated into the objective of the deep models to reduce the divergence of marginal distributions [263, 174, 180, 265] or joint distributions [175] between domains. Instead of using MMD metric, Sun and Saenko [246] extend the CORrelation ALignment (CORAL) method [245] that aligns the covariance of the source and target data to a deep learning-based method. Zellinger et al. [310] propose the Central Moment Discrepancy (CMD) method, which aligns the higher order central moments of distributions through order-wise moment differences.
stead of the statistical approach, the self-labelling is also used in the deep neural network-based method. Saito et al. [225] propose an asymmetric tri-training method, where feature extraction layers are used to drive three classifier sub-networks. The first two networks are used to label unlabelled target samples and the third network is to learn the final adapted classifier to operate on the target domain with the pseudo-labels obtained on the first two networks.

The statistical approaches (e.g. MMD distance [281, 22], and $\mathcal{H}$ divergence [22]) are also incorporated into deep autoencoders for learning more transferable features.

Motivated by adversarial learning [96], the GAN-based domain adaptation methods are proposed with the key idea that the JS divergence between domains is reduced [82, 83, 262, 23]. For example, the gradient reversal algorithm (ReverseGrad) proposed by Ganin and Lempitsky [82] minimizes the $\mathcal{H}$-divergence by considering the domain invariance as a binary classification task and employing a gradient reversing strategy. Tzeng et al. [262] propose to learn separate feature extraction networks for different domains, and a domain classifier is incorporated such that the embeddings produced by the source or target CNN cannot be distinguished. Bousmalis et al. [23] propose a GAN-based method to adapt the source domain data from the pixel level, such that they are not distinguishable to the target domain data. Differently, Liu and Tuzel [172] propose a Coupled GAN (Co-GAN) method that learns a joint distribution by jointly modeling two GANs, where the first one generates the source data while the second generates the target images. Instead of enforcing samples from different domains to be non-discriminant, the CoGAN enforce the layers that decode high-level features to share the weights so as to enforce the assumption that the images from different domains share the same high-level representations but have different low-level representations.

2.3.4 Imbalanced Unlabelled Target Dataset

This problem assumes the target domain is class imbalanced and only with unlabelled data. Thus, the statistical approach can be used. This problem is quite common in practice and known as prior probability shift, or imbalanced data in classification. For instance, the abnormal activities (e.g. kick, punch, fight, and fall down) are much less frequent than normal activities (e.g. walk, sit, eat, and drink) in the video surveillance but require higher recognition rate.

Statistical Approach In the classification scenario, the prior probability ($P(Y)$) shift was often considered to be a class imbalance problem [119, 320]. Zhang et al. [320] tackle the prior probability shift by re-weighting the source samples using the similar idea as the Kernel Mean Matching method [115]. They also define the situation where both $P(Y)$ and $P(X|Y)$ are shifted across datasets and propose a kernel approach to reduce the distribution
shift by re-weighting and transforming the source data. It is assumed that the source data are able to be transferred to the target domain by location-scale (LS) transformation (i.e. $P(X|Y)$ only differs in the location and scale). Instead of assuming that all the features can be transferred to the target domain by LS transformation, Gong et al. [94] propose to learn the conditional invariant components through a linear transformation, and then the source samples are re-weighted to reduce shift of $P(Y)$ and $P(Y|X)$ between domains.

Recently, Yan et al. [300] take both the domain shift and class weight bias across domains into account. To take the class prior probability into account, they introduce class-specific weights. Specifically, the domain adaptation is performed by iteratively generating the pseudo-labels to the target samples, learning the source class weights, and tuning the deep CNN model parameters.

### 2.3.5 Sequential Labelled Target Data

In practices, the target data can be sequential video streams or continuous evolving data. The distribution of the target data may also change with time. Since the target data are labelled, this problem is named *supervised online domain adaptation*.

**Self Labelling** Xu et al. [293] assume a weak-labelling setting and propose an incremental method for object detection across domains. Specifically, the adaptation model is a weighted ensemble of the source and target classifiers and the ensemble weights are updated with time.

### 2.3.6 Sequential Unlabelled Target Data

Similar to problem in 2.3.5, the target data are sequential in this problem, however, no labelled target data is available, which is named *unsupervised online domain adaptation* and related to but different from *concept drift*. The concept of drift [79] refers to changes in the conditional distribution ($P(Y|X)$), while the marginal distribution ($P(X)$) stays unchanged, while in online domain adaptation the changes between domains are caused by the changes of the input distribution.

**Geometric Approach** Hoffman et al. [109] extend the Subspace Alignment method [74] to handle continuous evolving target domain. Both the subspaces and subspace metrics that align the two subspaces are updated after each new target sample comes. Bitarafan et al. [18] tackle the continuously evolving target domain using the idea of GFK [93] to construct linear transformation. The linear transformation is updated after a new batch of unlabelled target domain data come. Each batch of arrived target data is classified after the transformation and included in the source domain for recognizing the next batch of data.
Self Labelling  Jain and Learned-Miller [118] address the online adaptation in the face detection task by adapting pre-trained classifiers using a Gaussian process regression scheme. The intuition is that the “easy-to-detect” faces can help the detection of “hard-to-detect” faces by normalizing the co-occurring “hard-to-detect” faces and thus reducing their difficulty of detection. Xu et al. [291] propose an online domain adaptation model for multiple object tracking using a two-level hierarchical tree framework, where the leaf nodes correspond to the object detectors while the root node corresponds to the class detector. The adaptation is executed in a progressive manner.

2.3.7 Unavailable Target Data

This problem is also named domain generalization in literature, where the target domain data are not presented for adaptation. Thus, multiple source datasets are generally required to learn the dataset invariant knowledge that can be generalized to a new dataset. Note that domain generalization is distinguished from multi-source domain adaptation (MSDA)[249, 63, 110, 61, 92, 294] since MSDA generally requires the access to the target data for adaptation. We will discuss transfer learning from multiple sources in details in Section 7.2.3.

Higher-level Representation  Most of the existing work tackle this problem by learning domain invariant and compact representation from multiple source domains [19, 132, 191, 69, 242, 86, 87, 190, 157]. For example, Khosla et al. [132] explicitly model the bias of each source domain and try to estimate the weights for the unbiased data by removing the source domain biases. Muandet et al. [191] propose the Domain-Invariant Component Analysis (DICA), a kernel-based method, to learn an invariant mapping that reduces the domain shift and preserve discriminative information at the same time. Fang et al. [69] propose an unbiased metric learning approach to learn unbiased metric from multiple biased datasets. Ghifary et al. [86] propose a Multi-Task Autoencoder (MTAE) method. It substitutes artificially induced corruption in standard denoising autoencoder with some specific variations of the objects (e.g. rotation) to form multiple views. Hence, MTAE learns representations that are invariant to multiple related domains.

Ensembling classifiers learned from multiple sources is also used for generalizing to unseen target domain [297, 195, 196, 161]. Xu et al. [297] propose to reduce the domain shift in an exemplar-SVMs framework by regularizing positive samples from the same latent domain to have similar likelihoods from each exemplar classifier. Similarly, Niu et al. [195] extend this idea to the source domain samples with multi-view features. Niu et al. [196] explicitly discover the multiple hidden domains [92], and then an ensemble of classifiers is formed by learning a single classifier for each individual category in each discovered hidden domain.
2.4 Heterogeneous Feature Spaces

This section discusses the problems that $S$ and $T$ are different due to $\mathcal{R}_S \neq \mathcal{R}_T$, but $\mathcal{Y}_S = \mathcal{Y}_T$. The different feature spaces can be generated from different data modalities or different feature extraction methods. Similar to the scenario defined in Section 2.3, sufficient labelled source domain data are assumed to be available in the following subproblems.

2.4.1 Labelled Target Dataset

This problem assumes limited labelled target data are presented for adaptation. This problem is named supervised heterogeneous domain adaptation.

**Higher-level Representation** Some methods assume that only the feature spaces are different while the distributions are the same between source and target datasets. Since the labelled data in the target dataset are scarce, Zhu et al. [335] propose to use the auxiliary heterogeneous data that contain both modalities from Web to extract the semantic concept and find the shared latent semantic feature space between different modalities.

**Class-based Approach** The class-based approach has also been used to connect heterogeneous feature spaces. Finding the relationship between different feature spaces can be seen as translating between different languages. Hence, Dai et al. [51] propose a translator using a language model to translate between different data modalities or feature spaces by borrowing the class label information. Kan et al. [131] propose a multi-view discriminant analysis method that learns view-specific linear mappings for each view using label information to find a shared space that is view-invariant. Manifold alignment method [270] is also used for heterogeneous domain adaptation with the class-based approach.

Inspired by [52], the feature augmentation based method has also been proposed [62, 162] for heterogeneous domain adaptation, which transforms the data from two domains into a shared subspace, and then two data augmentation transformations are proposed such that the transformed features in the subspace are augmented with the original data as well as zeros.

Kulis et al. [141] extend [224] to learn an asymmetric mapping that transforms samples between domains using labelled data from both domains, with the similar assumption as [224] that the label spaces of target training set and target test set are non-overlapping subsets of source label space. Different from previous metric learning based domain adaptation that learns the asymmetric feature transformation between heterogeneous features [141], the asymmetric metric of classifiers can also be learned to bridge source and target classifiers on heterogeneous features [331].
Hybrid Approach  The first group of work focuses on cross-modal representation learning by combing class-based and higher level representation approaches. Gong et al. [95] propose a three-view Canonical Correlation Analysis (CCA) model that explicitly incorporates the high-level semantic information (i.e. high-level labels or topics) as a third view. A recent work[269] incorporates the adversarial learning to the supervised representation learning for cross-modal retrieval.

Another line of research assumes that both the feature spaces and the data distributions are different. Shekhar et al. [231] extend [232] to heterogeneous feature spaces, where the two projections and a latent dictionary are jointly learned to simultaneously find a common discriminative low-dimensional space and reduce the distribution shift. Similarly, Sukhija et al. [244] assume the label distributions between domains are shared. Then the shared label distributions are used as pivots to derive a sparse projection between the two domains.

2.4.2 Labelled plus Unlabelled Target Dataset

In this problem, both limited labelled and sufficient unlabelled target data are presented, which is named semi-supervised heterogeneous domain adaptation.

Statistical Approach  Tsai et al. [260] propose the Cross-Domain Landmark Selection (CDLS) method for heterogeneous domain adaptation (HDA) using the statistical approach (MMD). Specifically, the CDLS method derives a heterogeneous feature transformation which results in a domain-invariant subspace for associating the heterogeneous domains. and assigns the weight to each instance according to their adaptation ability using both labelled and unlabelled target samples.

Class-based Approach  Xiao and Guo [288] propose a kernel matching method, where a kernel matrix of the target domain is matched to a source domain sub-matrix by exploiting the label information such that the target samples are mapped to similar source samples. The unlabelled target samples are expected to be aligned with the source samples from the same class with the guides of labelled target samples via the function of kernel affinity measures between samples.

2.4.3 Unlabelled Target Dataset

This problem assumes no labelled target domain data is available. We name this problem as unsupervised heterogeneous domain adaptation. In this problem, the feature spaces could be completely different between datasets. It can also be assumed that the source data consist of multiple modalities while the target data only contain one of the modalities, or vice versa.
**Statistical Approach**  
Chen et al. [34] and Li et al. [167] assume the source datasets contain multiple modalities and target dataset only contains one modality and the distribution shift between datasets also exists. Hence, the distribution mismatch also needs to be explicitly considered. Specifically, the statistical approach (e.g. MMD) is used such that the source and target common modalities are projected to a shared subspace. In the meantime, the multiple source modalities are also transformed to the same representation in the shared space. They iteratively refine the shared space and the robust classifier.

**Correspondence Approach**  
The co-occurrence data between different feature spaces or modalities have been employed for heterogeneous domain adaptation [211, 302].

**Hybrid Approach**  
The correspondence approach or statistical approach are generally incorporated into higher-level representation approach for transferring between data modalities or feature spaces.

Canonical Correlation Analysis (CCA)[5] is a standard approach to learning two linear projections of two sets of data that are maximally correlated. Neither supervised data nor the paired data are required. Many cross-modal recognition or retrieval methods incorporate the idea of CCA[6, 72, 299] into deep models. Cross-media multiple deep networks (CMDN)[207] jointly preserve the intra-media and inter-media information and then hierarchically combine them for learning the rich cross-media correlation. Castrejn et al. [30] introduce a cross-modal representation method across RGB modality, sketch modality, clipart, and textual descriptions of indoor scenes. The cross-modal convolutional neural networks are regularized using statistical regularization so that they have a shared representation that is invariant to different modalities.

The paired correspondence data are used in [104], where a cross-modal supervision transfer method is proposed. Deep CNNs are pre-trained on the source data (e.g. a large-scale labelled RGB dataset). Then the paired target data (unlabelled RGB and depth image pairs) are used for transferring the source parameters to the target networks by constraining the paired samples from different modalities to have the similar representations.

A line of research focuses on the task of translation between different domains. For example, in machine translation between languages, the sentence pairs are presented in the form of a parallel training corpus for learning the translation system. Traditional translation system [136] is generally phrase-based, whose sub-components are usually learned separately. Differently, a newly emerging approach, named Neural machine translation [129, 250, 40, 8], constructs and trains a neural network that inputs a sentence and outputs the translated sentence.

Similarly, in the computer vision domain, image-to-image translation [117] has also been extensively exploited, which aims at converting an image from one representation of a given scene to another (e.g. texture synthesis [154], sketch to photograph [117], RGB to
depth [104], time hallucination [234, 145, 117], image to semantic labels [173, 65, 289], stimulated to real image [236], style transfer [154, 278, 84, 128, 322], and general image-to-image translation [172, 117, 306, 133, 333, 15, 156, 171]). The key idea for tackling these tasks is to learn a translation model between paired (correspondence approach) or unpaired samples (statistical approach) from different domains. The recent deep learning based techniques have greatly advanced the image-to-image translation task. For example, the deep convolutional neural networks based methods [173, 289, 65, 84, 128, 322], and the Generative Adversarial Networks (GANs [96]) based methods [278, 154, 172, 236, 117, 306, 133, 333, 15, 156, 171] have been extensively exploited for learning the translation model. Though the original purposes of some of these work on translation between domains may not be cross-dataset recognition, the ideas can be borrowed for cross-modality or cross feature spaces recognition. If a proper translation between domains can be obtained, the target task can be boosted by the translated source domain data.

2.5 Heterogeneous Label Spaces

This section discusses the problems that $\mathcal{X}_S = \mathcal{X}_T$ and $\mathcal{Y}_S \neq \mathcal{Y}_T$. For example, in the classification tasks, when the label spaces between datasets are different, there still exists shared knowledge between previous categories (e.g. horse) and new categories (e.g. zebra) that can be used for learning new categories. The source domain is assumed to be labelled except for the last sub-problem (Section 2.5.5).

2.5.1 Labelled Target Dataset

This setting is commonly used in the deep learning context. In practice, the deep networks are rarely trained from scratch (with random initialization), since the target datasets rarely have sufficient labelled data. Thus, transfer learning is generally used. The pre-trained deep models from a very large source dataset are used either as an initialization (then fine-tune the model according to the target data) or as a fixed feature extractor for the target, which is generally different from the original task (i.e. different label spaces).

The fine-tuning procedure is similar to one-shot learning or few-shot learning. The key difference is that the available target data are sufficient for the target task in fine-tuning but in few-shot learning, the target data are generally rare (e.g. only one sample per class in the extreme case). The few-shot learning also has a close connection with multi-task learning. The difference is that one-shot learning emphasizes on the recognition of the target data with limited labelled data while the objective of multi-task learning is to improve all the tasks with good training data for each task.
CHAPTER 2. LITERATURE REVIEW

Higher-level Representation Approach  Since the training of deep learning models requires a large scale dataset to avoid overfitting, the transfer learning techniques [307] can be used for small-scale target datasets. The most commonly used transfer learning technique is to initialize the weights from a pre-trained model and then the target training data are used to fine-tune the parameters for the target task. When the pre-trained source model is used as the initialization, two strategies can be employed. First is to fine-tune all the layers of the deep neural network, while the second strategy is to freeze several earlier layers and only fine-tune the later layers to reduce the effects of overfitting. This is inspired by the observation that the features extracted from the early layers show more general features (e.g. edge or color) that are transferable to different tasks. However, the later layers are gradually more specific to the details of the original source tasks. Other transfer methods [58, 219] directly use the pre-trained deep convolutional nets (normally after removing the last one or two fully connected layers) on a large dataset (e.g. ImageNet [53]) as a fixed feature extractor for the target data.

Note that when the pre-trained deep models are used as an initialization or a fixed feature extractor in the deep learning frameworks, only the pre-trained weights need to be stored without the need of storing the original large-scale source data, which is appealing.

Class-based Approach  Patricia and Caputo [205] treat the pre-trained models from multi-source domains as experts to augment the target features. The output confidence values of prior models are treated as features and the features from the target samples are augmented with these confidence values to build a target classifier.

Several classifier-based methods are proposed to transfer the parameters of classifiers using generative models [71, 146], or discriminative models [255, 7, 182, 126]. The key idea is using source models as prior knowledge to regularize the models of the target task. These methods are also called the Hypothesis Transfer Learning (HTL) since it assumes no explicit access to the source domain data and only uses source models learned from a source domain. The HTL has been theoretically analyzed [144, 280, 59]

Hybrid Approach  Recently, the deep learning based approaches have been proposed for few-shot learning, most of which are metric learning based methods. One early neural network approach to one-shot learning was provided by Siamese networks [134], which employs a structure to rank similarity between inputs. Vinyals et al. [267] propose the matching networks, where a differentiable neural attention mechanism is used over a learned embedding of the limited labelled target data. This method can be considered as a weighted nearest-neighbour classifier in an embedded space. Snell et al. [240] transform the input into an embedding space by proposing a prototypical network and the prototype from each class is taken as the mean of the embedded support set. Differently, Ravi and Larochelle [218] propose a meta-learning-based few-shot learning method, where a
meta-learner LSTM [108] model is used to produce updates for training the few-shot neural network classifier. Given a few target labelled examples, this approach can generalize well on the target set.

### 2.5.2 Unlabelled Target Dataset

Some researches also try to tackle the heterogeneous label space problem by assuming that only unlabelled target data are presented. This problem can be named as *unsupervised transfer learning*.

**Higher-level Representation**  The higher-level representation approach is generally used for this problem. Two different scenarios are considered in literature.

The first scenario assumes that only the label spaces between datasets are disjoint while the distribution shift is not considered. Since no labelled target data are available, the unseen class information is generally gained from a higher level semantic space shared between datasets. For example, some research assumes that the human-specified high-level semantic space (e.g. attributes [199], or text descriptions [220]) shared between datasets are available. Given a defined attribute or text description ontology, a vector in the semantic space can be used for representing each class. However, it is expensive to acquire the attribute annotations or text descriptions. Hence, to avoid human involved annotations, another strategy learns the semantic space by borrowing the large and unrestricted, but freely available, text corpora (e.g. Wikipedia) to derive a word vector space [76, 187, 241]. The related work on semantic space (e.g. attributes, text descriptions, or word vector) will be further discussed in Section 2.5.4, since the target data are generally not required when the semantic space is involved.

The second scenario assumes that apart from the different label spaces, the domain shift (i.e. the distribution shift of features) also exists between datasets [78, 164, 135, 271, 327, 305, 295]. This is named the *projection domain shift* problem by [78]. To reduce the domain shift explicitly, the training data (unlabelled) in the target domain are generally required to be available. For example, Fu et al. [78] introduce a multi-view embedding space in a transductive setting, such that different semantic views are aligned. Kodirov et al. [135] propose a regularised sparse representation framework that utilizes the target class prototypes estimated from target images to regularise the projections of the target data and thus overcomes the projection domain shift problem.

### 2.5.3 Sequential Labelled Target Data

This problem assume the target data are sequential and can be from different classes, which is also called *online transfer learning*, and closely related to *lifelong learning* [253, 223, 166]. Both concepts focus on the continuous learning processes for evolving tasks.
However, the online transfer learning emphasizes on improving the target domain performance (without sufficient target training data), but lifelong learning tries to improve the future target task (with sufficient target training data) as well as all the past tasks [37]. Also, the lifelong learning can be seen as incremental/online multi-task learning.

**Self Labelling**  Nater et al. [192] address an action recognition scenario where the unseen activities to be recognized only have one labelled sample per new activity. They build a multi-class model which uses the prior knowledge of seen classes and progressively learns the new classes. Then the newly labelled activities are integrated into the previous model to update the activity model. Zhao and Hoi [328] propose an ensemble learning based online transfer learning method (OTL) that learns a classifier in an online fashion using the target data, and combines it with the pre-learned source classifier. The combination weights are tuned dynamically based on the loss between the ground-truth label of the incoming sample and the current prediction. Tommasi et al. [257] then extended OTL [328] and addressed the case of online transfer learning from multiple sources.

### 2.5.4 Unavailable Target Data

This problem is also named zero-shot learning in literature, where unseen target categories are to be recognized without having access to the target data. Different from domain generalization (see Section 2.3.7), the categories of unseen target data are different from the source categories in zero-shot learning. As mentioned in Section 2.5.2, the unseen categories can be generally connected via some auxiliary information, such as a common semantic space.

**Higher-level Representation**  Most of the methods for this problem rely on the existence of a labelled source dataset of seen categories and the prior knowledge about the semantic relationship between each unseen category and the seen categories. In general, the seen and unseen categories are correlated in a high-level semantic space. Such a semantic space can be an attribute space [199], text description space [220], or a word vector space [76, 187, 241]. Since multiple semantic spaces are often complementary to each other, some methods are proposed to fuse multiple semantic spaces [1, 321].

The attribute space is the most commonly used intermediate semantic space. The attributes are defined as properties observable in images, which are described with human-designated names such as “white”, “hairy”, “four-legged”. Hence, in addition to label annotation, the attribute annotations are required for each class. However, the attributes are annotated per-class rather than per-image. Thus, the effort to annotate a new category is small. Two main strategies are proposed for recognizing unseen object categories using attributes. The first is recognition using independent attributes, consists of learning an
independent classifier per attribute [147, 199, 143, 169, 204]. At test time, the attribute values for test data are predicted using the independent classifiers and the labels are then inferred. Since attribute detectors are expected to generalize well on both seen and unseen categories, some research is devoted to discovering discriminant attributes [217, 32, 213], or modelling the uncertainty of attributes [279, 120], or robustly detecting attributes from images [80, 24]. However, Akata et al. [2] argue that the attribute classifiers in previous works are learned independently of the end-task, and thus they may be able to predict the attributes from new images but may not be able to effectively infer the classes. Hence, the second strategy is recognition by assuming a fixed transformation between the attributes and the class labels [1, 221, 326, 325, 3, 212, 287, 165] to learn all attributes simultaneously. To sum up, the attribute-based zero-shot learning methods are promising for recognizing unseen classes, while with a key drawback that the attribute annotations are still required for each class. Instead of using attributes, the second semantic space is image text descriptions [220], which provides a natural language interface. However, similar to attribute space, the expensive manual annotation is required for obtaining the good performance. The third semantic space is the word vector space [76, 187, 241, 153], which is derived from a huge text corpus and generally learned by a deep neural network. The word vector space is attractive since extensive annotations are not required for obtaining the semantic space.

2.5.5 Unlabelled Source Dataset

This problem assumes that the source data are unlabelled but the contained information (e.g. basic visual patterns) can be used for target tasks, which is known as self-taught learning.

Higher-level Representation Raina et al. [215] firstly presented the idea of “self-taught learning”. They learn the sparse coding from the source data to extract higher-level features. Some variations of Raina et al. [215]’s method are proposed either by generalizing the Gaussian sparse coding to exponential family sparse coding [151] , or by taking the supervision information contained in labelled images into consideration [273]. Moreover, Kumagai [142] provide a theoretical analysis for self-taught learning with the focus on discussing the learning bound of sparsity-based methods.

The idea of self-taught learning has also been used in deep learning framework, where the unlabelled data are used for pre-training the network to obtain good starting point of parameters [149, 81, 140]. For instance, Gan et al. [81] use the unlabelled samples to pre-train the first layer of Convolutional deep belief network (CDBN) for initializing the network parameters. Kuen et al. [140] extract the domain-invariant features from unlabelled source image patches for the tracking tasks using stacked convolutional autoencoders.
2.6 Heterogeneous Feature Spaces and Label Spaces

In this section, a more challenging scenario is discussed, where $\mathcal{X}_S \neq \mathcal{X}_T$ and $\mathcal{Y}_S \neq \mathcal{Y}_T$. There is little work regarding this scenario due to the challenges and the common assumption that sufficient source domain labelled data is available.

2.6.1 Labelled Target Dataset

This problem assumes the labelled target data are available. We name this problem as heterogeneous supervised transfer learning.

Higher-level Representation  Rather than assuming completely different feature spaces, most methods in this setting assume that the source domain contains data with multi-modality but the target domain only has one of the source domain modalities. Ding et al. [55] propose to uncover the missing target modality by finding similar data from the source domain, where a latent factor is incorporated to uncover the missing modality based on the low-rank criterion. Similarly, Jia et al. [122] propose to transfer the knowledge of RGB-D (RGB and depth) data to the dataset that only has RGB data. They applied the latent low-rank tensor method to discover the common subspace of the two datasets.

Hybrid Approach  Hu and Yang [112] assume the feature spaces, the label spaces, as well as the underlying distributions are all different between source and target datasets and propose to transfer the knowledge between different activity recognition tasks by learning a mapping between different sensors. They adopt the similar idea of translated learning [51] to find a translator between different feature spaces using statistical approach (e.g. JS divergence). Then the Web knowledge is used to link the different label spaces using self-labelling.

2.6.2 Sequential Labelled Target Data

This problem assumes the sequential target data have different feature space with source data, which is named as heterogeneous online transfer learning.

Self Labelling  As mentioned in Section 2.5.3, Zhao and Hoi [328] propose the OTL method for online transfer learning. They also consider the case of heterogeneous feature spaces by assuming the source domain feature space to be a subspace of the target domain feature space. Then a multi-view approach is proposed by adopting a co-regularization principle of online learning of two target classifiers simultaneously from the two views (the source domain feature space and the new space). The unseen target example is classified by the combination of the two target classifiers.
2.7 Datasets

Table 2.1 lists the commonly used visual datasets for transfer learning. They are categorized into object recognition, Hand-Written digit recognition, face recognition, person re-identification, scene categorization, action recognition and video event detection. In the table, the ✓ indicates the dataset has been evaluated on the corresponding problem while the # indicates that the datasets have the potential to be used in the evaluation of the algorithms for the problem though reported results are not publicly available to our knowledge.

Sample images of the datasets that are evaluated in this thesis are illustrated to highlight the inherent challenges in transfer learning, especially visual recognition across different domains (which is the focus of this thesis).

2.7.1 Object Recognition

There are several object datasets designed for domain adaptation tasks (e.g. Office (31 classes, 3 domains) [224], Office+Caltech (10 classes, 4 domains) [93], VLCS (5 classes, 4 domains) [132], ImageCLEF (12 classes, 4 domains) [28], cross-dataset testbed (dense: 40 classes, 4 domains; sparse: 105 classes, 12 domains) [256], Office-Home (65 classes, 4 domains) [265], PACS (7 classes, 4 domains) [157]). These datasets are consisted of data from different domains but from the same modality and for the same task (i.e. same label space between domains). These datasets were used for evaluating cross-domain recognition methods. They are illustrated in Figure 2.2, 2.3, 2.4, 2.5. Though the data modalities between domains are the same, different feature extractors can results in different data dimensions, which are also used for heterogeneous transfer in the previous research.

The datasets (e.g. NUS-WIDE (81 classes, 5,018 unique tags)[43], Wikipedia (10 classes, median text length: 200 words) [208], Pascal Sentence (20 classes, 5 sentence descriptions per image) [216], MSCOCO (91 classes, 5 sentence descriptions per image) [168]) contain multiple data modalities (e.g. images, text, etc.) can be used for evaluating
CHAPTER 2. LITERATURE REVIEW

Figure 2.3: Sample images of class “dog” in VLCS dataset.

Figure 2.4: Sample images of class “aeroplane” in ImageCLEF dataset.

Figure 2.5: Sample images of class “horse” in PACS dataset.

cross-modal transfer methods.

Several datasets (e.g. a-Pascal & a-Yahoo (aP&Y) (32 classes, 65 attributes)[70], Animals with Attributes (AwA) (50 classes, 85 attributes)[147], Caltech-UCSD CUB-200-2011 Birds (CUB-200-2011) (200 classes, 312 attributes)[268]) contain images annotated with not only class labels but also visual attributes to evaluate on transferring between different categories (i.e. heterogeneous label spaces).

2.7.2 Hand-Written Digit and Character Recognition

Compared to object transfer learning datasets, the Hand-Written Digit datasets (e.g. MNIST[150], USPS[116], SVHN[193], SYN DIGITS[82]) are generally large scale in terms of the
number of samples. The combinations of these datasets form pairs of source and target domains, which are commonly used for evaluating deep learning-based transfer methods.

The MNIST and USPS datasets were used in this thesis and sample images are shown in Figure 2.6. It can be seen that the writing style of the two datasets are very different.

![MNIST USPS](image)

**Figure 2.6**: Sample images of MNIST and USPS digit datasets.

The Omniglot character dataset (1623 characters from 50 different alphabets)[146] is a commonly used in one-shot learning due to its large number of classes and few data per class.

The dataset pairs with tasks (e.g. Handwritten digits v.s. Handwritten English characters[215], Handwritten English characters v.s. Font characters[215]) have been evaluated for self-taught learning.

### 2.7.3 Face Recognition

The dataset shift in face recognition can be caused by poses, resolution, illuminations, and expressions (e.g. CMU Multi-PIE (337 subjects, across 15 poses, 20 illuminations, 6 expressions and 4 different sessions) [103]). Another type of dataset shift is caused by different modalities (e.g. Oulu-CASIA NIR&VIS facial expression (80 subjects) databasea, BUAA-VisNir Face Database (150 subjects) b, CUHK Face Sketch (80 subjects) [277], CASIA NIR-VIS 2.0 (725 subjects)[158], ePRIP VIS-Sketch (123 subjects)[188]). Both types of datasets have been extensively used for evaluating transfer learning algorithms.

### 2.7.4 Person Re-identification

The person re-identification requires transfer learning to identify person from different viewpoints and environments. The commonly used datasets include VIPeR dataset (632 persons, 2 views) [100], CUHK02 Person Re-identification dataset (1,816 persons in total from five pairs of views which have 971, 306, 107, 193 and 239 persons respectively) [160], PRID dataset (385 persons viewed from camera A, 749 persons viewed

---

*a*http://www.ee.oulu.fi/~gyzhao/

*b*http://irip.buaa.edu.cn/Research/Research_Highlights.htm
from camera B, 200 persons appear in both cameras) [107], iLIDS dataset (119 subjects, 3 views) [330], CAVIAR (72 subjects, 2 views) [38], and 3DPeS (192 subjects, 8 views)[11].

2.7.5 Scene Categorization

The application of Scene Categorization includes transferring between categories (e.g. SUN Attribute (717 classes, 102 attributes)[206]) and modalities (e.g. CMPlaces[30] that contains RGB, sketches, clipart, and textual descriptions of indoor scenes with 205 categories). Besides, the Scene over time (2 classes: busy or empty) dataset[109] can be used for evaluating the online domain adaptation methods, since the dataset contains traffic scene images that change over time.

2.7.6 Action Recognition

Multiple action recognition datasets with shared (e.g. KTH → MSR Action[26]) or un-shared actions (e.g. UCF YouTube (11 actions) [170] v.s. HMDB51 (7 actions) [139] used in [332], KTH (6 actions)[227] v.s. UCF YouTube (11 actions)[170] used in [182], HumanEva (5 actions)[238] v.s. KSA (5 actions)[182] used in [182], A combination of KTH, Weizmann, UIUC[169], MSR pair action 3D (10 actions)[197] → MSR daily (16 activities)[274] used in [122]) are used for the cross-dataset action recognition. Different RGB-D action datasets were used in this thesis for evaluating the domain adaptation algorithms. Sample video frames were shown in Figure 2.7 to illustrate the challenges due to the differences in background, performing style, and subject sizes among datasets.

![Sample video frames](image)

Figure 2.7: Sample video frames of depth map of RGB-D-based action datasets.

Another scenario of transfer of actions is cross-view action recognition. The most commonly used multi-view action dataset is IXMAS dataset (11 actions, 5 views)[282]. Recently, some RGB-D based multi-view action datasets (Northwestern-UCLA Multiview Action3D (N-UCLA) (10 actions, 3 views) [272], ACT4^2 dataset (14 actions, 4 views) [39]) are also proposed after the release of Kinect sensors. The RGB-D-based datasets can also evaluate cross-modality recognition. A recently released large scale
combined RGB-D action dataset [316] consists of multiple individual datasets, which can be evaluated on cross-dataset set-up. More complete review of RGB-D-based action datasets can be referred to [318].

The action attributes are also exploited by manually annotating on existing action datasets for evaluating cross-category transfer learning. For example, a combination of KTH [148], Weizmann [20], and UIUC [259] with 21 actions are evaluated for zero-shot learning in [169] by manually defining 34 action attributes.

### 2.7.7 Video Event Detection

Video event detection or video concept detection is to automatically classify video shots into a certain semantic event (such as making a cake, wedding ceremony, and changing a vehicle tire) or concept (such as meeting, sports, and entertainment). Most commonly used data for evaluating video event detection are from TREC Video Retrieval Evaluation (TRECVID) dataset series\(^c\). The transfer learning is used for cross-task video event detection by previous research. For example, the TRECVID 2005 (13 programs, 39 concepts) [239] dataset is used by Yang et al. [301], where one of the 39 concepts is picked as the target concept and one of the 13 programs is the target program. Differently, the TRECVID 2010 (3 events) and TRECVID 2011 (15 events) are used together by Ma et al. [183] for the cross-domain event detection, where the TRECVID 2011 semantic indexing task (SIN11) is used as the source auxiliary dataset.

The transferring from image to video modalities\(^c\) is also evaluated on the event detection datasets (e.g. ImageNet [53]→TRECVID 2011 (15 complex events)[198], and ImageNet[53]→LabelMe Video (5 objects are selected)[308]).

\(^c\)http://www-nlpir.nist.gov/projects/trecvid/
Table 2.1: Suitability of the widely used datasets where the ✓ indicates the dataset has been used the corresponding problems while the # indicates the datasets can be potentially used for the problem. Problem notations: P3.1, supervised domain adaptation (DA); P3.2, Semi-supervised DA; P3.3, Unsupervised DA; P3.4, Supervised online DA; P3.5, Supervised online DA; P3.6, Unsupervised online DA; P3.7, Domain generalization; P4.1, Supervised Heterogeneous DA; P4.2, Semi-supervised Heterogeneous DA; P4.3, Unsupervised Heterogeneous DA; P5.1, Few-shot Learning; P5.2, Unsupervised transfer learning (TL); P5.3, Online TL; P5.4, Zero-shot Learning; P5.5, Self-taught Learning; P6.1, Heterogeneous TL; P6.2, Heterogeneous online TL.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>P3.1</th>
<th>P3.2</th>
<th>P3.3</th>
<th>P3.4</th>
<th>P3.5</th>
<th>P3.6</th>
<th>P3.7</th>
<th>P4.1</th>
<th>P4.2</th>
<th>P4.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office[224]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Office+Caltech[93]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cross-dataset testbed[256]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Office-Home[265]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>VLS[132]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ImageCLEF-DA[28]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PACS[157]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CIFAR-10 v.s. STL-10[75]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RGB-D→Caltech[256]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Syn Signs v.s. GTSRB[82]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NUS-WIDE[43]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>aP&amp;Y[70]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AwA[147]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Caltech-UCSD CUB[268]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Car over time[109]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>STL-10 dataset[44]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LabelMe→NUS-WIDE[275]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Outdoor scene v.s. Caltech[101][215]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MNIST v.s. MNIST-M[82]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MNIST v.s. SVHN[82]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>USPS v.s. SVHN[82]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SYN DIGITS v.s. SVHN[82]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Omniglot[146]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ePRIP VIS-Sketch[188]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>English characters v.s. Font characters[215]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CMU Multi-PIE[101]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CMU-PIE v.s. Yale B[55]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Oulu-CASIA NIR&amp;VIS v.s. BUAA-VisNir[55]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CASIA NIR-VIS 2.0[158]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ePRIP VIS-Sketch[188]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>VIPFace[101]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>VIPFace[101]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ILIDS[330]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CAVAR[38]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3DPeS[11]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CMPlaces[30]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Scene over time[109]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NYUv2[104]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>UCF YouTube v.s. HMDB51[352]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>KTH v.s. UCF YouTube[182]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>KTH v.s. CarsMedia[182]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>KTH→MSR Action[25]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HumanEva v.s. KSA[182]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>A combination of KTH, Weizmann, UIUC[169]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Multiview IXMAS dataset[282]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>N-UCLA Multiview Action3D[272]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ACT4 dataset[39, 195]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MSR pair action 3D→MSR daily[122]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Transferring Activities[192]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TRECVID 2005[301]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TRECVID 2010&amp;2011[183]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TRECVID MED[3, 296]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ImageNet→TRECVID 2011[252]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ImageNet→LabelMe Video[252]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Chapter 3

Joint Geometrical and Statistical Alignment for Visual Domain Adaptation

3.1 Motivation

The most commonly used domain adaptation approaches include instance-based adaptation, feature representation adaptation, and classifier-based adaptation [200, 228]. In unsupervised domain adaptation, as there is no labeled data in the target domain, the classifier-based adaptation is not straightforward. Alternatively, we can deal with this problem by minimizing distribution divergence between domains as well as the empirical source error [13]. It is generally assumed that the distribution divergence can be compensated either by an instance based adaptation method, such as reweighting samples in the source domain to better match the target domain distribution, or by a feature transformation based method that projects features of two domains into another subspace with small distribution shift. The instance-based approach requires the strict assumptions [200, 185] that 1) the conditional distributions of the source and target domain are identical, and 2) a certain portion of the data in the source domain can be reused for learning in the target domain through reweighting. While the feature transformation based approach relaxes these assumptions, and only assumes that there exists a common space where the distributions of the two domains are similar. This chapter follows the feature transformation based approach.

Two main categories of feature transformation methods are identified [303] in the literature, namely data-centric methods and subspace centric methods. The data-centric methods seek a unified transformation that projects data from two domains into a domain invariant space to reduce the distributional divergence between domains while preserving data properties in original spaces, such as [201, 177, 178, 87]. The data-centric methods
only exploit shared feature in two domains, which will fail when the two different domains have a large discrepancy, because there may not exist such a common space where the distributions of two domains are the same and the data properties are also maximally preserved in the meantime. For the subspace centric methods, the domain shift is reduced by manipulating the subspaces of the two domains such that the subspace of each individual domain all contributes to the final mapping [93, 74, 73]. Hence, the domain-specific features are exploited. For example, Gong et al. [93] regard two subspaces as two points on Grassmann manifold, and find points on a geodesic path between them as a bridge between the source and target subspaces. Fernando et al. [74] align source and target subspaces directly using a linear transformation matrix. However, the subspace centric methods only manipulate on the subspaces of the two domains without explicitly considering the distribution shift between projected data of two domains. The limitations of both data-centric and subspace centric methods will be illustrated on a synthetic dataset in Section 3.3.1.

In this chapter, we propose a unified framework that reduces the distributional and geometrical divergence between domains simultaneously by exploiting both the shared and domain-specific features. Specifically, we learn two coupled projections to map the source and target data into respective subspaces. After the projections, 1) the variance of target domain data is maximized to preserve the target domain data properties, 2) the discriminative information of source data is preserved to effectively transfer the class information, 3) both the marginal and conditional distribution divergences between source and target domains are minimized to reduce the domain shift statistically, and 4) the divergence of two projections is constrained to be small to reduce domain shift geometrically.

Hence, different from data-centric based methods, we do not require the strong assumption that a unified transformation can reduce the distribution shift while preserving the data properties. Different from subspace centric based methods, we not only reduce the shift of subspace geometries but also reduce the distribution shifts of two domains. In addition, our method can be easily extended to a kernelized version to deal with the situations where the shift between domains is nonlinear. The objective function can be solved efficiently in a closed form. The proposed method has been verified through comprehensive experiments on a synthetic dataset and three different real-world cross-domain visual recognition tasks: object recognition (Office, Caltech-256), hand-written digit recognition (USPS, MNIST), and RGB-D-based action recognition (MSRAction3DExt, G3D, UTD-MHAD, and MAD).

3.2 Joint Geometrical and Statistical Alignment

This section presents the Joint Geometrical and Statistical Alignment (JGSA) method in detail.
3.2.1 Problem Definition

We begin with the definitions of terminologies. The source domain data denoted as $X_s \in \mathbb{R}^{D \times n_s}$ are drawn from distribution $P_s(X_s)$ and the target domain data denoted as $X_t \in \mathbb{R}^{D \times n_t}$ are drawn from distribution $P_t(X_t)$, where $D$ is the dimension of the data instance, $n_s$ and $n_t$ are number of samples in source and target domain respectively. We focus on the unsupervised domain adaptation problem. In unsupervised domain adaptation, there are sufficient labeled source domain data, $D_s = \{(x_i, y_i)\}_{i=1}^{n_s}$, $x_i \in \mathbb{R}^D$, and unlabeled target domain data, $D_t = \{(x_j)\}_{j=1}^{n_t}$, $x_j \in \mathbb{R}^D$, in the training stage. We assume the feature spaces and label spaces between domains are the same: $X_s = X_t$ and $Y_s = Y_t$. Due to the dataset shift, $P_s(X_s) \neq P_t(X_t)$. Different from previous domain adaptation methods, we do not assume that there exists a unified transformation $\phi(\cdot)$ such that $P_s(\phi(X_s)) = P_t(\phi(X_t))$ and $P_s(Y_s|\phi(X_s)) = P_t(Y_t|\phi(X_s))$, since this assumption becomes invalid when the dataset shift is large.

3.2.2 Formulation

To address limitations of both data-centric and subspace centric methods, the proposed framework (JGSA) reduces the domain divergence both statistically and geometrically by exploiting both shared and domain-specific features of two domains. The JGSA is formulated by finding two coupled projections (A for source domain, and B for target domain) to obtain new representations of respective domains, such that 1) the variance of the target domain is maximized, 2) the discriminative information of source domain is preserved, 3) the divergence of source and target distributions is small, and 4) the divergence between source and target subspaces is small.

**Target Variance Maximization**

To avoid projecting features into irrelevant dimensions, we encourage the variances of target domain is maximized in the respective subspaces. Hence, the variance maximization can be achieved as follows

$$\max_B Tr(B^T S_t B)$$

where

$$S_t = X_t H_t X_t^T$$

is the target domain scatter matrix, $H_t = I_t - \frac{1}{n_t} 1_t 1_t^T$ is the centering matrix, $1_t \in \mathbb{R}^{n_t}$ is the column vector with all ones.
Source Discriminative Information Preservation

Since the labels in the source domain are available, we can employ the label information to constrain the new representation of source domain data to be discriminative.

\[
\max_A Tr(A^T S_b A) \tag{3.3}
\]

\[
\min_A Tr(A^T S_w A) \tag{3.4}
\]

where \(S_w\) is the within class scatter matrix, and \(S_b\) is the between class scatter matrix of the source domain data, which are defined as follows,

\[
S_w = \sum_{c=1}^C X_s^{(c)} H_s^{(c)} (X_s^{(c)})^T \tag{3.5}
\]

\[
S_b = \sum_{c=1}^C n_s^{(c)} (m_s^{(c)} - \bar{m}_s) (m_s^{(c)} - \bar{m}_s)^T \tag{3.6}
\]

where \(X_s^{(c)} \in \mathbb{R}^{D \times n_s^{(c)}}\) is the set of source samples belonging to class \(c\), \(m_s^{(c)} = \frac{1}{n_s^{(c)}} \sum_{i=1}^{n_s^{(c)}} x_i^{(c)}\), \(\bar{m}_s = \frac{1}{n_s} \sum_{i=1}^{n_s} x_i\), \(H_s^{(c)} = I_s^{(c)} - \frac{1}{n_s^{(c)}} 1_s^{(c)} (1_s^{(c)})^T\) is the centering matrix of data within class \(c\), \(I_s^{(c)} \in \mathbb{R}^{n_s^{(c)} \times n_s^{(c)}}\) is the identity matrix, \(1_s \in \mathbb{R}^{n_s}\) is the column vector with all ones, \(n_s^{(c)}\) is the number of source samples in class \(c\).

Distribution Divergence Minimization

We employ the MMD criteria \[101, 201, 177\] to compare the distributions between domains, which computes the distance between the sample means of the source and target data in the k-dimensional embeddings,

\[
\min_{A,B} \frac{1}{n_s} \sum_{x_i \in X_s} A^T x_i - \frac{1}{n_t} \sum_{x_j \in X_t} B^T x_j \|_F^2 \tag{3.7}
\]

Long et al. \[177\] has been proposed to utilize target pseudo labels predicted by source domain classifiers for representing the class-conditional data distributions in the target domain. Then the pseudo labels of target domain are iteratively refined to reduce the difference in conditional distributions between two domains further. We follow their idea to minimize the conditional distribution shift between domains,

\[
\min_{A,B} \frac{1}{n_s} \sum_{c=1}^C \frac{1}{n_s^{(c)}} \sum_{x_i \in X_s^{(c)}} A^T x_i - \frac{1}{n_t^{(c)}} \sum_{x_j \in X_t^{(c)}} B^T x_j \|_F^2 \tag{3.8}
\]
Hence, by combining the marginal and conditional distribution shift minimization terms, the final distribution divergence minimization term can be rewritten as

$$\min_{A,B} \text{Tr} \left( [A^T \quad B^T] \begin{bmatrix} M_s & M_{st} \\ M_{ts} & M_t \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \right)$$

(3.9)

where

$$M_s = X_s(L_s + \sum_{c=1}^{C} L_s^{(c)})X_s^T, \quad L_s = \frac{1}{n_s^2} 1_s 1_s^T,$$

$$\begin{pmatrix} L_s^{(c)} \end{pmatrix}_{ij} = \begin{cases} \frac{1}{(n_s^{(c)})^2} & x_i, x_j \in X_s^{(c)} \\ 0 & \text{otherwise} \end{cases}$$

(3.10)

$$M_t = X_t(L_t + \sum_{c=1}^{C} L_t^{(c)})X_t^T, \quad L_t = \frac{1}{n_t^2} 1_t 1_t^T,$$

$$\begin{pmatrix} L_t^{(c)} \end{pmatrix}_{ij} = \begin{cases} \frac{1}{(n_t^{(c)})^2} & x_i, x_j \in X_t^{(c)} \\ 0 & \text{otherwise} \end{cases}$$

(3.11)

$$M_{st} = X_s(L_{st} + \sum_{c=1}^{C} L_{st}^{(c)})X_s^T, \quad L_{st} = -\frac{1}{n_s n_t} 1_s 1_t^T,$$

$$\begin{pmatrix} L_{st}^{(c)} \end{pmatrix}_{ij} = \begin{cases} -\frac{1}{n_s^{(c)} n_t^{(c)}} & x_i \in X_s^{(c)}, x_j \in X_t^{(c)} \\ 0 & \text{otherwise} \end{cases}$$

(3.12)

$$M_{ts} = X_t(L_{ts} + \sum_{c=1}^{C} L_{ts}^{(c)})X_t^T, \quad L_{ts} = -\frac{1}{n_s n_t} 1_t 1_s^T,$$

$$\begin{pmatrix} L_{ts}^{(c)} \end{pmatrix}_{ij} = \begin{cases} -\frac{1}{n_s^{(c)} n_t^{(c)}} & x_j \in X_t^{(c)}, x_i \in X_s^{(c)} \\ 0 & \text{otherwise} \end{cases}$$

(3.13)

Note that this is different from TCA and JDA, because we do not use a unified subspace because there may not exist such a common subspace where the distributions of two domains are also similar.

**Subspace Divergence Minimization**

Similar to SA [74], we also reduce the discrepancy between domains by moving closer the source and target subspaces. As mentioned, an additional transformation matrix $M$ is required to map the source subspace to the target subspace in SA. However, we do not learn an additional matrix to map the two subspaces. Rather, we optimize $A$ and
B simultaneously, such that the source class information and the target variance can be preserved, and the two subspaces move closer in the meantime. We use the following term to move the two subspaces close:

\[
\min_{A,B} \|A - B\|_F^2
\]  

(3.14)

By using term (3.14) together with (3.9), both shared and domain-specific features are exploited such that the two domains are well aligned geometrically and statistically.

**Overall Objective Function**

We formulate the JGSA method by incorporating the above five quantities ((3.1), (3.3), (3.4), (3.9), and (3.14)) as follows:

\[
\max_{\mu, \beta, \lambda} \{\text{Target Var.}\} + \beta \{\text{Between Class Var.}\} \\
\{\text{Distribution shift}\} + \lambda \{\text{Subspace shift}\} + \beta \{\text{Within Class Var.}\}
\]

where \(\lambda, \mu, \beta\) are trade-off parameters to balance the importance of each quantity, and Var. indicates variance.

We follow [87] to further impose the constraint that \(\text{Tr}(B^T B)\) is small to control the scale of \(B\). Specifically, we aim at finding two coupled projections \(A\) and \(B\) by solving the following optimization function,

\[
\max_{A,B} \frac{\text{Tr} \left( \begin{bmatrix} A^T & B^T \end{bmatrix} \begin{bmatrix} \beta S_b & 0 \\ 0 & \mu S_w \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \right)}{\text{Tr} \left( \begin{bmatrix} A^T & B^T \end{bmatrix} \begin{bmatrix} M_t + \lambda I + \beta S_w & M_s - \lambda I \\ M_s - \lambda I & M_t + (\lambda + \mu) I \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \right)}
\]  

(3.15)

where \(I \in \mathbb{R}^{d \times d}\) is the identity matrix.

Minimizing the denominator of (3.15) encourages small marginal and conditional distributions shifts, and small within class variance in the source domain. Maximizing the numerator of (3.15) encourages large target domain variance, and large between class variance in the source domain. Similar to JDA, we also iteratively update the pseudo labels of target domain data using the learned transformations to improve the labelling quality until convergence.
### 3.2.3 Optimization

To optimize (3.15), we rewrite $[A^T \ B^T]$ as $W^T$. Then the objective function and corresponding constraints can be rewritten as:

$$
\max_{W} \frac{Tr\left(W^T \begin{bmatrix} \beta S_b & 0 \\ 0 & \mu S_t \end{bmatrix} W\right)}{Tr\left(W^T \begin{bmatrix} M_s + \lambda I + \beta S_w & M_{st} - \lambda I \\ M_{ts} - \lambda I & M_t + (\lambda + \mu)I \end{bmatrix} W\right)} \quad (3.16)
$$

Note that the objective function is invariant to rescaling of $W$. Therefore, we rewrite objective function (3.16) as

$$
\max_{W} \frac{Tr\left(W^T \begin{bmatrix} \beta S_b & 0 \\ 0 & \mu S_t \end{bmatrix} W\right)}{Tr\left(W^T \begin{bmatrix} M_s + \lambda I + \beta S_w & M_{st} - \lambda I \\ M_{ts} - \lambda I & M_t + (\lambda + \mu)I \end{bmatrix} W\right)} = 1 \quad (3.17)
$$

The Lagrange function of (3.17) is

$$
L = Tr\left(W^T \begin{bmatrix} \beta S_b & 0 \\ 0 & \mu S_t \end{bmatrix} W\right) + Tr\left(W^T \begin{bmatrix} M_s + \lambda I + \beta S_w & M_{st} - \lambda I \\ M_{ts} - \lambda I & M_t + (\lambda + \mu)I \end{bmatrix} W - I \right) \Phi \quad (3.18)
$$

By setting the derivative $\frac{\partial L}{\partial W} = 0$, we get:

$$
\begin{bmatrix} \beta S_b & 0 \\ 0 & \mu S_t \end{bmatrix} W = \begin{bmatrix} M_s + \lambda I + \beta S_w & M_{st} - \lambda I \\ M_{ts} - \lambda I & M_t + (\lambda + \mu)I \end{bmatrix} W \Phi \quad (3.19)
$$

where $\Phi = \text{diag}(\lambda_1, \ldots, \lambda_k)$ are the k leading eigenvalues and $W = [W_1, \ldots, W_k]$ contains the corresponding eigenvectors, which can be solved analytically through generalized eigenvalue decomposition. Once the transformation matrix $W$ is obtained, the subspaces $A$ and $B$ can be obtained easily. The pseudo code of JGSA is summarized in Algorithm 1.

### 3.2.4 Kernelization Analysis

The JGSA method can be extended to nonlinear problems in a Reproducing Kernel Hilbert Space (RKHS) using some kernel functions $\phi$. We use the Representer Theorem $P = \Phi(X)A$ and $Q = \Phi(X)B$ to kernelize our method, where $X = [X_s, X_t]$ denotes all the source and target training samples, $\Phi(X) = [\phi(x_1), \ldots, \phi(x_n)]$ and $n$ is the number of all samples. Hence, the objective function becomes,
**Algorithm 1:** Joint Geometrical and Statistical Alignment

**Input**: Data and source labels: \(X_s, X_t, Y_t\); Parameters: \(\lambda = 1, \mu = 1, k, T, \beta\).

**Output**: Transformation matrices: \(A\) and \(B\); Embeddings: \(Z_s, Z_t\); Adaptive classifier: \(f\).

Construct \(S_s, S_t, B_s, M_s, M_t, M_{st},\) and \(M_{ts}\) according to (3.2), (3.3), (3.4), (3.10), (3.11), (3.12), and (3.13); Initialize pseudo labels in target domain \(\hat{Y}_t\) using a classifier trained on original source domain data;

repeat

- Solve the generalized eigendecomposition problem in Equation (3.19) and select the \(k\) corresponding eigenvectors of \(k\) leading eigenvalues as the transformation \(W\), and obtain subspaces \(A\) and \(B\);
- Map the original data to respective subspaces to get the embeddings: \(Z_s = A^TX_s\), \(Z_t = B^TX_t\);
- Train a classifier \(f\) on \(\{Z_s, Y_s\}\) to update pseudo labels in target domain \(\hat{Y}_t = f(Z_t)\);
- Update \(M_s, M_t, M_{st}\), and \(M_{ts}\) according to (3.10), (3.11), (3.12), and (3.13).

until Convergence;

Obtain the final adaptive classifier \(f\) on \(\{Z_s, Y_s\}\).

\[
\max_{P, Q} \frac{Tr\left(\begin{bmatrix} P^T & Q^T \end{bmatrix} \begin{bmatrix} \beta S_b & 0 \\ 0 & \mu S_s \end{bmatrix} \begin{bmatrix} P \\ Q \end{bmatrix} \right)}{Tr\left(\begin{bmatrix} P^T & Q^T \end{bmatrix} \begin{bmatrix} M_s + \lambda I + \beta S_w & M_{st} - \lambda I \\ M_{ts} - \lambda I & M_t + (\lambda + \mu) I \end{bmatrix} \begin{bmatrix} P \\ Q \end{bmatrix} \right)}
\]  
(3.20)

where all the \(X_t\)'s are replaced by \(\Phi(X_t)\) and all the \(X_s\)'s are replaced by \(\Phi(X_s)\) in \(S_s, S_w, S_b, M_s, M_t, M_{st},\) and \(M_{ts}\) in the kernelized version.

We replace \(P\) and \(Q\) with \(\Phi(X)A\) and \(\Phi(X)B\) and obtain the objective function as follows,

\[
\max_{A, B} \frac{Tr\left(\begin{bmatrix} A^T & B^T \end{bmatrix} \begin{bmatrix} \beta S_b & 0 \\ 0 & \mu S_s \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \right)}{Tr\left(\begin{bmatrix} A^T & B^T \end{bmatrix} \begin{bmatrix} M_s + \lambda K + \beta S_w & M_{st} - \lambda K \\ M_{ts} - \lambda K & M_t + (\lambda + \mu) K \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \right)}
\]  
(3.21)

where \(S_t = \bar{K}_tK_t^T, S_w = K_tH_t^{(c)}K_t^T\), with \(K = \Phi(X)^T\Phi(X), K_s = \Phi(X)^T\Phi(X_s), K_t = \Phi(X)^T\Phi(X_t), K_{ts} = K - I_kK_s1_n + 1_nK_{ts}\), and \(1_n \in \mathbb{R}^{n \times n}\) are matrices with all \(1\). In \(S_w, m_s^{(c)} = \frac{1}{n_s} \sum_{i=1}^{n_s} k_i^{(c)}, \bar{m}_s = \frac{1}{n_s} \sum_{i=1}^{n_s} k_i\), with \(k_i = \Phi(x_i)^T\Phi(x_i)\). In MMD terms, \(M_s = K_s(L_s + \sum_{c=1}^{C} L_s^{(c)})K_t^T, M_t = K_t(L_t + \sum_{c=1}^{C} L_t^{(c)})K_t^T, M_{st} = K_s(L_{st} + \sum_{c=1}^{C} L_{st}^{(c)})K_t^T, M_{ts} = K_t(L_{ts} + \sum_{c=1}^{C} L_{ts}^{(c)})K_t^T\). Once the kernelized objective function (3.21) is obtained, we can simply solve it in the same way as the original objective function to compute \(A\) and \(B\).
3.3 Experiments

In this section, we first conduct experiments on a synthetic dataset to verify the effectiveness of the JGSA methods. Then we evaluate our method for cross-domain object recognition, cross-domain digit recognition, and cross-dataset RGB-D-based action recognition. The codes are available online\(^a\). We compare our method with several state-of-the-art methods: subspace alignment (SA) [74], subspace distribution alignment (SDA) [247], geodesic flow kernel (GFK) [93], transfer component analysis (TCA) [201], joint distribution analysis (JDA) [177], transfer joint matching (TJM) [178], scatter component analysis (SCA) [87], optimal transport (OTGL) [45], and kernel manifold alignment (KEMA) [261]. We use the parameters recommended by the original papers for all the baseline methods. For JGSA, we fix \( \lambda = 1, \mu = 1 \) in all the experiments, such that the distribution shift, subspace shift, and target variance are treated as equally important. We empirically verified that the fixed parameters can obtain promising results on different types of tasks. Hence, the subspace dimension \( k \), number of iteration \( T \), and regularization parameter \( \beta \) are free parameters.

3.3.1 Synthetic Data

Here, we aim to synthesize samples of data to demonstrate that our method can keep the domain structures as well as reduce the domain shift. The synthesized source and target domain samples are both drawn from a mixture of three RBFian distributions. Each RBFian distribution represents one class. The global means, as well as the means of the third class are shifted between domains. The original data are 3-dimensional. We set the dimensionality of the subspaces to 2 for all the methods.

Figure 3.2 illustrates the original synthetic dataset and domain adaptation results of different methods on the dataset. It can be seen that after the SA method the divergences between domains are still large after aligning the subspaces. Hence, the aligned subspaces are not optimal to reduce the domain shift if the distribution divergence is not considered. The SDA method does not demonstrate obvious improvement over SA, since the variance shift is reduced based upon the aligned subspaces (which may not be optimal) as in SA. TCA method reduces the domain shift effectively. However, two of the classes are mixed up since there may not exist a unified subspace to reduce domain shift and preserve the original information simultaneously. Even with conditional distribution shift reduction (JDA) or instances reweighting (TJM), the class-1 and class-2 still cannot be distinguished. SCA considers the total scatter, domain scatter, and class scatter using a unified mapping. However, there may not exist such a common subspace that satisfies all the constraints. Obviously, JGSA aligns the two domains well even though the shift

\(^a\)http://www.uow.edu.au/~jz960/
between source and target domains is large.

### 3.3.2 Real World Datasets

We evaluate our method on three cross-domain visual recognition tasks: object recognition (Office, Caltech-256), hand-written digit recognition (USPS, MNIST), and RGB-D-based action recognition (MSRAction3DExt, G3D, UTD-MHAD, and MAD). The sample images or video frames are shown in Figure 3.1.

![Sample images of object datasets, digit datasets, and sample video frames of depth map of RGB-D-based action datasets.](image)

**Figure 3.1:** Sample images of object datasets, digit datasets, and sample video frames of depth map of RGB-D-based action datasets.

**Setup**

**Object Recognition** We adopt the public Office+Caltech object datasets released by Gong et al. [93]. This dataset contains images from four different domains: Amazon (images downloaded from online merchants), Webcam (low-resolution images by a web camera), DSLR (high-resolution images by a digital SLR camera), and Caltech-256. Amazon, Webcam, and DSLR are three datasets studied in [224] for the effects of domain shift. Caltech-256 [102] contains 256 object classes downloaded from Google images. Ten classes common to four datasets are selected: backpack, bike, calculator, headphones,
keyboard, laptop, monitor, mouse, mug, and projector. Two types of features are considered: SURF descriptors (which are encoded with 800-bin histograms with the codebook trained from a subset of Amazon images), and Decaf6 features (which are the activations of the 6th fully connected layer of a convolutional network trained on ImageNet). As suggested by [93], 1-Nearest Neighbor Classifier (NN) is chosen as the base classifier. For the free parameters, we set $k = 30$, $T = 10$, and $\beta = 0.1$.

**Digit Recognition**  For cross-domain hand-written digit recognition task, we use MNIST [150] and USPS [116] datasets to evaluate our method. The MNIST dataset contains a training set of 60,000 examples, and a test set of 10,000 examples of size $28 \times 28$. USPS dataset consists of 7,291 training images and 2,007 test images of size $16 \times 16$. Ten shared classes of the two datasets are selected. We follow the settings of [177, 178] to construct a pair of cross-domain datasets USPS $\rightarrow$ MNIST by randomly sampling 1,800 images in USPS to form the source data, and randomly sampling 2,000 images in MNIST to form the target data. Then source and target pair are switched to form another dataset MNIST $\rightarrow$ USPS. All images are uniformly rescaled to size $16 \times 16$, and each image is represented by a feature vector encoding the gray-scale pixel values. For the free parameters, we set $k = 100$, $T = 10$, and $\beta = 0.01$.

**Table 3.1:** Accuracy(%) on cross-domain object datasets using SURF features. Notation for datasets: Caltech:C; Amazon:A; Webcam:W; DSLR:D.

<table>
<thead>
<tr>
<th>data</th>
<th>Raw</th>
<th>SA</th>
<th>SDA</th>
<th>GFK</th>
<th>TCA</th>
<th>JDA</th>
<th>TJM</th>
<th>SCA</th>
<th>JGSA primal</th>
<th>JGSA linear</th>
<th>JGSA RBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>C $\rightarrow$ A</td>
<td>36.01</td>
<td>49.27</td>
<td>49.69</td>
<td>46.03</td>
<td>45.82</td>
<td>45.62</td>
<td>46.76</td>
<td>45.62</td>
<td>51.46</td>
<td>52.30</td>
<td>53.13</td>
</tr>
<tr>
<td>C $\rightarrow$ W</td>
<td>29.15</td>
<td>40.00</td>
<td>38.98</td>
<td>36.95</td>
<td>31.19</td>
<td>41.69</td>
<td>38.98</td>
<td>40.00</td>
<td>45.42</td>
<td>45.76</td>
<td>48.47</td>
</tr>
<tr>
<td>C $\rightarrow$ D</td>
<td>38.22</td>
<td>39.49</td>
<td>40.13</td>
<td>40.76</td>
<td>34.39</td>
<td>45.22</td>
<td>44.59</td>
<td>47.13</td>
<td>45.86</td>
<td>48.41</td>
<td>48.41</td>
</tr>
<tr>
<td>A $\rightarrow$ C</td>
<td>34.19</td>
<td>39.98</td>
<td>39.54</td>
<td>40.69</td>
<td>42.39</td>
<td>39.36</td>
<td>39.45</td>
<td>39.72</td>
<td>41.50</td>
<td>38.11</td>
<td>41.50</td>
</tr>
<tr>
<td>A $\rightarrow$ W</td>
<td>31.19</td>
<td>33.22</td>
<td>30.85</td>
<td>36.95</td>
<td>36.27</td>
<td>37.97</td>
<td>42.03</td>
<td>34.92</td>
<td>45.76</td>
<td>49.49</td>
<td>45.08</td>
</tr>
<tr>
<td>A $\rightarrow$ D</td>
<td>35.67</td>
<td>33.76</td>
<td>33.76</td>
<td>40.13</td>
<td>33.76</td>
<td>39.49</td>
<td>45.22</td>
<td>39.49</td>
<td>47.13</td>
<td>45.86</td>
<td>45.22</td>
</tr>
<tr>
<td>W $\rightarrow$ C</td>
<td>28.76</td>
<td>35.17</td>
<td>34.73</td>
<td>24.76</td>
<td>29.39</td>
<td>31.17</td>
<td>30.19</td>
<td>31.08</td>
<td>33.21</td>
<td>32.68</td>
<td>33.57</td>
</tr>
<tr>
<td>W $\rightarrow$ A</td>
<td>31.63</td>
<td>39.25</td>
<td>39.25</td>
<td>27.56</td>
<td>28.91</td>
<td>32.78</td>
<td>29.96</td>
<td>29.96</td>
<td>39.87</td>
<td>41.02</td>
<td>40.81</td>
</tr>
<tr>
<td>W $\rightarrow$ D</td>
<td>84.71</td>
<td>75.16</td>
<td>75.80</td>
<td>85.35</td>
<td>89.17</td>
<td>89.17</td>
<td>89.17</td>
<td>87.26</td>
<td>90.45</td>
<td>90.45</td>
<td>88.54</td>
</tr>
<tr>
<td>D $\rightarrow$ C</td>
<td>29.56</td>
<td>34.55</td>
<td>35.89</td>
<td>29.30</td>
<td>30.72</td>
<td>31.52</td>
<td>31.43</td>
<td>30.72</td>
<td>29.92</td>
<td>30.19</td>
<td>30.28</td>
</tr>
<tr>
<td>D $\rightarrow$ A</td>
<td>28.29</td>
<td>39.87</td>
<td>38.73</td>
<td>28.71</td>
<td>31.00</td>
<td>33.09</td>
<td>32.78</td>
<td>31.63</td>
<td>38.00</td>
<td>36.01</td>
<td>38.73</td>
</tr>
<tr>
<td>D $\rightarrow$ W</td>
<td>83.73</td>
<td>76.95</td>
<td>76.95</td>
<td>80.34</td>
<td>86.10</td>
<td>89.49</td>
<td>85.42</td>
<td>84.41</td>
<td>91.86</td>
<td>91.86</td>
<td>93.22</td>
</tr>
<tr>
<td>Average</td>
<td>40.93</td>
<td>44.72</td>
<td>44.52</td>
<td>43.13</td>
<td>43.26</td>
<td>46.38</td>
<td>46.33</td>
<td>45.16</td>
<td>50.04</td>
<td>50.18</td>
<td>50.58</td>
</tr>
</tbody>
</table>

**RGB-D-based Action Recognition**  For cross-dataset RGB-D-based Action Recognition, four RGB-D-based Action Recognition datasets are selected, namely MSRAAction3DExt [159, 275], UTD-MHAD [33], G3D[21], and MAD [114]. All the four datasets are captured...
Figure 3.2: Comparisons of baseline domain adaptation methods and the proposed JGSA method on the synthetic data

(a) regularization parameter $\beta$  (b) dimentionality of subspace $k$  (c) number of iteration $T$

Figure 3.3: Parameter sensitivity study of JGSA on different types of datasets

by both RGB and depth sensors. We select the shared actions between MSRAction3DExt and other three datasets to form 6 dataset pairs. There are 8 common actions between MSRAction3DExt and G3D: wave, forward punch, hand clap, forward kick, jogging, tennis swing, tennis serve, and golf swing. There are 10 common actions between MSRARc-
Table 3.2: Accuracy(%) on cross-domain object datasets using Decaf6 features. Notation for datasets: Caltech:C; Amazon:A; Webcam:W; DSLR:D.

<table>
<thead>
<tr>
<th>data</th>
<th>JDA</th>
<th>OTGL</th>
<th>JGSA primal</th>
<th>JGSA linear</th>
<th>JGSA RBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>C→A</td>
<td>90.19</td>
<td></td>
<td>92.15</td>
<td>91.44</td>
<td>91.75</td>
</tr>
<tr>
<td>C→W</td>
<td>85.42</td>
<td>84.17</td>
<td>86.78</td>
<td>85.08</td>
<td>83.39</td>
</tr>
<tr>
<td>C→D</td>
<td>85.99</td>
<td>87.25</td>
<td>93.63</td>
<td>92.36</td>
<td>92.36</td>
</tr>
<tr>
<td>A→C</td>
<td>81.92</td>
<td></td>
<td>85.51</td>
<td>84.86</td>
<td>84.86</td>
</tr>
<tr>
<td>A→W</td>
<td>80.68</td>
<td>83.05</td>
<td>81.02</td>
<td>84.75</td>
<td>80.00</td>
</tr>
<tr>
<td>A→D</td>
<td>81.53</td>
<td>85.00</td>
<td>88.54</td>
<td>85.35</td>
<td>84.71</td>
</tr>
<tr>
<td>W→C</td>
<td>81.21</td>
<td>81.45</td>
<td>84.95</td>
<td>84.68</td>
<td>84.51</td>
</tr>
<tr>
<td>W→A</td>
<td>90.71</td>
<td>90.62</td>
<td>90.71</td>
<td>91.44</td>
<td>91.34</td>
</tr>
<tr>
<td>W→D</td>
<td>100.00</td>
<td>96.25</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>D→C</td>
<td>80.32</td>
<td>84.11</td>
<td>86.20</td>
<td>85.75</td>
<td>84.77</td>
</tr>
<tr>
<td>D→A</td>
<td>91.96</td>
<td>92.31</td>
<td>91.96</td>
<td>92.28</td>
<td>91.96</td>
</tr>
<tr>
<td>D→W</td>
<td>99.32</td>
<td>96.29</td>
<td>99.66</td>
<td>98.64</td>
<td>98.64</td>
</tr>
<tr>
<td>Average</td>
<td>87.44</td>
<td>88.18</td>
<td>89.98</td>
<td>89.76</td>
<td>88.97</td>
</tr>
</tbody>
</table>

Table 3.3: Accuracy (%) on cross-domain digit datasets.

<table>
<thead>
<tr>
<th>data</th>
<th>Raw</th>
<th>SA</th>
<th>SDA</th>
<th>GFK</th>
<th>TCA</th>
<th>JDA</th>
<th>TJM</th>
<th>SCA</th>
<th>JGSA primal</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST→USPS</td>
<td>65.94</td>
<td>67.78</td>
<td>65.00</td>
<td>61.22</td>
<td>56.33</td>
<td>67.28</td>
<td>63.28</td>
<td>65.11</td>
<td>80.44</td>
</tr>
<tr>
<td>USPS→MNIST</td>
<td>44.70</td>
<td>48.80</td>
<td>35.70</td>
<td>46.45</td>
<td>51.20</td>
<td>59.65</td>
<td>52.25</td>
<td>48.00</td>
<td>68.15</td>
</tr>
<tr>
<td>Average</td>
<td>55.32</td>
<td>58.29</td>
<td>50.35</td>
<td>56.84</td>
<td>53.77</td>
<td>63.47</td>
<td>57.77</td>
<td>56.56</td>
<td>74.30</td>
</tr>
</tbody>
</table>

Table 3.4: Accuracy (%) on cross-dataset RGB-D-based action datasets.

<table>
<thead>
<tr>
<th>data</th>
<th>Raw</th>
<th>SA</th>
<th>SDA</th>
<th>TCA</th>
<th>JDA</th>
<th>TJM</th>
<th>SCA</th>
<th>JGSA linear</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSR→G3D</td>
<td>72.92</td>
<td>77.08</td>
<td>73.96</td>
<td>68.75</td>
<td>82.29</td>
<td>70.83</td>
<td>70.83</td>
<td>89.58</td>
</tr>
<tr>
<td>G3D→MSR</td>
<td>54.47</td>
<td>68.09</td>
<td>67.32</td>
<td>50.58</td>
<td>65.37</td>
<td>63.04</td>
<td>55.25</td>
<td>66.93</td>
</tr>
<tr>
<td>MSR→UTD</td>
<td>66.88</td>
<td>73.75</td>
<td>73.75</td>
<td>65.00</td>
<td>77.50</td>
<td>65.00</td>
<td>64.38</td>
<td>76.88</td>
</tr>
<tr>
<td>UTD→MSR</td>
<td>62.93</td>
<td>67.91</td>
<td>66.67</td>
<td>57.63</td>
<td>61.06</td>
<td>60.12</td>
<td>55.14</td>
<td>61.37</td>
</tr>
<tr>
<td>MSR→MAD</td>
<td>80.71</td>
<td>85.00</td>
<td>83.57</td>
<td>79.29</td>
<td>82.86</td>
<td>82.14</td>
<td>78.57</td>
<td>86.43</td>
</tr>
<tr>
<td>MAD→MSR</td>
<td>80.09</td>
<td>81.48</td>
<td>80.56</td>
<td>81.02</td>
<td>83.33</td>
<td>79.63</td>
<td>79.63</td>
<td>85.65</td>
</tr>
<tr>
<td>Average</td>
<td>69.67</td>
<td>75.55</td>
<td>74.30</td>
<td>67.05</td>
<td>75.40</td>
<td>70.13</td>
<td>67.30</td>
<td>77.81</td>
</tr>
</tbody>
</table>

Joint3DExt and UTD-MHAD: wave, hand catch, right arm high throw, draw x, draw circle, two hand front clap, jogging, tennis swing, tennis serve, and pickup and throw. There are 7 shared actions between MSRAction3DExt and MAD: wave, forward punch, throw, forward kick, side kick, jogging, and tennis swing forehand. The local HON4D feature is used for the cross-dataset action recognition tasks. We extract local HON4D descriptors around 15 skeleton joints by following the process similar to [197]. The selected joints include head, neck, left knee, right knee, left elbow, right elbow, left wrist, right wrist, left shoulder, right shoulder, hip, left hip, right hip, left ankle, and right an-
We use a patch size of $24 \times 24 \times 4$ for depth map with resolution of $320 \times 240$ and $48 \times 48 \times 4$ for depth map with resolution of $640 \times 480$, then divide the patches into a $3 \times 3 \times 1$ grid. Since most of the real world applications of action recognition are required to recognize unseen data in the target domain, we further divide the target domain into training and test sets using cross-subject protocol, where half of the subjects are used as training and the rest subjects are used as test when a dataset is evaluated as target domain. Note that the target training set is also unlabeled. For the free parameters, we set $k = 100$ and $\beta = 0.01$. To avoid overfitting to the target training set, we set $T = 1$ in action recognition tasks. LibLINEAR [68] is used for action recognition by following the original paper [197].

**Results and Discussion**

The results on three types of real-world cross-domain (object, digit, and action) datasets are shown in Table 3.1, 3.2, 3.3, and 3.4. The JGSA primal represents the results of JGSA method on original data space, while the JGSA linear and JGSA RBF represent the results with linear kernel and RBF kernel respectively. We follow JDA to report the results on digit datasets in the original feature space. For the action recognition task, it is hard to do eigendecomposition in the original space due to the high dimensionality, hence, the results are obtained using a linear kernel. It can be observed that JGSA outperforms the state-of-the-art domain adaptation methods on most of the datasets. As mentioned, the general drawback of subspace centric approach is that the distribution shifts between domains are not explicitly reduced. The data-centric methods reduce the distribution shift explicitly. However, a unified transformation may not exist to both reduce distribution shift and preserve the properties of the original data. Hence, JGSA outperforms both subspace centric and data-centric methods on most of the datasets. We also compare the primal and kernelized versions of the algorithm on the object recognition task (Table 3.1 and 3.2). The results show that the primal and kernelized versions can obtain similar results on average. To evaluate the effectiveness of pseudo labelling, we compare our method with a semi-supervised method KEMA [261]. We use the same Decaf7 feature on 8 Office-Caltech dataset pairs as did in KEMA. Our method obtains 90.18% (linear) and 89.91% (RBF), both of which are higher than 89.1% reported in KEMA.

We also evaluated the runtime complexity on the cross-domain object datasets (SURF with the linear kernel). The average runtime is 28.97s, which is about three times as long as the best baseline method (JDA). This is because JGSA learns two mappings simultaneously, the size of the matrix for eigendecomposition is doubled compared to JDA.
Parameter Sensitivity

We analyse the parameter sensitivity of JGSA on different types of datasets to validate that a wide range of parameter values can be chosen to obtain satisfactory performance. The results on different types of datasets have validated that the fixing $\lambda = 1$ and $\mu = 1$ is sufficient for all the three tasks. Hence, we only evaluate the other three parameters ($k$, $\beta$, and $T$). We conduct experiments on the USPS→MNIST, W→A (SURF descriptor with the linear kernel), and MSR→MAD datasets for illustration, which are shown in Figure 3.3. The solid line is the accuracy on JGSA using different parameters, and the dashed line indicates the results obtained by the best baseline method on each dataset. Similar trends are observed on other datasets.

$\beta$ is the trade-off parameter within and between class variance of the source domain. If $\beta$ is too small, the class information of the source domain is not considered. If $\beta$ is too big, the classifier would overfit to the source domain. However, it can be seen from Figure 3.3a, a large range of $\beta$ ($\beta \in [2^{-15}, 0.5]$) can be selected to obtain better results than the best baseline method.

Figure 3.3b illustrates the relationship between various $k$ and the accuracy. We can choose $k \in [20, 180]$ to obtain better results than the best baseline method.

For the number of iteration $T$, the results on object and digit recognition tasks can be converged to the optimum value after several iterations. However, for the action recognition, the accuracy has no obvious change (Figure 3.3c). This may be because we use a different protocol for action recognition as mentioned in Section 3.3.2. After iterative labelling (which is done on the target training set), the mappings may be sufficiently good for fitting the target training set, but it is not necessarily the case for the test set.

3.4 Summary

In this chapter, we propose a novel framework for unsupervised domain adaptation, referred to as Joint Geometrical and Statistical Alignment (JGSA). JGSA reduces the domain shifts by taking both geometrical and statistical properties of source and target domain data into consideration and exploiting both shared and domain-specific features. Comprehensive experiments on synthetic data and three different types of real-world visual recognition tasks validate the effectiveness of JGSA compared to several state-of-the-art domain adaptation methods.
Chapter 4

Unsupervised Domain Adaptation: A Multi-task Learning-based Method

4.1 Motivation

Similar to Chapter 3, we also focus on the problem of unsupervised domain adaptation in this chapter, where source data are labelled, but available data in the target domain are unlabelled. One approach to this problem is referred to as feature transformation-based domain adaptation [201, 93, 177, 74, 10, 313], which transforms the original feature into another space where the distributions of the two domains would be similar. A classifier can then be trained in the space for both source and target data. Another approach [214, 176], referred to as classifier-based domain adaptation, aims at learning a single and shared classifier directly such that the distribution of classification scores on the source data aligns with that on the target data. However, there may not exist such a classifier that could perform well on both domains, especially when the domain shift is large. An alternative approach is to jointly learn two classifiers, one is optimized for the source domain and the other is optimized for the target domain. Yang et al. [301] proposed a method to learn two classifiers for the source and target domain respectively for the case where both the source and target data are labelled (i.e. supervised domain adaptation). Duan et al. [64] propose a multi-source domain adaptation method that requires labelled target data. Recently, several deep-learning based domain adaptation methods are also proposed (DAN [174], RTN [180], and WDAN [300]). Since most of these deep learning methods still follow the idea of shared classifiers (parameters), their capability of adaptation would drop when the shift becomes large.

Unlike the works [214, 176, 174, 300] that generally assume the existence of a shared classifier that could perform well on both source and target domain, and the works [301, 64] that require the availability of labelled target data, this chapter provides a new perspective that unsupervised domain adaptation can be formulated as a multi-task learning
problem and proposes a novel classifier-based method to implement this formulation. Such a formulation allows us to learn unshared classifiers between the source and target to deal with a large shift. Specifically, the source task is to learn a linear classifier from the labelled source data and the target task is to learn a linear transform to cluster the unlabelled target data such that the original target data are mapped to a lower dimensional subspace where the geometric structure is preserved. To effectively transfer the discriminative information from the source domain, the two tasks are jointly learned by enforcing the target transformation is close to the source classifier and the class distribution shift between domains is reduced in the meantime, such that the mapped target data are clustered to their corresponding classes.

The proposed method is inspired by the adaptation regularization based transfer learning method (ARTL) proposed by Long et. al. [176]. The ARTL method learns a shared adaptive classifier by simultaneously optimizing the structural risk functional, the joint distribution matching between domains, and the manifold consistency underlying marginal distribution. The ARTL is an effective domain adaptation method which learns the adaptive classifier directly without the need of intermediate subspace learning procedures, and the state-of-the-art results were obtained on many cross-domain recognition datasets at the time. However, there are two key differences between our method and ARTL. First, ARTL learns a shared classifier for both source and target domain. By contrast, we propose to learn unshared classifiers between the source and target to deal with a large shift. Secondly, to cluster the unlabelled target data, we propose to regularize the manifold structure in the target domain only instead of regularizing the cross-domain (i.e. all the data in source and target domains) manifold structure in ARTL. The assumption for doing so in ARTL is that if the marginal distributions of two samples in the source and target domain are similar, the conditional distributions of them are also similar. However, if the geometrical structure of cross-domain data in the original space is preserved, the domain shift cannot be effectively reduced. By contrast, the proposed method does not require this assumption to be true and only regularize the target domain to preserve the structure information in the target domain. Experiments on both synthetic and real-world cross-domain recognition tasks have shown the effectiveness of the proposed method.

4.2 Proposed Method

This section presents the multi-task learning-based Unsupervised Domain Adaptation (mtUDA) method in detail. It begins with the definitions of terminologies. The source domain data denoted as $X_s \in \mathbb{R}^{D \times n_s}$ are drawn from distribution $P_s(X_s)$ and the target domain data denoted as $X_t \in \mathbb{R}^{D \times n_t}$ are drawn from distribution $P_t(X_t)$, where $D$ is the dimension of the data instance, $n_s$ and $n_t$ are number of samples in source and target domain respectively. We focus on the unsupervised domain adaptation problem which assumes that
there are sufficient labeled source domain data, \( \mathcal{D}_s = \{(x_i, y_i)\}_{i=1}^{n_s}, x_i \in \mathbb{R}^D \), and unlabeled target domain data, \( \mathcal{D}_t = \{(x_j)\}_{j=1}^{n_t}, x_j \in \mathbb{R}^D \), in the training stage. The feature spaces and label spaces between domains are assumed same: \( X_s = X_t \) and \( Y_s = Y_t \). However, due to the dataset shift, \( P_s(X_s) \neq P_t(X_t) \) and \( P_s(Y_s | X_s) \neq P_t(Y_t | X_t) \).

### 4.2.1 Formulation

#### Regularized Risk Minimization

Suppose there is no domain shift between source and target domains, the classifier learnt on the labeled source data can be applied on the target samples directly using the standard regularization based supervised learning algorithm in a Reproducing Kernel Hilbert Space (RKHS),

\[
\min_{f \in \mathcal{H}_K} \frac{1}{n_s} \sum_{i=1}^{n_s} \mathcal{L}(f(x_i), y_i) + \gamma \|f\|_K^2
\]

where \( \mathcal{H}_K \) is an appropriately chosen RKHS, \( K \) is a Mercer kernel, \( \mathcal{L} \) denotes some loss functions, which can be squared loss \((y_i - f(x_i))^2\) for RLS or hinge loss \(\max[0, 1 - y_if(x_i)]\) for SVM, \(\gamma\) is the shrinkage regularization parameter to regulate the complexity of learned model.

However, the source classifier is likely not optimal for the target domain due to the existence of domain shift. Hence, two different classifiers on source and target domain respectively are to learned with additional regularization on the two classifiers in a similar way to multi-task learning.

#### Multi-task Learning

The key idea of multi-task learning (MTL) is that the performances of the related tasks can be boosted by learning them jointly. In general, the MTL is based on the regularized risk minimization learning framework. In the context of domain adaptation, source and target tasks are two related tasks and the MTL formulation can be expressed as follows,

\[
\min_{f \in \mathcal{H}_K} \frac{1}{n_s} \sum_{i=1}^{n_s} \mathcal{L}(f_s(x_i), y_i) + \frac{1}{n_t} \sum_{j=1}^{n_t} \mathcal{L}(f_t(x_j), y_j) + \gamma_s \|f_s\|_K^2 + \gamma_t \|f_t\|_K^2 + \gamma_M \Omega(f_s, f_t)
\]

where \(\Omega\) is a regularization on the source and target classifiers, \(\gamma_M\) is the classifier regularization parameter. In our problem, the source and target tasks are the same, suggesting that the label spaces between domains are identical. A simple regularization is \(\Omega(f_s, f_t) = \|f_s - f_t\|_K^2\).

The MTL formulation in Eq.(4.2) requires labeled data in both domains. However, there are no labeled data in the target domain. Eq.(4.2) cannot be solved directly. In
CHAPTER 4. A MULTI-TASK LEARNING-BASED METHOD

this chapter, it is proposed to leverage the idea of manifold regularization [12] to learn the intrinsic structure of the target domain to allow that the target task is treated as an unsupervised clustering task.

Multi-task Learning-based Unsupervised Domain Adaptation

Since there are no labeled data in the target domain, the target risk minimization term in Eq.(4.2) is not computable. This chapter proposes to replace the risk minimization term with an intrinsic regularization term to preserve the structure of the target data. Note that we use the manifold assumption here because it is assumed that the unlabelled target domain data are sufficient (such that the statistic distribution of the target domain can be represented by the target domain unlabelled data) in the general setting of unsupervised domain adaptation for feasible adaptation. Hence, the assumption in our multi-task learning-based method is the same as regular unsupervised domain adaptation methods since the target domain data should not be sparse in general. Eq.(4.2) becomes

\[
\min_{f \in \mathcal{H}} \sum_{s=1}^{n_s} \mathcal{L}(f_s(x_i), y_i) + \frac{\gamma}{2} \|f_t\|^2 + \gamma_A (\|f_s\|_K^2 + \|f_t\|_K^2) + \gamma_M \Omega(f_s, f_t) \quad (4.3)
\]

where \(\|f_t\|^2 = \frac{1}{n_t} \sum_{i,j} (f_t(x_i) - f_t(x_j))^2 W_{ij} = \frac{1}{n_t^2} \text{tr}(f_t^T L f_t)\) is the intrinsic manifold regularization for the target domain, \(f_t = [f_t(x_1), ..., f_t(x_{n_t})]^T\), \(W_{ij}\) are edge weights in the adjacency graph (with p-nearest neighbours of each data point) of the target data, \(L = D - W\) is the graph Laplacian, \(D\) is a diagonal matrix given by \(D_{ii} = \sum_{j=1}^{n_t} W_{ij}\), \(\gamma\) is a manifold regularization parameter.

The use of multi-task regularization can remove additional conditions, such as orthogonal constraint, to avoid degenerate solutions as required in manifold regularization-based unsupervised learning [12]. Note that the term \(\|f_t\|^2\) is different from the manifold regularization term in [176]. It only regulates the manifold structure in the target domain rather than the cross-domain (all the data in source and target domains) as done in [176]. The assumption for doing so in [176] is that if the marginal distributions of two samples in the source and target domain are similar, the conditional distributions of them are also similar. However, if the geometrical structure of cross-domain data in the original space is preserved, the domain shift cannot be effectively reduced. By contrast, the proposed method does not require this assumption to be true and treats the source and target domain separately as two joint tasks rather than as a whole for exploiting more domain specific features and reducing the domain shift effectively.

Eq.(4.3) has considered the source risk minimization, task relatedness of the source and target domains, and target domain intrinsic structure in a multi-task learning framework. The target task can be seen as an unsupervised clustering task. However, the ultimate goal is to assign class labels to target samples rather than just grouping them. Hence, the
class information from the source domain needs to be leveraged. Since there is the shift between the source and target domains and the manifold structure across the domains is not regularized, additional terms to reduce the class distribution shift are required to make sure that the target clusters are assigned with right class labels. In this chapter, it is proposed to use the Maximum Mean Discrepancy (MMD) criterion to regulate both the marginal and the conditional distribution shift [177, 176].

The final objective function of the multi-task learning-based Unsupervised Domain Adaptation (mtUDA) method is

$$
\min_{f_s, f_t \in H} \frac{1}{n_s} \sum_{i=1}^{n_s} \mathcal{L}(f_s(x_i), y_i) + \gamma_l \|f_t\|_T^2 + \gamma_A (\|f_s\|_K^2 + \|f_t\|_K^2) + \gamma_M \Omega(f_s, f_t) + \gamma_D D(P_s, P_t) \tag{4.4}
$$

where $\gamma_D$ is the MMD regularization parameter,

$$
D(P_s, P_t) = \frac{1}{n_s} \sum_{i=1}^{n_s} f_s(x_i) - \frac{1}{n_t} \sum_{j=1}^{n_t} f_t(x_j) \|_F^2 + \sum_{c=1}^{C} \frac{1}{n_s} \sum_{x_i \in D_s^{(c)}} f_s(x_i) - \frac{1}{n_t} \sum_{x_j \in D_t^{(c)}} f_t(x_j) \|_F^2
$$

is the MMD measure of joint distribution distance between source and target domains, $M = M_0 + \sum_{c=1}^{C} M_c$, and

$$(M_0)_{ij} = \begin{cases} 
\frac{1}{n_i n_j} & \text{if } x_i, x_j \in D_s \\
\frac{1}{n_i n_t} & \text{if } x_i, x_j \in D_t \\
\frac{-1}{n_t n_i} & \text{otherwise}
\end{cases}, \quad (M_c)_{ij} = \begin{cases} 
\frac{1}{n_i^{(c)} n_j^{(c)}} & \text{if } x_i, x_j \in D_s^{(c)} \\
\frac{1}{n_i^{(c)} n_t^{(c)}} & \text{if } x_i, x_j \in D_t^{(c)} \\
\frac{-1}{n_t^{(c)} n_i^{(c)}} & \text{if } x_i \in D_s^{(c)}, x_j \in D_t^{(c)} \\
\frac{-1}{n_t^{(c)} n_i^{(c)}} & \text{if } x_j \in D_s^{(c)}, x_i \in D_t^{(c)} \\
0 & \text{otherwise}
\end{cases}$$

$f_s = [f_s(x_1), ..., f_s(x_{n_s})]^T$. Since there is no labeled data in the target domain, pseudo labels are obtained using some base classifiers (e.g. NN) trained on the source domain data in a similar way to [176]. Thus, the conditional distributions can be compared. The pseudo labels are iteratively updated after obtaining the adaptive classifier for the target data.

In the following, two mtUDA algorithms with two different loss functions are presented, namely Regularized Least Squares and Support Vector Machines.
4.2.2 Regularized Least Squares Algorithm

The Regularized Least Squares algorithm (denoted as mtUDA-RLS) can be expressed as,

\[
\min_{f_s, f_t} \frac{1}{n_s} \sum_{i=1}^{n_s} (y_i - f_s(x_i))^2 + \frac{\gamma}{n_t^2} \text{tr}(f_t^T L f_t) + \gamma_3 (\|f_s\|^2_K + \|f_t\|^2_K) + \gamma_t \|f_s - f_t\|^2_K + \gamma_d \text{tr}(f_t^T M f_t) \tag{4.6}
\]

Based on the Representer Theorem, the solution is an expansion of kernel functions over all the data:

\[
f_s^*(x) = \sum_{i=1}^{n_s+n_t} \alpha_i^s K(x, x_i), \quad f_t^*(x) = \sum_{i=1}^{n_s+n_t} \alpha_i^t K(x, x_i) \tag{4.7}
\]

Substituting them into Eq.(4.6), the following objective function can be obtained,

\[
(\alpha_i^s, \alpha_i^t) = \arg \min_{\alpha_s, \alpha_t \in \mathbb{R}^{(n_s+n_t) \times C}} \frac{1}{n_s} \text{tr}((Y_s - \alpha_s^T K_s^T)(Y_s - \alpha_s^T K_s^T)^T) + \frac{\gamma}{n_t^2} \text{tr}(\alpha_t^T K_t^T L K_t \alpha_t) + \gamma_t \text{tr}(\alpha_s^T K_s \alpha_s + \alpha_t^T K_t \alpha_t) + \gamma_3 (\|f_s\|^2_K + \|f_t\|^2_K) + \gamma_d \text{tr}(f_t^T M f_t) \tag{4.8}
\]

where \(Y_s \in \mathbb{R}^{C \times n_s}\) is a label matrix, \(y_s^c = 1\) if \(y_s(x) = c\), otherwise \(y_s^c = 0\) (mtUDA-RLS can be naturally applied to multi-class classification problem directly with this form of label matrix), \(K = \Phi(X)^T \Phi(X)\), \(K_s = \Phi(X_s)^T \Phi(X)\), and \(K_t = \Phi(X_t)^T \Phi(X)\) are the kernel matrices, \(X = [X_s, X_t]\) denotes all the source and target training samples, \(\Phi(X) = [\phi(x_1), \ldots, \phi(x_{n_s+n_t})]\) is the feature mappings to a space of a higher or even infinite dimension.

To simultaneously optimize \(\alpha_s\) and \(\alpha_t\), we write \([\alpha_s^T \quad \alpha_t^T]\) as \(\alpha^T\). The objective function (Eq.(4.8)) can be rewritten as follows,

\[
\alpha^* = \arg \min_{\alpha \in \mathbb{R}^{(n_s+n_t) \times C}} \frac{1}{n_s} \text{tr}(Y_s Y_s^T - 2\alpha^T \begin{bmatrix} K_s^T Y_s \\ 0 \end{bmatrix}) + \frac{1}{n_s} \text{tr}(\alpha^T \begin{bmatrix} K_t^T K_s & 0 \\ 0 & 0 \end{bmatrix} \alpha) + \frac{\gamma}{n_t^2} \text{tr}(\alpha^T \begin{bmatrix} 0 & 0 \\ 0 & K_t^T L K_t \end{bmatrix} \alpha) + \gamma_t \text{tr}(\alpha^T \begin{bmatrix} K_s & 0 \\ 0 & K_t \end{bmatrix} \alpha) + \gamma_3 \text{tr}(\alpha^T \begin{bmatrix} K_s^T & 0 \\ 0 & K_t^T \end{bmatrix} M \begin{bmatrix} K_s & 0 \\ 0 & K_t \end{bmatrix} \alpha) + \gamma_d \text{tr}(\alpha^T \begin{bmatrix} K_t^T & 0 \\ 0 & K_t \end{bmatrix} M \begin{bmatrix} K_s & 0 \\ 0 & K_t \end{bmatrix} \alpha) \tag{4.9}
\]
Setting the derivatives with respect to $\alpha$ to zero, the closed form solution $\alpha^\star$ is obtained

$$
\alpha^\star = \left( \begin{bmatrix} K_T K_s & 0 \\ 0 & 0 \end{bmatrix} + \frac{\gamma_I}{n_l^2} \begin{bmatrix} 0 & 0 \\ 0 & K_T L K_s \end{bmatrix} + \gamma_H \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} + \gamma_M \begin{bmatrix} K & -K \\ -K & K \end{bmatrix} + \gamma_D \begin{bmatrix} K_T Y_s & 0 \\ 0 & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ K_T Y_s \end{bmatrix}
$$

(4.10)

where $\varepsilon I$ is included to obtain numerically more stable solution with $I$ be an identity matrix and $\varepsilon$ be a small constant. Following [12], each trade-off coefficient is treated as a whole, e.g. $\hat{\gamma}_I = \frac{\gamma_I}{n_l^2}$, $\hat{\gamma}_A = \gamma_A n_s$, $\hat{\gamma}_M = \gamma_M n_s$, and $\hat{\gamma}_D = \gamma_D n_s$, when tuning the parameters.

### 4.2.3 Support Vector Machines Algorithm

The Support Vector Machines algorithm (denoted as mtUDA-SVM) can be formulated as,

$$
\begin{align*}
\min_{f_s, f_t \in H} & \frac{1}{2} \sum_{i=1}^{n_s} \max(0, 1 - y_i f_s(x_i)) + \frac{\eta}{n_l^2} tr(f_t^T L f_t) + \gamma_A (\|f_s\|^2_K + \|f_t\|^2_K) \\
&+ \gamma_M \|f_s - f_t\|^2_K + \gamma_D tr([f_s^T f_t^T M [f_s f_t])
\end{align*}

(4.11)

Based on the Representer Theorem, substituting Eqs.(4.7) into Eq.(4.11), and following [12] to solve the objective function by introducing the Lagrangian, yields

$$
\beta^\star = \max_{\beta \in \mathbb{R}^{n_s}} \sum_{i=1}^{n_s} \beta_i - \frac{1}{2} \beta^T Q \beta
$$

(4.12)

s.t. $\sum_{i=1}^{n_s} \beta_i y_i = 0$, $0 \leq \beta_i \leq \frac{1}{n_s}$, $i = 1, ..., n_s$

where

$$
Q = \begin{bmatrix} K_s^T Y_s & 0 \\ 0 & 0 \end{bmatrix}^T \begin{bmatrix} 2 \frac{\eta}{n_l^2} & 0 \\ 0 & 0 \end{bmatrix} + 2 \gamma_A \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} + 2 \gamma_M \begin{bmatrix} K & -K \\ -K & K \end{bmatrix} + 2 \gamma_D \begin{bmatrix} K_T Y_s & 0 \\ 0 & 0 \end{bmatrix} \\
+ 2 \gamma_D \begin{bmatrix} K_T Y_s & 0 \\ 0 & 0 \end{bmatrix}^T \begin{bmatrix} K_T Y_s & 0 \\ 0 & 0 \end{bmatrix} + \varepsilon I \right)^{-1} \begin{bmatrix} 0 \\ K_T Y_s \end{bmatrix},
$$
and $\tilde{Y}_s = \text{diag}(y_1, \ldots, y_{n_s})$. Then a standard SVM solver [31] with quadratic form induced by $Q$ can be used to obtain $\beta^*$, and the final classifier can be obtained by

$$
\alpha^* = \left( \frac{2\gamma_I}{n_t^2} \begin{bmatrix} 0 & K^T L K_t \end{bmatrix} + 2\gamma_A \begin{bmatrix} K & 0 \end{bmatrix} + 2\gamma_M \begin{bmatrix} K & -K \\ -K & K \end{bmatrix} + 2\gamma_D \begin{bmatrix} K^T S \end{bmatrix} + \varepsilon I \right)^{-1} \begin{bmatrix} K^T \tilde{Y}_s \end{bmatrix} \beta^*
$$

(4.13)

where $\varepsilon I$ is also added to obtain numerically stable solution. For multi-class classification problem, one-against-all strategy on mtUDA-SVM method is employed. Similar to mtUDA-RLS and [12], each trade-off coefficient is treated as a whole, e.g. $\hat{\gamma}_I = \frac{\gamma_I}{n_t^2}$, $\hat{\gamma}_A = \gamma_A$, $\hat{\gamma}_M = \gamma_M$, and $\hat{\gamma}_D = \gamma_D$.

The pseudo code of our algorithm is summarised in Algorithm 2. For easily tuning the parameters, $M$ and $L$ matrices are normalized by the same method as in [176, 12].

**Algorithm 2: multi-task learning-based Unsupervised Domain Adaptation**

**Input:** Data and source labels: $X_s, X_t, Y_s$; Parameters: $p$, $\hat{\gamma}_I$, $\hat{\gamma}_A$, $\hat{\gamma}_M$, $\hat{\gamma}_D$.

**Output:** Target domain output function: $f_t$.

1. Initialize pseudo labels in target domain $\tilde{Y}_t$ using a base classifier trained on original source domain data;
2. Compute kernel matrices $K_s$, $K_t$, and $K$ by choosing a kernel function $K(x, y)$;
3. Construct $M$ according to (4.2.1), and $L = D - W$, then normalize $M \leftarrow \frac{M}{\|M\|_F}$, and $L \leftarrow D^{-1/2}L D^{-1/2}$;
4. repeat
   - Compute $\alpha$ according to (4.10) for mtUDA-RLS, or using (4.12) and (4.13) for mtUDA-SVM;
   - Use the learnt adaptive classifier $\alpha_t$ to update pseudo labels $\tilde{Y}_t$ in target domain;
   - Update $M$ according to (4.2.1) and Normalize $M \leftarrow \frac{M}{\|M\|_F}$.
5. until Convergence;
6. Obtain the final target domain output function $f^*_t(x_t) = \sum_{i=1}^{n_s+n_t} \alpha^*_t K(x_t, x_i)$.

### 4.3 Experiments

**Datasets** The proposed mtUDA methods were evaluated on a synthetic dataset, real-world digit recognition datasets and object recognition datasets. The synthetic data has two classes. The two classes of source data are generated by sampling from Gaussians centered at (0,2), and (2,0), while the two classes of target data are sampled from Gaussians centered at (-1,-1), and (2,0). For cross-domain hand-written digit recognition, MNIST [150] and USPS [116] datasets were used. Data released by [177, 176] were used to construct a pair of cross-domain
datasets USPS v.s. MNIST by randomly sampling 1,800 images in USPS and 2,000 images in MNIST. All images were uniformly rescaled to size $16 \times 16$, and each image is represented by a feature vector encoding the gray-scale pixel values.

For object recognition, the methods were evaluated on two different datasets. One is Office-31 dataset studied by Saenko et al. [224], which contains three different domains: Amazon (images downloaded from online merchants), Webcam (low-resolution images by a web camera), DSLR (high-resolution images by a digital SLR camera). There are 31 classes of the object shared by the three domains, forming 6 pairs of cross-domain tasks. Decaf$_7$ features [58] were used. The other object dataset is the Office-Caltech-10 dataset released by Gong et al. [93]. This dataset is built upon the Office-31 dataset and contains images from four different domains: Amazon, Webcam, DSLR, and Caltech-256, where Caltech-256 [102] contains 256 object classes downloaded from Google images. Ten classes common to four datasets are selected to form 12 pairs of datasets. Two types of features are considered: Decaf$_7$ features [58] and SURF descriptors $^b$.

**Baselines and Settings** The proposed methods are compared with the state-of-the-art unsupervised domain adaptation method, namely SA [74], JDA [177], JGSA [313], ARTL [176] (includes ARRLS and ARSVM), and RTN [180]. SA, JDA, and JGSA are feature transformation-based methods, ARTL is a classifier-based method, DAN, RTN, and WDAN are deep learning based methods. For deep learning based methods, AlexNet [138] is used as the base model for fair comparison. The proposed methods are also compared with the source only baseline, which is the results obtained by the Nearest Neighbour classifier on the source domain data without adaptation. For all the compared methods, the parameters recommended by the original papers were used. For the proposed methods, $\gamma_M = 1$, $\gamma_A = 0.1$, $\gamma_I = 1$, $p = 5$, and 1) $\gamma_D = 10$ for Office-31; 2) $\gamma_D = 1$ for other datasets. In fact, the proposed methods perform well on a wide range of parameter values based on the empirical results. The number of iterations is fixed to 10 since the algorithms generally converge within 10 iterations. All algorithms were evaluated in a fully transductive setup [90].

### 4.3.1 Results

Figure 4.1 shows the comparison between ARRLS [176] and the proposed mtUDA-RLS methods with linear and gauss kernels. The green shade represents the adaptive classifier. The ARRLS method tries to learn a unified classifier that can perform well on both domains, which does not exist or is hard to find when the domain shift is large. Since the goal of the adaptive classifier is to classify the target samples, it is not necessary to perform
Figure 4.1: Comparisons of ARRLS and the proposed mtUDA-RLS on the synthetic data.

well on both domains. By contrast, our mtUDA-RLS method jointly learns two different but related classifiers for source and target domain respectively. Without the unified classifier constraint, the target domain adaptive classifier obtained by our mtUDA-RLS method performs much better than ARRLS on the target task.

For the real world datasets, the comparison results in Tables 4.1, 4.2, 4.3, 4.4 show that the proposed method outperforms the state-of-the-art domain adaptation methods on most of the datasets. Based on the results, It is observed that all the domain adaptation methods outperform the No Adaptation results, which means that the domain shift indeed exists on these datasets and both feature transformation-based methods and classifier-based methods can reduce the shift to different degrees. Secondly, the classifier-based methods perform better than feature transformation-based methods in general on the evaluated datasets, which verifies that the two-step solution in the feature transformation-based methods may not be optimal. Thirdly, the proposed mtUDA methods outperform both ARTL and the deep-learning-based methods (i.e. DAN, RTN, WDAN). As analysed before, ARTL assumes shared classifier between domains, which may not exist. In addition, compared to ARTL, the good results obtained by mtUDA are also due to the application of manifold regularization to target data only. Compared to the deep-learning-based meth-
Table 4.1: Accuracy on the digit dataset.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>MNIST → USPS</th>
<th>USPS→ MNIST</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Only</td>
<td>65.9</td>
<td>44.7</td>
<td>55.3</td>
</tr>
<tr>
<td>SA [74]</td>
<td>67.8</td>
<td>48.8</td>
<td>58.3</td>
</tr>
<tr>
<td>JDA [177]</td>
<td>67.3</td>
<td>59.7</td>
<td>63.5</td>
</tr>
<tr>
<td>JGSA [313]</td>
<td>80.4</td>
<td>68.2</td>
<td>74.3</td>
</tr>
<tr>
<td>AR SVM linear [176]</td>
<td>67.8</td>
<td>57.2</td>
<td>62.5</td>
</tr>
<tr>
<td>AR SVM Gaussian [176]</td>
<td>88.2</td>
<td>64.4</td>
<td>76.3</td>
</tr>
<tr>
<td>AR RLS linear [176]</td>
<td>67.2</td>
<td>52.1</td>
<td>59.6</td>
</tr>
<tr>
<td>AR RLS Gaussian [176]</td>
<td>88.8</td>
<td>67.7</td>
<td>78.2</td>
</tr>
<tr>
<td>mtUDA-SVM linear</td>
<td>75.7</td>
<td>62.6</td>
<td>69.2</td>
</tr>
<tr>
<td>mtUDA-SVM Gaussian</td>
<td>88.3</td>
<td>68.0</td>
<td>78.1</td>
</tr>
<tr>
<td>mtUDA-RLS linear</td>
<td>70.3</td>
<td>51.4</td>
<td>60.8</td>
</tr>
<tr>
<td>mtUDA-RLS Gaussian</td>
<td><strong>89.2</strong></td>
<td><strong>71.9</strong></td>
<td><strong>80.5</strong></td>
</tr>
</tbody>
</table>

Table 4.2: Accuracy on the Office-31 dataset.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>A→W</th>
<th>A→D</th>
<th>W→A</th>
<th>W→D</th>
<th>D→A</th>
<th>D→W</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Adaptation</td>
<td>55.6</td>
<td>59.2</td>
<td>41.8</td>
<td>98.2</td>
<td>44.9</td>
<td>93.1</td>
<td>65.5</td>
</tr>
<tr>
<td>SA [74]</td>
<td>55.4</td>
<td>58.4</td>
<td>44.0</td>
<td>98.6</td>
<td>46.5</td>
<td>92.7</td>
<td>65.9</td>
</tr>
<tr>
<td>JDA [177]</td>
<td>56.9</td>
<td>56.6</td>
<td>45.0</td>
<td>98.0</td>
<td>47.0</td>
<td>94.1</td>
<td>66.3</td>
</tr>
<tr>
<td>JGSA [313]</td>
<td>60.4</td>
<td>65.3</td>
<td>54.0</td>
<td>97.2</td>
<td>51.5</td>
<td>96.4</td>
<td>70.8</td>
</tr>
<tr>
<td>AR SVM linear [176]</td>
<td>63.4</td>
<td>61.5</td>
<td>50.8</td>
<td>90.8</td>
<td>55.9</td>
<td>59.1</td>
<td>69.0</td>
</tr>
<tr>
<td>AR SVM Gaussian [176]</td>
<td>65.5</td>
<td>66.9</td>
<td>57.1</td>
<td>92.4</td>
<td>60.7</td>
<td>93.5</td>
<td>72.7</td>
</tr>
<tr>
<td>AR RLS linear [176]</td>
<td>66.0</td>
<td>69.7</td>
<td>57.4</td>
<td>97.8</td>
<td>58.5</td>
<td>94.5</td>
<td>74.0</td>
</tr>
<tr>
<td>AR RLS Gaussian [176]</td>
<td>68.3</td>
<td>70.3</td>
<td><strong>60.2</strong></td>
<td>98.4</td>
<td>61.5</td>
<td>95.2</td>
<td>75.6</td>
</tr>
<tr>
<td>DAN [174]</td>
<td>68.5</td>
<td>67.0</td>
<td>53.1</td>
<td>99.0</td>
<td>54.0</td>
<td>96.0</td>
<td>72.9</td>
</tr>
<tr>
<td>RTN [180]</td>
<td>70.2</td>
<td>69.3</td>
<td>51.0</td>
<td><strong>99.3</strong></td>
<td>50.5</td>
<td><strong>96.8</strong></td>
<td>72.9</td>
</tr>
<tr>
<td>WDAN [300]</td>
<td>66.8</td>
<td>64.5</td>
<td>52.7</td>
<td>98.7</td>
<td>53.8</td>
<td>95.9</td>
<td>72.1</td>
</tr>
<tr>
<td>mtUDA-SVM linear</td>
<td>67.4</td>
<td>65.9</td>
<td>54.6</td>
<td>88.6</td>
<td>59.9</td>
<td>88.8</td>
<td>70.9</td>
</tr>
<tr>
<td>mtUDA-SVM Gaussian</td>
<td>68.1</td>
<td>66.7</td>
<td>57.0</td>
<td>90.6</td>
<td>62.2</td>
<td>92.8</td>
<td>72.9</td>
</tr>
<tr>
<td>mtUDA-RLS linear</td>
<td>69.2</td>
<td>72.9</td>
<td>58.0</td>
<td>98.4</td>
<td>61.4</td>
<td>95.4</td>
<td>75.9</td>
</tr>
<tr>
<td>mtUDA-RLS Gaussian</td>
<td><strong>70.4</strong></td>
<td><strong>73.1</strong></td>
<td>58.8</td>
<td><strong>98.4</strong></td>
<td><strong>63.1</strong></td>
<td>96.0</td>
<td><strong>76.6</strong></td>
</tr>
</tbody>
</table>

ods, better performance can be obtained with much lower computational cost. Lastly, the comparison between the linear and Gaussian kernel versions of the proposed algorithms have shown that, for the digit recognition tasks, the Gaussian kernel outperforms linear kernel to a large degree, but the differences are not obvious on the object recognition tasks.

To visually illustrate the advantages by jointly optimizing two classifiers in the proposed method, the t-SNE visualization of the activations of the target classifier by the target samples using mtRLS and ARRLS methods is shown in Figure 4.2. Taking the challenging C→W task with SURF features as an example, it can be seen that the mtRLS method with two classifiers can obtain better classification results (i.e. different classes
Table 4.3: Accuracy on the Office-Caltech-10 dataset with decaf7 features.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No Adaptation</td>
<td>90.5</td>
<td>77.6</td>
<td>82.2</td>
<td>83.6</td>
<td>74.2</td>
<td>86.0</td>
<td>77.4</td>
<td>80.2</td>
<td>100.0</td>
<td>79.5</td>
<td>86.9</td>
<td>99.7</td>
<td>84.8</td>
</tr>
<tr>
<td>SA [74]</td>
<td>90.0</td>
<td>85.8</td>
<td>86.0</td>
<td>80.0</td>
<td>86.4</td>
<td>88.5</td>
<td>79.6</td>
<td>85.2</td>
<td>99.4</td>
<td>77.9</td>
<td>87.5</td>
<td>95.3</td>
<td>86.8</td>
</tr>
<tr>
<td>JDA [177]</td>
<td>89.6</td>
<td>78.6</td>
<td>87.3</td>
<td>81.8</td>
<td>77.3</td>
<td>83.4</td>
<td>83.1</td>
<td>90.4</td>
<td>100.0</td>
<td>82.6</td>
<td>91.7</td>
<td>98.3</td>
<td>87.0</td>
</tr>
<tr>
<td>JGSA [313]</td>
<td>90.9</td>
<td>86.8</td>
<td>89.8</td>
<td>85.6</td>
<td>84.1</td>
<td>88.5</td>
<td>87.4</td>
<td>92.0</td>
<td>100.0</td>
<td>86.6</td>
<td>92.8</td>
<td>99.0</td>
<td>90.3</td>
</tr>
<tr>
<td>AR SVM linear [176]</td>
<td>92.7</td>
<td>94.9</td>
<td>93.6</td>
<td>88.4</td>
<td>91.2</td>
<td>89.2</td>
<td>87.2</td>
<td>92.8</td>
<td>99.4</td>
<td>86.9</td>
<td>93.7</td>
<td>96.3</td>
<td>92.2</td>
</tr>
<tr>
<td>AR SVM Gaussian [176]</td>
<td>92.6</td>
<td>95.6</td>
<td>95.6</td>
<td>88.3</td>
<td>90.9</td>
<td>89.2</td>
<td>87.2</td>
<td>92.8</td>
<td>99.4</td>
<td>87.0</td>
<td>93.2</td>
<td>96.3</td>
<td>92.3</td>
</tr>
<tr>
<td>ARR S linear [176]</td>
<td>92.9</td>
<td>95.3</td>
<td>93.0</td>
<td>88.1</td>
<td>90.2</td>
<td>89.8</td>
<td>87.1</td>
<td>92.7</td>
<td>99.4</td>
<td>87.7</td>
<td>93.6</td>
<td>99.0</td>
<td>92.4</td>
</tr>
<tr>
<td>ARR S Gaussian [176]</td>
<td>92.9</td>
<td>94.9</td>
<td>93.6</td>
<td>87.9</td>
<td>90.5</td>
<td>89.8</td>
<td>87.0</td>
<td>92.7</td>
<td>99.4</td>
<td>87.5</td>
<td>93.5</td>
<td>99.0</td>
<td>92.4</td>
</tr>
<tr>
<td>WDAN (AlexNet) [300]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>92.4</td>
</tr>
</tbody>
</table>

Table 4.4: Accuracy on the Office-Caltech-10 dataset with SURF features.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No Adaptation</td>
<td>36.0</td>
<td>29.2</td>
<td>38.2</td>
<td>34.2</td>
<td>31.2</td>
<td>35.7</td>
<td>28.8</td>
<td>31.6</td>
<td>84.7</td>
<td>29.6</td>
<td>28.3</td>
<td>83.7</td>
<td>40.9</td>
</tr>
<tr>
<td>SA [74]</td>
<td>49.3</td>
<td>40.0</td>
<td>39.5</td>
<td>40.0</td>
<td>33.2</td>
<td>33.8</td>
<td>35.2</td>
<td>39.3</td>
<td>75.2</td>
<td>34.6</td>
<td>39.9</td>
<td>77.0</td>
<td>44.7</td>
</tr>
<tr>
<td>JDA [177]</td>
<td>44.8</td>
<td>41.7</td>
<td>45.2</td>
<td>39.4</td>
<td>38.0</td>
<td>39.5</td>
<td>31.2</td>
<td>32.8</td>
<td>89.2</td>
<td>31.5</td>
<td>33.1</td>
<td>89.5</td>
<td>46.3</td>
</tr>
<tr>
<td>DIP [10]</td>
<td>51.9</td>
<td>49.5</td>
<td>50.3</td>
<td>39.3</td>
<td>39.7</td>
<td>45.8</td>
<td>38.4</td>
<td>42.8</td>
<td>70.7</td>
<td>33.8</td>
<td>40.1</td>
<td>73.2</td>
<td>48.0</td>
</tr>
<tr>
<td>JGSA [313]</td>
<td>51.5</td>
<td>45.4</td>
<td>45.9</td>
<td>41.5</td>
<td>45.8</td>
<td>47.1</td>
<td>33.2</td>
<td>39.9</td>
<td>90.5</td>
<td>29.9</td>
<td>38.0</td>
<td>91.9</td>
<td>50.0</td>
</tr>
<tr>
<td>ARR S linear [176]</td>
<td>53.4</td>
<td>43.4</td>
<td>40.8</td>
<td>38.8</td>
<td>38.0</td>
<td>38.2</td>
<td>32.3</td>
<td>38.4</td>
<td>88.5</td>
<td>31.6</td>
<td>36.3</td>
<td>89.2</td>
<td>47.4</td>
</tr>
<tr>
<td>ARR S Gaussian [176]</td>
<td>50.7</td>
<td>43.1</td>
<td>46.5</td>
<td>40.9</td>
<td>40.7</td>
<td>42.7</td>
<td>31.5</td>
<td>38.7</td>
<td>91.1</td>
<td>30.5</td>
<td>32.6</td>
<td>91.9</td>
<td>48.4</td>
</tr>
<tr>
<td>mtUDA-SVM linear</td>
<td>51.7</td>
<td>47.8</td>
<td>42.7</td>
<td>38.5</td>
<td>44.4</td>
<td>47.8</td>
<td>31.8</td>
<td>40.6</td>
<td>84.7</td>
<td>30.0</td>
<td>37.1</td>
<td>90.9</td>
<td>49.0</td>
</tr>
<tr>
<td>mtUDA-SVM Gaussian</td>
<td>55.1</td>
<td>56.3</td>
<td>50.3</td>
<td>40.0</td>
<td>50.2</td>
<td>49.0</td>
<td>32.8</td>
<td>40.6</td>
<td>86.0</td>
<td>30.5</td>
<td>35.5</td>
<td>90.9</td>
<td>51.4</td>
</tr>
</tbody>
</table>

We take the Office-Caltech-10 with SURF features as an example to compare the CPU.
time of the proposed method with previous methods. The CPU time (training and evaluation) of mtUDA-SVM is 26.9s and mtUDA-RLS is 111.7s on average over different domain pairs. Compared to the previous state-of-the-art shallow-learning-based method, ARRLS (2.43s), the average runtime of mtUDA is longer, but is much shorter than DIP (728.7s) and similar to JGSA (28.9s), however, better performance can be obtained on most of the domain pairs. Compared to the deep-learning-based methods, the runtime is much shorter and the performance is similar or better.

### 4.3.2 Parameter Sensitivity

Experiments were conducted on the C→W (Office-Caltech-10 dataset with SURF descriptor), USPS→MNIST, and A→D (Office-31 dataset with decaf7 feature) datasets to study the sensitivity of the proposed algorithms to their parameters (Figure 4.3). The solid lines are the accuracies obtained by mtUDA-RLS with gauss kernel, and the dashed lines are the results obtained by the best baseline methods on each dataset. It can be seen that a wide range of values can be chosen to obtain satisfactory performances. \( \hat{\gamma}_M \) regulates the similarity between source and target classifiers. If \( \hat{\gamma}_M \) is too small, the source class information cannot be transferred to the target domain, if \( \hat{\gamma}_M \to \infty \) the source and target classifiers tend to be the same, which may not be desirable. \( \hat{\gamma}_A \) controls the complexity of the classifiers. A small \( \hat{\gamma}_A \) would lead to overfitting while a too large \( \hat{\gamma}_A \) leads that the models cannot fit the data. \( \hat{\gamma}_D \) controls the degree of distribution shift. Though the larger \( \hat{\gamma}_D \) will lead to smaller distribution shift, a too large value will cancel out other regularizations. \( \hat{\gamma}_I \) regulates the geometry structure of the target domain. If \( \hat{\gamma}_I \) is too small, the target domain structure is not preserved, but if \( \hat{\gamma}_I \) is too large the source class information is discarded. The number of nearest neighbours \( p \) in constructing the adjacency graph determines the degree of target structure preservation. A too small \( p \) captures little structure information, while a too large \( p \) connects samples without similarity. For the number of iterations, the results show that the accuracies increase steadily with more iterations and converge to the best value within 10 iterations.

### 4.4 Summary

This chapter presents a novel multi-task learning-based unsupervised domain adaptation method. It relaxes the single classifier assumption in the conventional classifier-based unsupervised domain adaptation and proposes to jointly optimize source and target classifiers by considering the manifold structure of target domain and the distribution divergence between the domains. Experimental results on both synthetic and real-world cross domain recognition datasets have shown the effectiveness of the proposed method.
Figure 4.3: Parameter sensitivity of the proposed methods on different datasets
Chapter 5

Importance Weighted Adversarial Nets for Partial Domain Adaptation

5.1 Motivation

Since recent advances in deep learning have shown that more transferable and domain invariant features can be extracted through deep framework, the domain adaptation techniques are also transferred from shallow learning-based [13, 200, 177, 178, 93, 74, 313] to deep learning-based [263, 174, 180, 175, 310, 246, 264, 82, 262, 23, 165, 186].

The deep learning based methods have witnessed better performances compared to the shallow learning based methods. However, most of the current approaches still assume the same label spaces between the source and target domains. For example, previous deep learning-based domain adaptation methods generally follow the idea that the divergence between source and target distributions is small in the feature space and adaptation can be achieved by matching the statistic moments [263, 174, 180, 175, 310, 246], or by relying on the domain adversarial nets [264, 82, 262, 23], or by using Batch Normalization statistics [165, 186]. Since all the approaches rely on the comparison of marginal distributions between the source and target domains, the label spaces between the two domains are required to be the same for feasible adaptation. This chapter is concerned with a different and more practical scenario that the target domain only has a subset of classes, referred to as partial domain adaptation (similar to [27]). In addition, there is no labeled data in the target domain and the potential number and name of the target classes are unknown. We assume that the source domain is large and diverse to include all classes that appear in the target domain.

Intuitively, when target domain only contains a subset of classes, it is impossible to reduce the domain shift by comparing the source and target distributions directly. In other words, reducing the distribution shift will not benefit the target task, since the marginal distributions between domains should not be the same essentially due to different label
Figure 5.1: The overview of the proposed method. The green parts are the feature extractors for source and target domains. The block filled with slashes indicates the parameters are pre-learned and will not be updated during the training procedure. The blue parts are the first domain classifier for obtaining the source sample importance weights. The red parts are the second domain classifier that plays the minimax game with the weighted source domain samples and the target samples. The GRL \cite{82} indicates the gradient reversal layer, which acts as an identity transformation in forward propagation while changes the sign of the gradient in backward propagation.
spaces. In this case, a natural and possible way to transfer from the source domain to
the target domain is reweighting the source domain samples whose classes are likely to
appear in the target domain in the distribution matching procedure. However, the target
domain is unlabelled, it is not straightforward to uncover which classes are presented and
which source domain samples are important for transferring. This chapter proposes a
weighted adversarial nets-based deep domain adaptation method for such a problem.

An adversarial net based domain adaptation consists of a feature extractor and a domain
classifier. The domain classifier aims at identifying the difference between distributions
of the source and target samples to find a tighter lower bound on the true domain distri-
bution divergence, while the feature extractor, on the other hand, reduces the distribution
divergence by stepping to the opposite direction as the domain classifier. This chapter
proposes a two domain classifier strategy to identify the importance score of the source
samples. Specifically, given any feature extractor, the output of the optimum parameters
of the first domain classifier gives the probability of the sample coming from the source
domain. The intuition of the weighting scheme is that if the activation of the first domain
classifier is large, the sample can be almost perfectly discriminated from the target do-
main by the domain classifier. Thus, the sample is highly likely from the outlier classes in
the source domain, since the neighbourhood region of the sample covers little or no target
sample at all, and a small weight is assigned to the sample. Hence, we use the activations
of the first domain classifier as an indicator of the importance of each source sample to
the target domain. Then the learned weights are applied to the source samples and the
weighted source samples and target samples are fed into the second domain classifier for
optimizing the feature extractor. We have shown that the minimax two-player game be-
tween the feature extractor and the second domain classifier is theoretically equivalent
to reducing the Jensen-Shannon divergence between the weighted source density and the
target density.

The proposed methods were evaluated on three commonly used cross-domain object
datasets with the setting that the target domain has a subset of classes. The results have
shown that the proposed method outperforms previous domain adaptation methods to a
large degree and is comparable to the state-of-the-art partial transfer method.

5.2 Proposed Method

This section presents the proposed method in details. It begins with the definitions of
terminologies. The source domain data denoted as $X_s \in \mathbb{R}^{D \times n_s}$ are draw from distribution
$p_s(x)$ and the target domain data denoted as $X_t \in \mathbb{R}^{D \times n_t}$ are draw from distribution $p_t(x)$,
where $D$ is the dimension of the data instance, $n_s$ and $n_t$ are number of samples in source
and target domain respectively. We focus on the unsupervised domain adaptation problem
which assumes that there are sufficient labeled source domain data, $\mathcal{D}_s = \{(x^s_i, y^s_i)\}_{i=1}^{n_s}$,
\( x^t_j \in \mathbb{R}^D \), and unlabeled target domain data, \( \mathcal{D}_t = \{ (x^t_j)_{j=1}^n \}, x^t_j \in \mathbb{R}^D \), in the training stage. The feature spaces are assumed same: \( \mathcal{X}_s = \mathcal{X}_t \) while the target domain label space is contained in the source domain label space \( \mathcal{Y}_t \subseteq \mathcal{Y}_s \). In addition, due to the domain shift, \( p_s(x) \neq p_t(x) \) even when the label spaces between domains are the same.

### 5.2.1 Adversarial Nets-based Domain Adaptation

The works in [82, 262] apply a domain classifier on the general feed-forward models to form the adversarial nets-based domain adaptation methods. The general idea is to learn both class discriminative and domain invariant features, where the loss of the label predictor of the source data is minimized while the loss of the domain classifier is maximized. Specifically, the adversarial nets-based domain adaptation framework is similar to the original GAN with minimax loss:

\[
\min_{F_s, F_t} \max_D \mathcal{L}(D, F_s, F_t) = \mathbb{E}_{x \sim p_s(x)} [\log D(F_s(x))] + \mathbb{E}_{x \sim p_t(x)} [\log (1 - D(F_t(x)))] \tag{5.1}
\]

where \( F_s \) and \( F_t \) are the feature extractors for source and target data respectively, which can be identical [82] (shared weights) or distinct [262] (unshared weights), and \( D \) is the domain classifier. The \( D \) is a binary domain classifier (corresponding to the discriminator in original GAN) with all the source data labelled as 1 and all the target data labelled as 0. Maximizing the minimax loss with respect to the parameters of \( D \) yields a tighter lower bound on the true domain distribution divergence, while minimizing the minimax loss with respect to the parameters of \( F \) minimizes the distribution divergence in the feature space.

In this chapter, we adopt the unshared feature extractors for source and target domains to capture more domain specific features than a shared feature extractor as reported in [313, 262] and to train the source discriminative model separately. We follow a similar procedure as [262] to train the source discriminative model \( C(F_s(x)) \) for classification task by learning the parameters of the source feature extractor \( F_s(x) \) and classifier \( C \):

\[
\min_{F_s, C} \mathcal{L}_s = \mathbb{E}_{x,y \sim p_s(x,y)} L(C(F_s(x)), y) \tag{5.2}
\]

where \( L \) is the empirical loss for source domain classification task and the cross entropy loss is used in this chapter.

Given the learned \( F_s(x) \), a domain adversarial loss is used to reduce the shift between
domains by optimizing $F_t(x)$ and $D$:

$$
\min_{F_t} \max_D \mathcal{L}(D, F_s, F_t) = \mathbb{E}_{x \sim p_s(x)}[\log D(F_s(x))] 
+ \mathbb{E}_{x \sim p_t(x)}[\log (1 - D(F_t(x)))] \tag{5.3}
$$

To avoid a degenerate solution, we initialize $F_t$ using the parameter of $F_s$ by following [262].

Given $F_s(x)$ (corresponding to real images in GAN), for any $F_t(x)$ (corresponding to generated images in GAN), the optimum $D$ is obtained at:

$$
D^*(z) = \frac{p_s(z)}{p_s(z) + p_t(z)} \tag{5.4}
$$

where $z = F(x)$ is the sample in the feature space after feature extraction networks. Similar to [96], we give the proof of Equation 5.4 as follows.

**Proof 1** For any $F_s(x)$ and $F_t(x)$, the training criterion for the domain classifier $D$ is to maximize Equation 5.3:

$$
\max_D \mathcal{L}(D, F_s, F_t) = \int_x p_s(x) \log D(F_s(x)) 
+ p_t(x) \log (1 - D(F_t(x))) dx 
= \int_z p_s(z) \log D(z) 
+ p_t(z) \log (1 - D(z)) dz \tag{5.5}
$$

We take the partial differential of the objective 5.5 with respect to $D$, $\frac{\partial \mathcal{L}(D, F)}{\partial D}$, and achieves its maximum in $[0, 1]$ at 5.4, where the Leibnizs rule is used to exchange the order of differentiation and integration.

### 5.2.2 Importance Weighted Adversarial Nets-based Domain Adaptation

**Sample weights learning** In the minimax game of Equation 5.3, the domain classifier is given by

$$
D(z) = p(y = 1|z) = \sigma(a(z)) \tag{5.6}
$$

where $\sigma$ is the logistic sigmoid function. Suppose that the domain classifier has converged to its optimal value for the current feature extractor, the output value of the domain classifier gives the likelihood of the sample coming from source distribution. Thus, if the $D^*(z) \approx 1$, then the sample is highly likely from the outlier classes in the source domain, since the region that covers the sample has little or no target sample at all and can be
almost perfectly discriminated from target distribution by the domain classifier. The contribution of these samples should be small such that both the domain classifier and feature extractor will ignore them. On the other hand, if \( D^*(z) \) is small, the sample is more likely from the shared classes between domains. These samples should be given a larger importance weight to reduce the domain shift on the shared classes. Hence, the weight function should be inversely related to \( D^*(z) \) and a natural way to define the importance weights function of the source samples is:

\[
\tilde{w}(z) = 1 - D^*(z) = \frac{1}{\frac{p_s(z)}{p_t(z)} + 1} \quad (5.7)
\]

It can be seen that if \( D^*(z) \) is large, \( \tilde{w}(z) \) is small and thus \( \frac{p_s(z)}{p_t(z)} \) is large. Hence, the weights for the source samples from the outlier classes will be smaller than the shared class samples. Note that the weights function is also a function of density ratio between source and target features, which further verifies the reasonableness of the weights function, since the neighbourhood region of the sample that covers little or no target sample will be assigned a small weight. Our purpose is to obtain the relative importance of source samples, suggesting that the samples from outlier classes should be assigned a relatively small weight than the samples from the shared classes. Hence, the weights are normalized as follows

\[
w(z) = \frac{\tilde{w}(z)}{\mathbb{E}_{z \sim p_s(z)} \tilde{w}(z)} \quad (5.8)
\]

such that \( \mathbb{E}_{z \sim p_s(z)} w(z_i) = 1 \). Note that the weights are defined as a function of the domain classifier. Thus if we apply the weights on the same domain classifier, the theoretical results of the minimax game will not be reducing the Jensen-Shannon divergence between two densities (since the optimum domain classifier (e.g. Equation 5.4) will not be the ratio between the source density and the sum of the source and target densities due to the introducing of the weight function which is also a function of \( D \)). Hence, we propose to solve this issue by applying the second domain classifier on the extracted features, namely \( D_0 \), for comparing the weighted source data and the target data. In this way, the first domain classifier \( D \) is only used for obtaining the importance weights for the source domain based on \( F_s \) and the current \( F_t \). Thus, the gradient of \( D \) will not be back-propagated for updating \( F_t \), since the gradients of \( D \) are learned on unweighted source samples and would not be a good indicator for reducing domain shift on the shared classes. After all, it is \( D_0 \) (with the weighted source data and the target data) who plays the minimax game with \( F_t \) to reduce the shift on the shared classes.

After adding the importance weights to the source samples for the domain classifier \( D_0 \),
the objective function of weighted domain adversarial nets $\mathcal{L}_w(D_0, F)$ is:

$$\min_{F_t} \max_{D_0} \mathcal{L}_w(D_0, F_s, F_t) = \mathbb{E}_{x \sim p_s(x)} [w(z) \log D_0 (F_s(x))]$$

$$+ \mathbb{E}_{x \sim p_t(x)} [\log(1 - D_0 (F_t(x)))] \quad (5.9)$$

where the $w(z)$, as a function of $D$, is independent of $D_0$ and can be seen as a constant. Thus, given $F_s$ and $D$, for any $F_t$, the optimum $D_0$ of the weighted adversarial nets is obtained at:

$$D_0^*(z) = \frac{w(z)p_s(z)}{w(z)p_s(z) + p_t(z)} \quad (5.10)$$

Note that since we normalized the importance weights $w(z)$, the $w(z)p_s(z)$ is still a probability density function:

$$\mathbb{E}_{z \sim p_s(z)} w(z) = \int w(z)p_s(z)dz = 1 \quad (5.11)$$

Given the optimum $D_0$, the minimax game of 5.9 can be reformulated as:

$$\mathcal{L}_w(F_t) = \mathbb{E}_{x \sim p_s(x)} [w(z) \log D_0^*(F_s(x))]$$

$$+ \mathbb{E}_{x \sim p_t(x)} [\log(1 - D_0^*(F_t(x)))]$$

$$= \int w(z)p_s(z) \log \frac{w(z)p_s(z)}{w(z)p_s(z) + p_t(z)} \quad (5.12)$$

$$+ p_t(z) \log \frac{p_t(z)}{w(z)p_s(z) + p_t(z)} dz$$

$$= - \log(4) + 2 \cdot JS(w(z)p_s(z) || p_t(z))$$

Hence, the weighted adversarial nets-based domain adaptation is essentially reducing the Jensen-Shannon divergence between the weighted source density and the target density in the feature space, which obtains it’s optimum on $w(z)p_s(z) = p_t(z)$.

**Target data structure preservation** Since the target domain does not have labels, it is important to preserve the data structure for effective transfer. If the shift between the weighted source distribution and target distribution in the feature space is small, the classifier $C$ learned from the source data can be directly used for the target domain. Here, we further constrain $F_t$ by employing the entropy minimization principle [99] to encourage the low-density separation between classes:

$$\min_{F_t} \mathbb{E}_{x \sim p_t(x)} H(C(F_t(x))) \quad (5.13)$$

where $H(\cdot)$ is the information entropy function. Since the source classifier $C$ is directly applied to the adapted target features, the target entropy minimization is only used to constrain $F_t$, which is different from previous usage [180, 27]. We argue that if target
entropy minimization is applied on both feature extractor and classifier as in [180, 27], a
side effect is that the target samples may easily be stuck into a wrong class due to the large
domain shift in the early stage of training and hard to be corrected later on. By contrast,
if target entropy minimization is only used to constrain $F_t$, it will reduce the side effect.

**Overall objective function** Hence, the overall objectives of the weighted adversarial
nets-based method are:

$$
\min_{F_s, C} \mathcal{L}_s(F_s, C) = -\mathbb{E}_{x, y \sim p_s(x, y)} \sum_{k=1}^K \mathbb{I}_{[k=y]} \log C(F_s(x))
$$

$$
\min_D \mathcal{L}_D(D, F_s, F_t) = -\left(\mathbb{E}_{x \sim p_s(x)} [\log D(F_s(x))] + \mathbb{E}_{x \sim p_t(x)} [\log (1 - D(F_t(x)))]\right)
$$

$$
\min_{F_t} \max_{D_0} \mathcal{L}_w(C, D_0, F_s, F_t) =
$$

$$
\gamma \mathbb{E}_{x \sim p_t(x)} H(C(F_t(x))) + \lambda \left(\mathbb{E}_{x \sim p_s(x)} [w(z) \log D_0(F_s(x))] + \mathbb{E}_{x \sim p_t(x)} [\log (1 - D_0(F_t(x)))]\right)
$$

where $\lambda$ is the tradeoff parameter. The objectives are optimized in stages. $F_s$ and $C$
are pre-trained on the source domain data and fixed afterwards. Then the $D$, $D_0$ and $F_t$
are optimized simultaneously without the need of revisiting $F_s$ and $C$. Note that $D$ is only
used for obtaining the importance weights for the source domain using $F_s$ and current $F_t$,
while $D_0$ plays the minimax game with the target domain feature extractor for updating
$F_t$. To solve the minimax game between $F_t$ and $D_0$, we can either iteratively train the
two objectives respectively similar to GAN, or insert a gradient reversal layer (GRL) [82]
to multiply the gradient by -1 for the feature extractor to learn the feature extractor and
domain classifier simultaneously. In this chapter, we choose to use the GRL for solving
the problem for the fair comparison with previous methods. The proposed architecture
can be found in Figure 5.1.

## 5.3 Experiments

### 5.3.1 Set-ups

**Datasets** The proposed method is evaluated on three commonly used real-world cross-
domain object recognition datasets. The public Office+Caltech-10 object datasets re-
leased by Gong et al. [93] contains four different domains: Amazon (images downloaded
from online merchants), Webcam (low-resolution images by a web camera), DSLR (high-
resolution images by a digital SLR camera), and Caltech-256 [102], where the first three
CHAPTER 5. PARTIAL DOMAIN ADAPTATION

domains come from Office-31 [224]. Ten shared classes of the four domains form the Office+Caltech-10 dataset. Figure 5.2 shows the sample images of the four different domains. When a domain is used as the target domain, the first five classes are selected. We denote the source domains with 10 classes as A10, W10, D10, and C10, while the target domains with 5 classes are denoted as A5, W5, D5, and C5.

We also evaluate our method on the Office-31 dataset studied by Saenko et al. [224], which consists of three different domains: Amazon, DSLR, and Webcam. Compared to Office+Caltech-10, more classes (31 classes) are involved. We follow the experimental setting of [27] to transfer from one domain with the 31 categories to another domain with 10 categories (which are the shared classes between Office31 and Caltech-256 [102]). Hence, the three source domains are denoted as A31, W31, and D31, and the three target domains are denoted as A10, W10, and D10.

To evaluate on the larger scale datasets, we conducted the experiments on three pairs of domains formed by Caltech256→Office10 datasets, where the source domain is the Caltech-256 dataset with 256 classes and the target domains are three Office domains with 10 shared classes (denoted as Office-10) between Caltech-256 and Office-31.

Figure 5.2: Sample images of Caltech and Office datasets.

Baselines and Settings. The proposed method is compared with the baseline that fine-tuning the CNN using source data only (AlexNet+bottleneck) and several state-of-the-art deep learning-base domain adaptation methods: RevGrad [82], RTN [180], ADDA-grl [262], and SAN [27]. Note that ADDA-grl is a variant of the original ADDA [262] method, where the minimax game is not trained iteratively but using the GRL layer as done in our method for fair comparison. Thus the ADDA-grl can be seen as a special case of our method without the weighting scheme.

Since the cross-domain datasets are relatively small, to successfully train a deep network, we finetune the AlexNet pre-trained on ImageNet similar to previous deep learning-based domain adaptation methods [82, 180]. For the fair comparison, the same network architectures as the RevGrad method [82] are used for feature extractors and domain classifiers. Specifically, the feature extractors are the AlexNet without fc8 layer, and an additional bottleneck layer is added to fc7 layer with the dimension of 256. The two
domain classifiers are with the same architecture, which is three fully connected layers (1024→1024→1) attached to the bottleneck layer. The $F_s$ is obtained from the source domain data by finetuning the AlexNet+bottleneck.

To avoid the noisy signal at the early stage of training procedure, we use similar schedule method as [82] for the tradeoff parameter to update $F_t$ by initializing it at 0 and gradually increasing to a pre-defined upper bound. The schedule is defined as:

$$\lambda = \frac{2u}{1+\exp(-\alpha \cdot p)} - u,$$

where $p$ is the training progress linearly changing from 0 to 1, $\alpha = 1$, and $u$ is the upper bound set to 0.1 in our experiments.

5.3.2 Results and Analysis

**Evaluation of partial domain adaptation** Table 5.1, Table 5.2 and Table 5.3 show the results of the proposed methods compared with the baseline methods, where the results of SAN methods are directly copied from the original paper [27]. The proposed ($\gamma = 0$) in Table 5.1 and Table 5.2 is the variant of the proposed method without the target domain entropy minimization term. The results show that the proposed methods outperform AlexNet+bottleneck, RevGrad, RTN, and ADDA-grl to a large degree, and also comparable to the state-of-the-art partial domain adaptation method SAN on most of the datasets.

We also illustrate the A31→W10 data activations of the bottleneck layer for AlexNet+bottleneck, RevGrad, RTN, ADDA-grl, and the proposed method in Figure 5.4, where the red dots (outlier classes) and green dots (shared classes) indicate the source domain samples while the blue dots represent the target samples. The alignment is effective if the blue dots are well aligned with green dots. It shows that our method can effectively match the target classes into the relevant source domain classes compared to the baseline methods.

The RevGrad is an adversarial nets-based method with the domain classifier as a regularization for the source domain classification task. Since the adversarial training procedure only reduces the marginal distributions between domains without considering the conditional distributions, the RevGrad method obtains even much poorer results than the AlexNet+bottleneck baseline on most of the domain pairs in both datasets. Figure 5.4b also verifies that though the target domain only contains ten classes, the samples will spread to all the 31 classes in the source domain. Instead of using the adversarial loss, the RTN method reduces the domain shift based on MMD criterion. In addition, the unshared classifiers for source and target domains are proposed using a residual block and the target domain entropy minimization is applied for preserving the target domain structure. Figure 5.4c shows that the target samples are not spread to all the classes as in RevGrad due to the target domain structure preservation term. However, the RTN still performs unsatisfied for target domain classification task and the negative transfer can also be seen. Thus, though the residual nets and the target entropy minimization are in-
The ADDA-grl can be seen as the unweighted version of our method. For the fair comparison, we use exactly the same sets of parameters for ADDA-grl and our method. The results show that the proposed method outperforms ADDA-grl on most of the domain pairs. Thus the proposed weighting scheme can effectively detect the outlier classes and reduce the shift between the shared classes. Figure 5.4d and Figure 5.4e compares the activations of the two methods. The target data in the proposed method is better aligned with the selected source classes than in ADDA-grl.

The SAN methods have the same assumptions and perform comparably to the proposed method. However, a large number of domain classifiers are required in SAN compared to our method (i.e. the number of source classes v.s. two), which leads to far more parameters to train in SAN. The SAN-entropy is the SAN method without the target entropy minimization term, which corresponds to the proposed method with $\gamma = 0$. The results in Table 5.2 show that the proposed method ($\gamma = 0$) obtains better performance (86.73%) on average than that of SAN (85.64%), with a much smaller number of parameters.

**Further analysis and evaluations** For further verifying the effectiveness of the proposed weighting scheme, we also illustrate the alignment of the source and target class labels in Figure 5.5. The same activations are used as in Figure 5.4. The ten classes in the target domain are labeled as 0∼9 in blue which are the same set of classes as 0∼9 in red in the source domain. Thus the number 10∼30 in red are the outlier classes in the source domain. It shows that most of the target classes are aligned with the correct source classes. Figure 5.6 shows the learned weights using the first domain classifier $D_0$. If the weight of a source sample is large, the color of the sample tends to red while a smaller weight will be assigned with the blue color. The intermediate values are arranged based on the color bar. It can be seen that most of the red coloured samples are from 0∼9 classes while the outlier classes are mostly blue, demonstrating the effectiveness of the proposed weighting scheme for identifying samples from the outlier source classes.

We also conduct the experiments on evaluating the performance when the number of target domain classes varies. Figure 5.3 shows the results on A→W domain pair. The source domain has always 31 classes, but the number of target domain classes varies from 31 to 5, i.e. $\{31, 25, 20, 15, 10, 5\}$. The results show that the proposed method outperforms the AlexNets+bottleneck baseline largely in all cases. Specifically, when the number of target classes is getting smaller, the relative improvement is larger. It can also be observed that the less the target classes are, the lower the accuracy will be for the ADDA-grl method. Thus, when the number of target domain classes is unknown, our method can improve the performance consistently.

To evaluate the proposed method on the traditional non-partial domain adaptation set-
Table 5.1: The accuracy (%) obtained on the Office-Caltech cross-domain object dataset

| Datasets               | C10→A5 | C10→W5 | C10→D5 | A10→C5 | A10→W5 | A10→D5 | A10→C5 | A10→W5 | A10→D5 | W10→C5 | W10→W5 | W10→D5 | W10→C5 | W10→W5 | W10→D5 | D10→C5 | D10→W5 | D10→D5 | D10→C5 | D10→W5 | D10→D5 | Avg.  |
|------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|------|
| AlexNet+bottleneck     | 93.58  | 83.70  | 91.18  | 85.27  | 76.30  | 85.29  | 74.14  | 87.37  | 100.00 | 80.82  | 89.51  | 98.52  | 87.14  | 93.85  |
| RevGrad [82]           | 91.86  | 82.22  | 83.82  | 77.57  | 65.93  | 80.88  | 72.60  | 80.30  | 95.59  | 69.35  | 77.09  | 80.74  | 79.83  |
| RTN [180]              | 91.86  | 93.33  | 80.88  | 80.99  | 69.63  | 70.59  | 59.08  | 74.73  | 100.00 | 59.08  | 70.02  | 91.11  | 78.44  |
| ADDA-grl [262]         | 93.15  | 94.07  | 97.06  | 85.27  | 87.41  | 89.71  | 86.82  | 92.08  | 100.00 | 89.90  | 93.79  | 98.52  | 92.31  |
| Proposed ($\gamma = 0$) | 94.00  | **99.26** | 95.59  | **90.41** | **87.41** | 88.24  | 90.07  | **95.29** | 100.00 | 91.44  | 94.43  | 98.52  | 93.72  |
| proposed               | **94.22** | 97.78  | **98.53** | 89.90  | **87.41** | 88.24  | **90.24** | **95.29** | 100.00 | **91.61** | **94.43** | **98.52** | **93.85** |
Table 5.2: The accuracy (%) obtained on the Office cross-domain object dataset

<table>
<thead>
<tr>
<th>Datasets</th>
<th>A31→W10</th>
<th>D31→W10</th>
<th>W31→D10</th>
<th>A31→D10</th>
<th>D31→A10</th>
<th>W31→A10</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet+bottleneck</td>
<td>62.03</td>
<td>95.25</td>
<td>97.45</td>
<td>71.97</td>
<td>68.27</td>
<td>62.94</td>
<td>76.32</td>
</tr>
<tr>
<td>RevGrad [82]</td>
<td>56.95</td>
<td>75.59</td>
<td>89.17</td>
<td>57.32</td>
<td>57.62</td>
<td>63.15</td>
<td>66.64</td>
</tr>
<tr>
<td>RTN [180]</td>
<td>68.14</td>
<td>91.53</td>
<td>98.09</td>
<td>69.43</td>
<td>68.27</td>
<td>77.35</td>
<td>78.80</td>
</tr>
<tr>
<td>ADDA-grl [262]</td>
<td>63.39</td>
<td>98.31</td>
<td>98.73</td>
<td>73.25</td>
<td>70.46</td>
<td>72.34</td>
<td>79.41</td>
</tr>
<tr>
<td>SAN-selective [27]</td>
<td>71.51</td>
<td>98.31</td>
<td>100.00</td>
<td>78.34</td>
<td>77.87</td>
<td>76.32</td>
<td>83.73</td>
</tr>
<tr>
<td>SAN-entropy [27]</td>
<td>74.61</td>
<td>98.31</td>
<td>100.00</td>
<td>80.29</td>
<td>78.39</td>
<td>82.25</td>
<td>85.64</td>
</tr>
<tr>
<td>SAN [27]</td>
<td><strong>80.02</strong></td>
<td>98.64</td>
<td><strong>100.00</strong></td>
<td><strong>81.28</strong></td>
<td>80.58</td>
<td><strong>83.09</strong></td>
<td>87.27</td>
</tr>
<tr>
<td>proposed (γ = 0)</td>
<td>75.25</td>
<td><strong>98.98</strong></td>
<td><strong>100.00</strong></td>
<td>80.25</td>
<td>84.66</td>
<td>81.21</td>
<td>86.73</td>
</tr>
<tr>
<td>proposed</td>
<td>76.27</td>
<td><strong>98.98</strong></td>
<td><strong>100.00</strong></td>
<td>78.98</td>
<td><strong>89.46</strong></td>
<td>81.73</td>
<td><strong>87.57</strong></td>
</tr>
</tbody>
</table>

Table 5.3: Average accuracies (%) on Caltech256→Office10

<table>
<thead>
<tr>
<th>Methods</th>
<th>Alex</th>
<th>RevGrad[82]</th>
<th>RTN[180]</th>
<th>SAN[27]</th>
<th>proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>49.86</td>
<td>61.80</td>
<td>71.56</td>
<td><strong>85.83</strong></td>
<td>84.14</td>
</tr>
</tbody>
</table>

To validate our statement that the unshared feature extractors can capture more domain specific features than a shared feature extractor, we compare the shared and unshared F networks on the most challenging domain pair A31→W10, and the results are 71.5% for shared, and 76.3% for unshared.

5.4 Summary

This chapter extends the adversarial nets-based unsupervised domain adaptation to partial domain adaptation. A weighting scheme based on the activations of the adversarial nets is proposed for detecting the samples from the source domain outlier classes to effectively reduce the shift between the target data and the source data that are within the target classes. The experimental results show that the proposed method outperforms previous domain adaptation methods to a large degree and is comparable to the state-of-the-art.
partial transfer methods, demonstrating the effectiveness of the proposed method. For the future work, we will further exploit the method with the focus on larger scale partial domain adaptation.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5.3.png}
\caption{The accuracy curve of varying the number of target classes for the baselines and the proposed method.}
\end{figure}
Figure 5.4: The t-SNE visualization of the activations of baseline methods and the proposed method. The blue dots are expected to be aligned with green dots for effective domain adaptation.
Figure 5.5: The t-SNE visualization of the alignment of source and target classes for the proposed method. The red numbers and blue numbers represent samples of the source and target domain, respectively, and the values represent the classes.
Figure 5.6: The t-SNE visualization of the learned weights of source samples for the proposed method. The red colored numbers indicate higher weights, while the blue colored numbers indicate lower weights. The intermediate values are arranged based on the color bar.
Chapter 6

Domain Expansion from Multiple Sources

6.1 Introduction

This chapter formally introduces a new problem referred to as multi-source domain expansion (MSDE). It is defined as follows.

Definition 5 (Multi-source Domain Expansion (MSDE)) Let $\Delta = \{P_1, ..., P_m\}$ be a set of $m$ source domains and $P_{\text{new}} \notin \Delta$ be a new domain. The source domains and the new domain have different distributions and focus on the same type of tasks (e.g. classification) with overlapping label spaces. $P_t = \Delta \cup P_{\text{new}}$ is the target domain. The task of MSDE is to learn a labelling function $f_{P_t} : X \rightarrow Y$ that performs well on $P_t$.

The domain expansion problem has its unique practicality. Its goal is to learn an unbiased classifier that is able to classify or recognize real and diverse visual data, i.e. expanded domain being composed of different domains, rather than merely a small and specific domain. For example, in the application of classification task on the Mobile devices, a system can be pre-trained on different sources of images (i.e. the source domains). When customers use this system, the classification capacity of the original system could be improved using the unlabelled customer photos (which are generally from a new domain and can be used for expanding the source domains), such that the new updated model is more robust to classify new images (which can be from either the source domains or the new domain). In addition, domain expansion would potentially offer a new approach to continuously improving and expanding a previously learned model without the need of learning it from scratch.

The problem of MSDE is different from domain adaptation (DA) whose target domain is defined as the new domain $P_{\text{new}}$ alone, hence, the labelling function to be learned is only required to work on the new domain $P_{\text{new}}$ in DA. Though some DA methods [314, 13, 83, 175] use as a constraint to avoid trivial solutions the condition for the newly
learned labelling function to maintain its performance in the source domains as much as possible, it is not a MUST to have criteria. In MSDE, performance in both source domains and the new domain are equally important. Note that the domain labels of the unseen data are generally unknown. In this case, though the source model can classify the source samples well, the performance on the test data will be unknown because we do not know which domain the test data belong to. The MSDE is also related to domain generalization. Domain Generalization (DG) \([19, 132, 191, 297, 157]\) uses the data from multiple source domains to learn an unbiased model that can be generalized to the unseen target domain. Most of the existing work tackle this problem by learning domain invariant and compact representation from multiple source domains \([19, 191]\), or by learning robust classifiers from multiple sources to generalize well on unseen target domain \([132, 297, 157]\). However, DG assumes the data from multiple source domains are available for feasible domain generalization and the new domain data are not exploited when available. In addition, the learned DG model is expected to perform well on average, but not the best on any particular domain \([132]\), which means the performances on the source domains will be dropped. By contrast, we expect the model on the expanded domain preserves the performance on the source domain(s) and improves the performance on the new domain at the same time. MSDE is also different from multi-task learning (MTL) \([29]\), lifelong Learning (i.e. online MTL) \([253, 223]\), and incremental learning \([155, 152, 311]\) all of which assume no domain shift among different tasks. Thus, the existing approaches to multi-task learning, lifelong Learning, multi-source domain adaptation, and incremental learning are usually not directly applicable to the MSDE problem.

Depending on whether data or labelling functions are available for the source domains, whether the new domain share the identical labelling space to the source domains, and whether there are sufficient data and/or labels are available for the new domain, many sub-problems of MSDE can be formulated similarly to that in the conventional DA \([314]\). For instance, if labelled data in the source domain and limited labelled data in the new domain are available, this problem is called supervised MSDE. If only unlabelled data are available for the new domain, it becomes unsupervised MSDE. This chapter is concerned with a challenging sub-problem in MSDE, referred to as unsupervised MSDE without source data. It is assumed that no data in the \(m\) source domains are available, instead, models or labelling functions for the individual \(m\) source domains and unlabelled data from the new domain \(P_{\text{new}}\) are given. Since the individual models are learned from the corresponding individual source data, they are inevitably biased. To learn an unbiased model that can perform well on the expanded domain, the biases among different individual domains need to be dealt with. However, due to the unavailable data in the source domain, the biases (e.g. distribution shift) among different domains (i.e. the multiple sources and the new domain) cannot be measured and dealt with explicitly. The key advantage of this scenario is that there is no need to store data of
source domains and this advantage is more appealing when the number $m$ of the source
domain is large. This is practical in the lightweight systems like mobile portable devices,
where the storage space is limited.

Specifically, the proposed solution is based on the observation that, after feeding the
data in $P_{\text{new}}$ into different source models, their activations reflect the biases among do-
mains, as well as the discriminative information in the new domain. Therefore, this chap-
ter proposes to deal with the domain bias by reducing the divergence among the activa-
tions on different source models by the data in $P_{\text{new}}$ with a properly designed mechanism
of learning the source model weights to emphasize more on the source domains that are
more similar to $P_{\text{new}}$. In order to transfer the source domain discriminative information
and preserve the source domain performance simultaneously, the activations of the origi-
nal source models and the adapted models by the new domain data should not change too
much. Some recent work [155, 152, 311] proposed to use only the new labelled data to
tune a network pre-trained on source data (without access to the original source data) as
a means of incremental learning (or continual learning) without assuming the existence
of bias between the unseen source data and new data. The proposed method extends the
idea in [155] to adapt the source models to a target model for the target (i.e. expanded)
domain using the unlabelled data from the new domain such that the target model works
for both the source domains and the new domain while reducing the domain bias among
them.

6.2 Unsupervised MSDE without Source Data

In general, the degree of domain bias among the $m$ sources, $P_1, P_2, \cdots, P_m$ and new domain
$P_{\text{new}}$ are different, suggesting that some sources may have more similar distributions to
the new domain than others. Effective handling of the biases is a key in the unsupervised
MSDE without source data. We propose to use the activations of different source models
by the new domain data, the only information that can be obtained by merely using new
domain data and source models, to tackle both domain alignment and source domain
information preservation.

6.2.1 Multi-source and New Domain Alignment

To deal with the biases without access to the source domain data, we use the activations
of different source models by the new data. Different source domains have different
distributions, which can be observed through the activations of each source model by the
new data. To effectively transfer from the multiple sources, the distribution divergence
among them should be dealt with. By reducing the divergence among the activations of
different source models, not only the biases among sources are dealt, the biases between
sources and the new domain are also dealt with.

We take the case of two source domains as an example. The intuition is that if the two source domains have a large distribution divergence, the learned individual source models are very different from each other. Then the two individual source models would possibly be activated differently by a same new sample. The difference not only reflect the bias between the two source domains, but also reflect the new samples that are not in the support of the intersection of the two source domains (since if a new sample is within the support of the intersection of the two source domains then the activations of the two source models by this target sample should be similar). Hence, to deal with bias among multiple sources as well as between sources and the new domain, we propose to reduce the divergence among the activations of different source models by the new data as follows:

\[ L_{bias}^{i} = \min_{\theta_i} \sum_{j=1, j \neq i}^{m} l_{bias}(y_n(\theta_i), y_n(\theta_j)) \]  

(6.1)

where \( L_{bias}^{i} \) represents the domain bias loss for the \( i \)th source model, \( y_n(\theta_i) \) are the activations (classification scores) of the \( i \)th source model by the new domain data parametrized by \( \theta_i \) among \( m \) source models, and \( l \) is some loss function expressing the dissimilarity between activations of \( i \)th and \( j \)th source models. Here, we choose to use the \( L_2 \) losses with temperature between pairs of sources:

\[ l_{bias}(y_n(\theta_i), y_n(\theta_{oi})) = \frac{1}{k_n} \sum_{k=1}^{k_n} \| y_{nk}(\theta_i/T_0) - y_{nk}(\theta_j/T_0) \|_2^2 \]  

(6.2)

where \( k_n \) is the number of unlabelled training data in the new domain, \( T_0 \) is the temperature value. The use of temperature is inspired by the knowledge distillation [106]. If \( T_0 \) is large, the probability distribution over classes is softer. The obtained activations (classification scores) of the source models by the new domain data may not be able to well reflect which class the new domain sample belong to due to the domain bias, but they can reflect the information of class relationships (e.g. which classes the target sample is more close to). If \( T_0 \) is large, the softer probability distribution over classes gives a richer representation of the target sample, but if the \( T_0 \) is too large, the class relationship information will be destroyed since the probabilities of all the classes are similar. Hence, a proper value of \( T_0 \) will help the preservation of target data information.

### 6.2.2 Learning of Source Model Weight

To put more emphasize on the best source domains, we propose to evaluate the classification capabilities of different source models on the new domain data using the entropy
which measures the class overlap:

\[ E_i = \frac{1}{k_i} \sum_{k=1}^{k_i} H(y_{nk}(\theta_i)) \]  

where \( H(\cdot) \) is the information entropy function. If the entropy is low, it suggests that the new domain data have small class overlap on the specific source model and the source model performs well on the new domain since, intuitively, the decision boundary produced by the source model lies in low-density areas of the new domain data (i.e. the source model can successfully classify the new domain data). On the other hand, if the source model performs poorly on the new domain, most of the new domain samples will likely produce vague activation values on many classes rather than a large value on a certain class and thus the entropy of the new domain data activations is high. Thus, if the class overlap is small, we assume the model performs well on the new domain data and the domain bias is small between this source and the new domain.

Since some sources may have more similar distributions to the new domain than others and thus the corresponding source models should not be changed too much when reducing the domain bias among sources. Hence, we assign different weights based on the entropy (Eq. 6.3) to source domains when reducing the domain divergence. Hence, the weights are defined as follows

\[ w_i = \frac{\exp(E_i/T)}{\sum_{j=1}^{m} \exp(E_j/T)} \]  

where the \( w_i \) is the learned normalized weight on the \( i \)th source domain, \( T \) is the temperature to control the weight differences among sources (i.e. similar to \( T_0 \), a higher temperature value gives a softer probability distribution over all the weights). The weights \( w_i \) are assigned to the source domains when reducing the domain bias using Eq. 6.1. Previous works also learn the source model weights for MSDA tasks [184, 61, 319, 294]. However, the learning of weights in previous works generally rely on data from both source domains and the new domain, while only the source models and the unlabelled new domain data are required in our method. In addition, the criteria of the learning of the weights are different. In previous MSDA methods, the source model weights are generally learned based on the domain bias between each source domain and the new domain. By contrast, our source model weights are learned based on the classification capabilities of each source model on the new domain data. Thus, the proposed method not only considers the degree of domain bias between each source domain and the new domain but also considers the classification capability (which cannot be reflected merely through the domain bias) of each source model, which benefits to the performance of the new domain and the overall MSDE tasks. Moreover, the usages of the source weights are also different from the previous MSDA work. The source model weights in previous MSDA methods are generally directly used for construct the new target domain classifier (i.e. the weighted combina-
tion of multiple source classifiers). By contrast, our source model weights are used for reducing the bias among different domains.

### 6.2.3 Source Domain Information Preservation

However, if we simply reduce the divergence among the activations of different source models, the source domain discriminative information is missing and performance on the original source data will be destroyed. In addition, some trivial solution will be obtained without any constraints on $L_{\text{bias}}$. Hence, we propose another loss that after adaptation, the activations of new source models should not be far away from that of the old source models by the new domain data, which is similar to learning without forgetting [155]. This loss not only preserves the discriminative information and source domain performance, but also acts as a regularization to avoid trivial solution when reducing the domain bias.

The loss $L^i_{\text{original}}$ for the $i$th source model is defined as follows,

$$L^i_{\text{original}} = \min_{\theta_i} l_{\text{original}}(y_n(\theta_i), y_n(\theta_{oi}))$$  \hspace{1cm} (6.5)

where $y_n(\theta_{oi})$ is the activations of the original $i$th source model by the new domain data, and $l$ is some loss functions. Here, the $L_2$ loss with temperature is used:

$$l_{\text{original}}(y_n(\theta_i), y_n(\theta_{oi})) = \frac{1}{k_n} \sum_{k=1}^{k_n} \| y_{nk}(\theta_i/T_0) - y_{nk}(\theta_{oi}/T_0) \|_2^2$$  \hspace{1cm} (6.6)

For simplicity, the temperature ($T_0$) here is set identically to that in Eq. 6.2

### 6.2.4 Overall Model

By combining the weighted multi-source alignment loss (Eq. 6.1, 6.4) and source information preservation loss (Eq. 6.5), the overall objective for the $i$th source model is as follows,

$$L^i_{\text{overall}} = L^i_{\text{original}} + \lambda \cdot w_i \cdot L^i_{\text{bias}}$$  \hspace{1cm} (6.7)

where $\lambda$ is a trade-off parameter.

The source models are updated one by one in each step, which means when one model is updated the rest of the source models are kept fixed. After learning, the updated and/or original source models can be fused to construct a model for the expanded or target domain. This chapter, the model for the expanded domain is constructed in an ad-hoc way. First, the final classification score $y_{final}(x_t)$ of a test sample $x_t$ is the sum fusion over all the updated models, i.e.

$$y_{final}(x_t) = \sum_{i=1}^{m} y_t(\theta_i)$$  \hspace{1cm} (6.8)
If the original source models could be kept as well, the final classification score \( y_{\text{final}}(x_t) \) of a test sample \( x_t \) is the sum fusion of the max fusion of each original and updated source model, i.e.

\[
y_{\text{final}}(x_t) = \sum_{i=1}^{m} (\max(y_t(\theta_i), y_t(\theta_{oi})))
\]

(6.9)

The max fusion selects the individual model with more confidence on a certain class and the sum fusion over different individual models leverage the information from different source models. More effective fusion of the models in a principled way will be studied in the future.

### 6.3 Experiments

This section presents the experiments for evaluating the proposed method.

#### 6.3.1 Datasets and Implementation Details

**Datasets** The experiments were conducted on three real-world cross-domain object recognition datasets: VLCS [258, 132], ImageCLEF_DA\(^*\), and PACS [157]. Each dataset contains multiple domains, such that the proposed method can be evaluated.

The VLCS dataset consists of images from PASCAL VOC2007 (V) [67], LabelMe (L) [222], Caltech-101 (C) [71], and SUN09 (S) [41] datasets, each of which represents one domain. Five categories are shared among these datasets: bird, car, chair, dog, and person.

The ImageCLEF_DA is a benchmark dataset for ImageCLEF 2014 domain adaptation challenge. Twelve shared categories are selected from four public image datasets: Caltech-256 (C) [102], ImageNet ILSVRC2012 (I) [53], PASCAL VOC2012 (P) [67], and Bing (B) [17], forming four different domains.

The PACS dataset is a recently collected cross-domain recognition dataset. This dataset was initially designed for domain generalization task, which can also be used for evaluating MSDA algorithms since it consists of four different domains (Art painting (A), Cartoon (C), Photo (P), Sketch (S)) with seven common categories. The dataset is created by combining shared classes from four image datasets: Caltech256 (Photo) [102], Sketchy (Photo, Sketch) [226], TU-Berlin (Sketch) [66] and Google Images (Art painting, Cartoon, Photo). Note that the different domains in this dataset are not characterized by different original dataset, but by the image modalities or styles.

Sample images from different domains in the three datasets are shown in Fig. 6.1, 6.2, and 6.3. It can be seen that in VLCS dataset, the domain bias is quite large due to different

\*http://imageclef.org/2014/adaptation
background complexities as well as different sizes of the object of interest. For the ImageCLEF_DA dataset, the images selected from Caltech and ImageNet are quite similar, while the images from Bing have more cluttered backgrounds with noisy objects compared to the other three domains. As mentioned, the four domains in PACS are characterized by different image modalities or styles. From the sample images, it can be seen that the divergences between different domains in PACS are visually larger than the other two datasets. The authors of PACS [157] also evaluated the domain bias in PACS compared to that in VLCS from the perspective of both feature spaces and classifier performances. It has been shown that the divergences between domains in PACS are larger.

![Figure 6.1: Sample images of class “dog” in VLCS dataset.](image)

![Figure 6.2: Sample images of class “aeroplane” in ImageCLEF_DA dataset.](image)

![Figure 6.3: Sample images of class “horse” in PACS dataset.](image)
CHAPTER 6. DOMAIN EXPANSION FROM MULTIPLE SOURCES

Implementation Details

Implementation Details All three datasets have four different domains. In the experiments, three domains were used as source domains and the rest was used as the new domain, resulting four different cases from each dataset. Samples in each domain are randomly divided into a training set (70%) and a test set (30%). The training data of the source domains were used for training the source models and then will not be used in the MSDE task. The pre-trained source models and the unlabelled training data of the new domain were used for training the proposed method and the unseen test data were used for evaluating the learned model.

The individual deep source models were obtained separately by fine-tuning an AlexNet [138] for the VLCS dataset, a ResNet50 [105] for both ImageCLEF_DA and PACS datasets using Caffe [123], all networks were initialized by a corresponding model pre-trained on ImageNet. The proposed MSDE without source data method was implemented using PyTorch platform[^1]. The trade-off parameter was set to $\lambda = 10$ and the temperatures for aligning different domains and for the source model weights were set to $T_0 = 3$ and $T = 0.1$ respectively. The learning rate was $1e-6$. Note that these parameters were selected empirically.

6.3.2 Results

Tables 6.1, 6.2, and 6.3 show the experimental results using the deep source models. The performance of the new domain test data (New), the multiple source domain test data ($S_0$, $S_1$, and $S_2$), as well as that on the target (i.e. expanded) domain calculated as the average over the new domain and the multiple source domains are presented. In the tables, M1 and M2 represent the fusion methods defined by Eqs. (6.8) and (6.9) respectively. The results are also compared with a baseline, denoted as “Base” in the tables. The baseline is defined as the sum fusion of the original source models.

As shown in the tables, the proposed methods outperformed the baseline in the target or expanded domain in eleven of the twelve cases over the three datasets. The only exceptional case is on the imageCLEF_DA dataset in which the proposed method performed marginally worse (i.e. less than one percentage point). The two fusion methods (i.e. M1 and M2 in the tables) performed comparably.

Considering the performance in the new domain, the proposed method outperformed the baseline in most of the cases. In six of the twelve cases, it is by a large margin. Notice that there are two cases that the proposed method did not improve the performance in the new domain compared to the baseline. This is probably that the distributions of the multiple source domains overlap with that of the new domain, or the distribution shift

cannot be characterized merely by the source models. Comparing the two fusion methods, it can be seen that the first fusion method generally outperforms the second method on the new domain data as expected since only the models updated by the new domain are used in the first fusion methods. Note that since the source data are not available, fair comparisons cannot be made between the proposed method and the previous MSDA methods in the new domain.

Also as expected, the performance improvement in the new domain does not degrade the performance in the source domains in most of the cases. In contrast, the performances in the source domains are even improved after expansion using the proposed method in more than half of the cases over the three datasets. This is probably because the learned model not only deals with the bias between the source domains and the new domain, but also deals with the bias among multiple source domains.

Experiments were conducted using max fusion over all the individual models for both the baseline and the proposed method, the improvement of the performance over the various cases have the similar trend as using the sum fusion. Therefore, only results of sum fusion are presented in the chapter.

### Table 6.1: The accuracies (%) on the VLCS dataset.

(a) Source domains: L, S, V.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Base</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(New)</td>
<td>92.92</td>
<td><strong>96.23</strong></td>
<td><strong>95.05</strong></td>
</tr>
<tr>
<td>L(S0)</td>
<td><strong>64.87</strong></td>
<td>63.11</td>
<td>64.24</td>
</tr>
<tr>
<td>S(S1)</td>
<td><strong>77.64</strong></td>
<td>75.51</td>
<td>77.54</td>
</tr>
<tr>
<td>V(S2)</td>
<td><strong>76.01</strong></td>
<td>75.62</td>
<td><strong>76.01</strong></td>
</tr>
<tr>
<td>Expanded</td>
<td>77.86</td>
<td>77.62</td>
<td><strong>78.21</strong></td>
</tr>
</tbody>
</table>

(b) Source domains: C, S, V.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Base</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>L(New)</td>
<td><strong>60.73</strong></td>
<td>59.97</td>
<td><strong>60.48</strong></td>
</tr>
<tr>
<td>C(S0)</td>
<td><strong>96.70</strong></td>
<td>94.34</td>
<td>96.46</td>
</tr>
<tr>
<td>S(S1)</td>
<td>75.91</td>
<td>78.05</td>
<td><strong>78.46</strong></td>
</tr>
<tr>
<td>V(S2)</td>
<td><strong>78.28</strong></td>
<td>75.52</td>
<td>76.80</td>
</tr>
<tr>
<td>Expanded</td>
<td>77.91</td>
<td>76.97</td>
<td><strong>78.05</strong></td>
</tr>
</tbody>
</table>

(c) Source domains: C, L, V.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Base</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(New)</td>
<td>69.00</td>
<td><strong>73.98</strong></td>
<td>72.87</td>
</tr>
<tr>
<td>C(S0)</td>
<td><strong>98.11</strong></td>
<td>97.88</td>
<td>97.88</td>
</tr>
<tr>
<td>L(S1)</td>
<td>64.99</td>
<td><strong>65.37</strong></td>
<td>65.24</td>
</tr>
<tr>
<td>V(S2)</td>
<td>75.52</td>
<td><strong>78.87</strong></td>
<td>78.18</td>
</tr>
<tr>
<td>Expanded</td>
<td>76.91</td>
<td><strong>79.03</strong></td>
<td>78.54</td>
</tr>
</tbody>
</table>

(d) Source domains: C, S, L.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Base</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>V(New)</td>
<td>68.90</td>
<td>68.90</td>
<td><strong>69.69</strong></td>
</tr>
<tr>
<td>C(S0)</td>
<td><strong>94.81</strong></td>
<td>94.10</td>
<td>94.34</td>
</tr>
<tr>
<td>S(S1)</td>
<td>72.66</td>
<td><strong>76.83</strong></td>
<td>76.73</td>
</tr>
<tr>
<td>L(S2)</td>
<td>66.62</td>
<td><strong>69.51</strong></td>
<td>68.26</td>
</tr>
<tr>
<td>Expanded</td>
<td>75.74</td>
<td><strong>77.34</strong></td>
<td>77.26</td>
</tr>
</tbody>
</table>

### 6.3.3 Discussion and Analysis

#### Performance of Updated Source Models

To show how well the individual updated source models by the proposed method work, experiments were conducted to compare the performance of the original source models and the updated models in every source domain. Table 6.4 shows the results of the case (L,S,V→C) on the VLSC dataset using deep source models. It can be seen that after
expansion using the proposed method, the performance of each updated source model is not only improved in the new domain but also improved in other source domains. This verifies that the updated model not only deals with the bias between source domains and the new domain, but also deals with the bias among source domains. This trend is also observed in other cases on the other two datasets.

Performance using Shallow Models

To further demonstrate the robustness of the proposed method, experiments were conducted using shallow source models which are a multi-layer perceptron with one-layer of

---

### Table 6.2: The accuracies (%) on the imageCLEF_DA dataset.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Base</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>B(New)</td>
<td>62.22</td>
<td>61.67</td>
<td></td>
</tr>
<tr>
<td>C(S0)</td>
<td>95.00</td>
<td>94.44</td>
<td>95.00</td>
</tr>
<tr>
<td>I(S1)</td>
<td>90.56</td>
<td>91.67</td>
<td>90.56</td>
</tr>
<tr>
<td>P(S2)</td>
<td>76.67</td>
<td>78.89</td>
<td>78.89</td>
</tr>
<tr>
<td>Expanded</td>
<td>81.11</td>
<td>81.67</td>
<td>81.53</td>
</tr>
</tbody>
</table>

(a) Source domains: C, I, P.

(b) Source domains: B, I, P.

(c) Source domains: B, C, P.

(d) Source domains: B, I, C.

---

### Table 6.3: The accuracies (%) on the PACS dataset.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Base</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(New)</td>
<td>79.80</td>
<td>86.48</td>
<td>85.99</td>
</tr>
<tr>
<td>C(S0)</td>
<td>82.08</td>
<td>88.48</td>
<td>87.77</td>
</tr>
<tr>
<td>P(S1)</td>
<td>96.81</td>
<td>95.61</td>
<td>95.61</td>
</tr>
<tr>
<td>S(S2)</td>
<td>94.83</td>
<td>86.51</td>
<td>88.04</td>
</tr>
<tr>
<td>Expanded</td>
<td>88.38</td>
<td>89.27</td>
<td>89.35</td>
</tr>
</tbody>
</table>

(a) Source domains: C, P, S.

(b) Source domains: A, P, S.

(c) Source domains: A, C, S.

(d) Source domains: A, P, C.
Table 6.4: The accuracies (%) in individual domains using individual original and updated source models for the case of L,S,V→C on the VLCS dataset where “Orig.” refers to the original models and “Upd.” represents the updated models.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>85.61</td>
<td>93.40</td>
<td>C</td>
<td>43.63</td>
<td>95.05</td>
<td>C</td>
<td>95.28</td>
<td>95.99</td>
</tr>
<tr>
<td>L</td>
<td>75.91</td>
<td>75.41</td>
<td>L</td>
<td>58.85</td>
<td>58.59</td>
<td>L</td>
<td>57.97</td>
<td>58.09</td>
</tr>
<tr>
<td>S</td>
<td>49.29</td>
<td>52.95</td>
<td>S</td>
<td>79.57</td>
<td>74.39</td>
<td>S</td>
<td>71.85</td>
<td>72.15</td>
</tr>
<tr>
<td>V</td>
<td>58.93</td>
<td>63.97</td>
<td>V</td>
<td>58.34</td>
<td>62.88</td>
<td>V</td>
<td>80.55</td>
<td>80.36</td>
</tr>
<tr>
<td>Average</td>
<td>67.44</td>
<td>71.43</td>
<td>Average</td>
<td>60.10</td>
<td>72.73</td>
<td>Average</td>
<td>76.41</td>
<td>76.65</td>
</tr>
</tbody>
</table>

Table 6.5: The average accuracies (%) obtained on the expanded domain using shallow models on the three datasets and the average accuracies (%) over all the datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Base</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLCS</td>
<td>73.69</td>
<td>75.82</td>
<td>75.51</td>
</tr>
<tr>
<td>imageCLEF</td>
<td>77.67</td>
<td>78.47</td>
<td>78.44</td>
</tr>
<tr>
<td>PACS</td>
<td>77.11</td>
<td>76.12</td>
<td>77.21</td>
</tr>
<tr>
<td>Average</td>
<td>76.16</td>
<td>76.81</td>
<td>77.05</td>
</tr>
</tbody>
</table>

6.3.4 Ablation Studies and Parameter Sensitivities

We conducted the ablation experiments to validate the effectiveness of the proposed method for learning the source model weights and evaluate the sensitivities of parameters.

On the Source Model Weights

The first ablation study is to evaluate the effectiveness of the proposed source model weights when aligning the multiple source domains and the new domain. Figure 6.4 illustrates the performance on the new domain and the expanded domain by comparing between using equal weights and using the learned weights on the three datasets as well as the average over the three datasets. The results show that when the source models are assigned with equal weights, the proposed method can still obtain better results compared to the baselines and using the learned source model weights can further improve the results on average on both the new domain and the expanded domain. Table 6.6 shows the accuracies obtained on the new domain using a single original source model and the
learned weights of each source model using Eq. 6.4. It can be seen that the higher accuracy in a specific new domain generally corresponds to a lower weight while the lower accuracy generally corresponds to a higher weight. This is consistent with our assumption that the better the performance the smaller the entropy.

**Figure 6.4:** Comparison of the performance on the new and expanded domain data using equal weights (EW) and using the proposed learned weights (LW) on the three datasets.

**Table 6.6:** The the accuracies (%) obtained on the new domain using a single original source model and the learned weight of each source model using the proposed method on the three datasets.

<table>
<thead>
<tr>
<th>Model</th>
<th>VLCS Acc.</th>
<th>Model</th>
<th>imageCLEF Acc.</th>
<th>Model</th>
<th>PACS Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>weight</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L$</td>
<td>85.61</td>
<td>0.1168</td>
<td>$C$</td>
<td>58.89</td>
<td>0.4427</td>
</tr>
<tr>
<td>$S$</td>
<td>43.63</td>
<td>0.8353</td>
<td>$I$</td>
<td>56.67</td>
<td>0.2038</td>
</tr>
<tr>
<td>$V$</td>
<td>95.28</td>
<td>0.048</td>
<td>$P$</td>
<td>54.44</td>
<td>0.3535</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Domain: $C$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C$</td>
<td>55.96</td>
<td>0.7281</td>
<td>$B$</td>
<td>93.33</td>
<td>0.2683</td>
</tr>
<tr>
<td>$S$</td>
<td>58.85</td>
<td>0.1314</td>
<td>$I$</td>
<td>88.89</td>
<td>0.3327</td>
</tr>
<tr>
<td>$V$</td>
<td>57.97</td>
<td>0.1405</td>
<td>$P$</td>
<td>88.33</td>
<td>0.399</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Domain: $L$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C$</td>
<td>52.74</td>
<td>0.5557</td>
<td>$B$</td>
<td>80.56</td>
<td>0.3558</td>
</tr>
<tr>
<td>$L$</td>
<td>49.29</td>
<td>0.3192</td>
<td>$C$</td>
<td>77.22</td>
<td>0.4792</td>
</tr>
<tr>
<td>$V$</td>
<td>71.85</td>
<td>0.125</td>
<td>$P$</td>
<td>85.56</td>
<td>0.165</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Domain: $S$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C$</td>
<td>55.97</td>
<td>0.431</td>
<td>$B$</td>
<td>71.76</td>
<td>0.4252</td>
</tr>
<tr>
<td>$S$</td>
<td>58.34</td>
<td>0.2002</td>
<td>$I$</td>
<td>77.22</td>
<td>0.2428</td>
</tr>
<tr>
<td>$L$</td>
<td>58.93</td>
<td>0.3688</td>
<td>$C$</td>
<td>67.22</td>
<td>0.332</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Domain: $V$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
On the Temperature

The second ablation study is to validate the effectiveness of the use of a proper temperature when aligning different domains as well as adapting the source information to the new domain. Experiments were conducted using the deep models on the three datasets and the average accuracies on the new domain are reported because the performance of the source domain is not sensitive to the temperature. Figure 6.5 shows the average accuracy over the four cases on each dataset when changing the value of the temperature. It shows that when the temperature increases, the improvement of performance can be observed on most of the datasets. However, the performance will drop slightly if the temperature is too large because the class relationship information would be destroyed since the probabilities of all classes would be similar. In addition, even with a small temperature of 1.0, the proposed method outperforms the baseline. It indicates that the proposed method can effectively adapt multiple source models to the new domain. In addition, a higher temperature value produces a softer probability distribution over classes, and thus more information on class relationships can be preserved, resulting in the improvement of performances.

![Figure 6.5: The average accuracy (%) when changing the temperature value compared to the baselines.](image)

![Figure 6.6: The average accuracy (%) when changing the trade-off parameter \( \lambda \) compared to the baselines.](image)

On the Trade-off Parameter

This study aims at evaluating the sensitivity of the trade-off parameter \( \lambda \). The accuracies on the new domain are reported because the performances of the source domain test data are not sensitive to \( \lambda \). Figure 6.6 demonstrates the average accuracy over four cases on each dataset when changing the value of \( \lambda \) and shows that when the \( \lambda \) getting large, the performances on the new domain are improved because the bias among domains would be largely reduced. If the \( \lambda \) is too large (i.e. \( \lambda = 100 \)), the performance in the source
domains could be adversely affected. However, it can be seen that there is a large range of $\lambda$ values in which the performance can be improved compared to the baseline, therefore, $\lambda$ is easy to choose. In addition, this issue can be alleviated by including the original source models in the fusion (e.g. M2).

### 6.4 Summary

This chapter is concerned with multi-source domain expansion (MSDE), where the target domain is formed jointly by the source domains and new domain. Thus the learned label function is expected to work equally well for all source domains and the new domain. Specifically, this chapter proposes a method for an unsupervised MSDE problem where only models of source domains and unlabelled new domain data are available. A possible venue to further improve the proposed method is to effectively fuse the original source models and updated models.

The proposed concept of Domain expansion (DE) captures a new research dimension in transfer learning. Though keeping the performance in the source domains has sometimes been used to as constraints in traditional domain adaptation (DA) [314], but this has never formally been a compulsory requirement in DA. In addition, DE has the potential to formally bridge DA with online and incremental learning. Many problems in DE are yet to be investigated in future.
Chapter 7

Conclusions and Future Work

7.1 Conclusions

Transfer learning from previous data for current tasks has a wide range of real-world applications. The main objective of this thesis is to address some challenges in transfer learning, with the focus on cross-domain visual recognition. The following conclusions are drawn from the thesis.

- The JGSA method can effectively reduce the domain shifts by taking both geometrical and statistical properties of source and target domain data into consideration and exploiting both shared and domain-specific features. More discriminative information in the target domain can be further exploited in the future for the improvement of the target domain performance.

- The mtUDA method provides a new perspective that unsupervised domain adaptation can be formulated as a multi-task learning problem, which learns unshared classifiers between the source and target to deal with the large shift. A possible future work is to examine this idea in the deep learning framework.

- A novel method for partial domain adaptation is proposed, which extends the adversarial nets-based unsupervised domain adaptation to partial domain adaptation. Specifically, a weighting scheme based on the activations of the adversarial nets is proposed for detecting the samples from the source domain outlier classes to effectively reduce the shift between the target data and the source data that are within the target classes. For further improvement, the method can be exploited with the focus on larger scale partial domain adaptation.

- A new problem, named multi-source domain expansion (MSDE), is introduced. In MSDE, the target domain is formed jointly by the source domains and new domain. A novel method for the unsupervised MSDE problem is proposed, where only models of source domains and unlabelled new domain data are available. A
possible venue to further improve the proposed method is to effectively fuse the original source models and updated models.

In summary, four algorithms are proposed in this thesis. The first two algorithms aim at improving the performance of the unsupervised domain adaptation (UDA). Though the two algorithms obtain the state-of-the-art results compared to previous methods, the performances obtained using the proposed methods on the target domain data are still far from comparable to the performance obtained using the model trained on sufficient target domain labelled training data. Hence, UDA is still a challenging task and requires more exploration. One of the possible future venue is to further explore the measurement of the relatedness between the source and the target domain, such that the most related source domain can be selected to avoid negative transfer. To the best of our knowledge, there are only a couple of previous works have touched this topic. For example, in [93], the authors introduce a Rank of Domain (ROD) metric to measure the relevance between domains both geometrically (the alignment between subspace) and statistically (KL divergences between data distributions after projection to the subspaces). A recent work [309] mainly focuses on finding transfer learning dependencies across different learning tasks to identify the redundancies among tasks. This idea could be borrowed to the context of domain adaptation to find the relevance between different domains. In addition, most unsupervised domain adaptation methods (including ours) only use small scale source domain data, which largely limit the adaptation capacity of the algorithms. Another limitation of current methods is that sufficient and class-balanced target domain data are required for feasible adaptation in the context of UDA. The third algorithm proposed in this thesis focuses on the partial domain adaptation (PDA). Compared to traditional UDA, research in PDA much less explored and the development of more effective algorithms is encouraged. Specifically, the core challenges of PDA include how to detect the outlier classes and reduce the shift between the shared classes. The last contribution of this thesis introduces a new problem, named domain expansion (DE). Its goal is to learn an unbiased classifier that is able to classify or recognize real and diverse visual data with less human efforts, i.e. expanded domain being composed of different domains of data, rather than merely a small and specific domain of data. In addition, DE would potentially offer a new approach to continuously improving and expanding a previously learned model without the need of learning it from scratch. Hence, DE is an interesting and practical problem but has not been formally touched previously.

Except for the addressed problems in this thesis, there are still many challenging topics in the area of transfer learning in the literature but still require more future exploration. These topics are discussed in details in Section 7.2.
7.2 Challenges and Future Directions

Transfer learning is a promising and important approach to cross-dataset visual recognition and has been extensively studied in the past decades with much success. Figure 2.1 shows the problem-oriented taxonomy and the statistics on the number of papers for each problem has shown that most previous works concentrate on a subset of problems presented in Figure 2.1. Specifically, only nine out of the seventeen problems are relatively well studied where the source and target domains share at least either their feature spaces or label spaces, the source domain data are labelled and balanced, target domain data are balanced and non-sequential. The rest eight problems especially those where the target data is imbalanced and sequential are much less explored. Such a landscape together with the recent fast-advancing deep learning approach has revealed many challenges and opened many future opportunities as elaborated below for cross-dataset visual recognition.

7.2.1 Deep Transfer Learning

As deep learning advances, transfer learning is also shifted from traditional shallow-learning based approaches to deep neural network based approaches. In practice, the deep networks for the target task are rarely trained from scratch (i.e. with random initialization), since the target datasets rarely have sufficient samples. Thus, transfer learning is generally used. The pre-trained deep models from a very large source dataset are used either as an initialization [307] (then fine-tune the model according to the target data) or a fixed feature extractor for the target task of interest [58, 219].

Similarly, in deep domain adaptation, the deep models are either used as feature extractors (then shallow-based domain adaptation methods are used for further adaptation) [304, 87, 45, 260, 313, 137], or used in an end-to-end fashion (i.e. the domain adaptation module is integrated into the deep model) [264, 174, 180, 83, 175, 262, 23]. It is still unclear which approach would perform better. The advantage of using deep models as feature extractors is that the computational cost is much lower since shallow-based DA methods are generally much faster than deep learning-based methods. Another advantage is that many shallow-based methods have a global optimum value. The drawback is that the degree of adaptation may be insufficient in the shallow-based methods to fully leverage the deeply extracted features. On the other hand, the advantage of integrating an adaptation module into deep models is two-fold. First, it is end-to-end trainable. Secondly, the adaptation can be performed in multiple levels of features. While the drawbacks are the computational cost and the local optimum. To date, these two approaches have produced the similar performance on some datasets [180, 137, 190, 315] though the end-to-end deep systems involve more parameters and require more computational costs. One of the missing studies in the literature is a systematic study and comparison of the two approaches.
under same or similar conditions. For instance, both deep and shallow-based methods can use MMD metric between distributions as a constraint to the objective function. Thus, the comparison between the two approaches using MMD metric may be conducted.

The adversarial nets derived from GANs [96] are appealing in deep learning-based transfer methods. The adversarial loss measures the JS divergence between two sets of data. In practice, the adversarial loss achieves better results and requires smaller batch sizes compared to the MMD loss [83, 155]. Currently, the adversarial nets-based transfer methods have been used on many transfer learning tasks, such as domain adaptation [82, 83, 172, 262, 23], partial domain adaptation [27, 317], cross-modal transfer [278, 154, 172, 236, 117, 306, 133, 333, 15, 156, 171], and zero-shot learning [336, 286]. However, some of the drawbacks of GANs may also remain in adversarial nets-based transfer methods, such as unclear stopping criteria and hard training.

7.2.2 Partial Domain Adaptation

Partial domain adaptation aims at adapting from a source dataset to an unlabelled target dataset whose label space is known to be a subspace of that of the source [111, 27, 317] or in a more general and challenging setting where only a subset of the label spaces between the source and target is overlapping [203]. The former may be considered to be a special case of transfer learning between heterogeneous label spaces and a typical and practical example is to transfer from a large source dataset with more classes to a small target dataset with fewer classes. The latter is a problem bearing both domain adaptation and zero shot learning. Generally, the distribution shift is caused not only by label space difference but also by the intrinsic the divergence of distributions (i.e. the distribution shifts exist between the source and target data even on shared classes). Partial domain adaptation has a more realistic setting than conventional unsupervised domain adaptation. Solutions to this problem would expand the applications of domain adaptation and provide a basic mechanism for online transfer learning and adaptation. However, few papers have been found on partial domain adaptation.

7.2.3 Transfer Learning from Multiple Sources

The multi-source domain adaptation (MSDA) [249, 63, 110, 61, 92, 294] refers to adaptation from multiple source domains that have exactly the same label space as the target domain. Intuitively, the MSDA methods should be able to obtain superior performance compared to the single source setting. However, in practice, the adaptation from multiple sources generally can only give similar or even worse adaptation results compared to transferring from one of the source domains (though not every one of them) [121, 232]. This is probably due to the negative transfer issue. In addition, most source data contains multiple unknown latent domains [110, 92] in the real-world applications. Thus, how to
discover latent domains and how to measure the domain similarities are still fundamental issues in MSDA.

A more realistic setting is incomplete multi-source domain adaptation (IMSDA) [57, 294] here each source label space is only a subset in the target domain and the union of the multiple source label spaces covers the target label space. IMSDA is a more challenging problem compared with MSDA, since the distribution shifts among the sources as well as the target domain are harder to be reduced due to the incompleteness of each source domain. In addition, when the number of sources increases, this problem will become challenging.

Multiple sources can be generalised to a target task, referred to as domain generalization [19, 132, 191, 69, 242, 86, 87, 190, 157] without the need of any target data. Domain generalization is of practical significance, but less addressed in the previous research. Since there is no target data available, domain generalization often has to learn semantically meaningful model shared across different domains.

### 7.2.4 Domain Expansion

Domain Expansion (DE) is another interesting and practical problem, which is introduced in this thesis and has not been formally touched previously. The problem of DE is different from domain adaptation (DA) whose target domain is defined as the new domain alone, hence, the labelling function to be learned is only required to work on the new domain in DA. Though some DA methods [314, 13, 83, 175] use as a constraint to avoid trivial solutions the condition for the newly learned labelling function to maintain its performance in the source domains as much as possible, it is not a MUST to have criteria. In DE, performance in both source domain(s) and the new domain are equally important. DE is also different from multi-task learning (MTL) [29], lifelong Learning (i.e. online MTL) [253, 223], and incremental learning [155, 152, 311] all of which assume no domain shift among different tasks. Thus, the existing approaches to multi-task learning, lifelong Learning, multi-source domain adaptation, and incremental learning are usually not directly applicable to the MSDE problem.

Depending on whether data or labelling functions are available for the source domains, whether the new domain share the identical labelling space to the source domains, and whether there are sufficient data and/or labels are available for the new domain, many sub-problems of DE can be formulated similarly to that in the conventional DA [314]. For instance, if labelled data in the source domain and limited labelled data in the new domain are available, this problem is called supervised DE. If only unlabelled data are available for the new domain, it becomes unsupervised DE.

The concept of Domain expansion (DE) captures a new research dimension in transfer learning. Though keeping the performance in the source domains has sometimes been
used to as constraints in traditional domain adaptation (DA) [314], but this has never formally been a compulsory requirement in DA. In addition, DE has the potential to formally bridge DA with online and incremental learning. Many problems in DE are yet to be investigated in future.

7.2.5 Online Transfer Learning

In online transfer learning [328], source data may not be fully available when the adaptation or transfer learning is being performed and/or the target data may also arrive sequentially. In addition, the source or even the target data cannot be fully stored and revisited in the future learning process. The adapted model is often required to perform well not only on the new target data but also to maintain its performance on the source data or previously seen data. Such a setting is sometimes known as incremental learning or transfer learning without forgetting under certain assumptions [166, 152, 235]. Few studies on this problem have been reported as shown in Figure 2.1.

7.2.6 Data Imbalance

The issue of data imbalance in the target dataset has been much neglected in the previous research, while imbalanced source data may be converted to balanced ones by discarding or re-weighting the training (source) data during the learning procedure. However, the target data can hardly follow such a process especially when the target data is insufficient. Data imbalance can be another source of distribution divergence between datasets and is ubiquitous in real-world applications. So far, there has been little study on how the existing algorithms for cross-dataset recognition would perform on imbalanced target data or how the imbalance would affect the algorithm performance.

7.2.7 Few-shot and Zero-shot Learning

Few-shot learning and Zero-shot learning are interesting and practical sub-problems in transfer learning which aim to transfer the source models efficiently to the target task with only little (few-shot) or even no target data (zero-shot). In few-shot learning, the target data are generally rare (i.e. only one training sample is available for each class in the extreme case). Thus, the standard supervised learning framework could not provide an effective solution for learning new classes from only a few samples [71, 146]. This challenge becomes more obvious in the deep learning context, since it generally relies on larger datasets and suffers from overfitting in the case of insufficient data [267, 240].

Compared to few-shot learning, zero-shot learning does not require any target data. A key challenge in zero-shot learning is the issue of projection domain shift [77], which is neglected by most previous work. Since the source and target categories are disjoint,
the projection obtained from the source categories is biased if they are applied to the target categories directly. For example, both zebra (one of the source class) and pig (one of the target class) have the same attribute ‘hasTail’, but the visual appearances of the tails of zebra and pig are very different. However, to deal with the projection domain shift problem, the unlabelled target data are generally required. Thus, further exploration of new solutions to reduce the projection domain shift is useful for effective zero-shot learning. Another future direction is the exploration of more high-level semantic spaces for connecting seen and unseen classes. The most frequently used high-level semantics are manually annotated attributes or text descriptions. Some recent work [76, 187, 241, 153] employs the word vector as semantic space without relying on human annotation, but the performance of zero-shot learning using word vector is generally poorer than that using manually labelled attributes.

A recent work [285] presents a comprehensive analysis of the recent advances in zero-shot learning. They critically compare and analyse the state-of-the-art methods and unifies the data splits of training and test sets as well as the evaluation protocols for zero-shot learning. Their evaluation protocol emphasizes on the generalized zero-shot learning, which is considered more realistic and challenging. The traditional zero-shot learning generally assumes that the training categories do not appear at test time. By contrast, the generalized zero-shot setting relaxes this assumption and generalizes to the case where both seen and unseen categories are presented in the test stage, which provides standard evaluation protocols and data splits for fair comparison and realistic evaluation in the future.

7.2.8 Cross-modal Recognition

The cross-modal transfer, a sub-problem of heterogeneous domain adaptation and heterogeneous transfer learning as shown in Figure 2.1, refers to transfer between different data modalities (e.g. text v.s. image, image v.s. video, RGB v.s. Depth, etc.). Compared to cross-modal retrieval [269] and translation [117], fewer works are dedicated to cross-modal recognition through adaptation or transfer learning. The recognition across data modalities is ubiquitous in the real-world applications. For instance, the depth images acquired by the newly released depth cameras are much rarer compared to RGB images. Effectively using rich and massive labelled RGB images to help the recognition of depth images can reduce the extensive efforts of data collection and annotation. Some preliminary works can be found in [122, 104, 156, 167, 276].

7.2.9 Transfer Learning from Weakly Labelled Web Data

The data on the Internet are generally weakly labelled. Textual information (e.g., caption, user tags, or description) can also be easily obtained from the web as additional meta
information for visual data. Thus, effectively adapting the visual representations learned from the weakly labelled data (e.g. web data) or co-existent weakly labelled other modality data to new tasks is an interesting direction and practically important. A recent work releases a large scale weakly labelled web image dataset (WebVision [163]).

### 7.2.10 Self-taught Learning

A natural assumption among most of the literature is that the source data are extensive and labelled. This may be because the source data are generally treated as the auxiliary data for instructing or teaching the target task and the unlabelled source data could be unrelated and may lead to negative transfer. However, some research works argue that the redundant unlabelled source data can still be a treasure as a good starting point of parameters for the target task as mentioned in Section 2.5.5. How to effectively leverage the massively available unlabelled source data to improve the transfer learning approaches is an interesting problem.

### 7.2.11 Large Scale and Versatile Datasets for Transfer Learning

The development of algorithms usually depends very much on the available datasets for evaluation. Most of the current visual datasets for cross-dataset recognition are small scale in terms of either number of classes or number of samples and they are especially not suitable for evaluating deep learning algorithms. An establishment of truly large-scale versatile (i.e. suitable for different problems) and realistic dataset would drive the research a significant step forward. As well known, the creation of a large-scale dataset may be unaffordably expensive. Combinations and re-targeting of existent datasets can be an effective and economical way as demonstrated in [316].

As shown in Table 2.1, there are few visual recognition datasets designed for online transfer learning (e.g. P3.5, P3.6, P5.3, and P6.2). Most of the current online transfer learning deals with the detection tasks[293] or text recognition tasks[328]. To advance the transfer learning approaches for more broad and realistic applications, it is essential to create a few large-scale datasets for online transfer learning.
Bibliography


[36] Minmin Chen et al. “Marginalized Denoising Autoencoders for Domain Adapta-
774.

[37] Zhiyuan Chen, Nianzu Ma, and Bing Liu. “Lifelong learning for sentiment clas-
sification”. In: Association for Computational Linguistics. 2015.

[38] Dong Seon Cheng et al. “Custom Pictorial Structures for Re-identification.” In:
2011.

[39] Zhongwei Cheng et al. “Human daily action analysis with multi-view and color-
depth data”. In: Proc. European Conference on Computer Vision. Springer. 2012,
pp. 52–61.

Decoder Approaches”. In: Syntax, Semantics and Structure in Statistical Transla-

[41] Myung Jin Choi et al. “Exploiting Hierarchical Context on a Large Database of
Object Categories”. In: Proc. IEEE Conference on Computer Vision and Pattern
Recognition. 2010.

learning for domain adaptation by interpolating between domains”. In: Proc. ICML

[43] Tat-Seng Chua et al. “NUS-WIDE: a real-world web image database from Na-
tional University of Singapore”. In: Proc. ACM international conference on image

[44] Adam Coates, Andrew Ng, and Honglak Lee. “An analysis of single-layer net-
works in unsupervised feature learning”. In: Proc. international conference on arti-

[45] Nicolas Courty et al. “Optimal transport for Domain adaptation”. In: IEEE Trans-
actions on Pattern Analysis and Machine Intelligence (2016).

with a domain specific class means classifier”. In: Proc. European Conference on

[47] Zhen Cui et al. “Flowing on Riemannian manifold: Domain adaptation by shifting

[48] Zhen Cui et al. “Generalized unsupervised manifold alignment”. In: Proc. Ad-


[99] Yves Grandvalet and Yoshua Bengio. “Semi-supervised learning by entropy mini-
mization”. In: Advances in neural information processing systems. 2005, pp. 529–
536.

recognition, reacquisition, and tracking”. In: Proc. IEEE International Workshop
on Performance Evaluation for Tracking and Surveillance. Vol. 3. 5. 2007, pp. 1–
7.


[102] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category

pp. 807–813.

[104] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. “Cross modal distillation for
supervision transfer”. In: Proc. IEEE Conference on Computer Vision and Pattern


[106] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the Knowledge in a
Neural Network”. In: NIPS 2014 Deep Learning Workshop. 2014.

[107] Martin Hirzer et al. “Person re-identification by descriptive and discriminative
2011, pp. 91–102.


adaptation for evolving visual domains”. In: Proc. IEEE Conference on Computer

[110] Judy Hoffman et al. “Discovering latent domains for multisource domain adapta-
715.

[111] Tzu Ming Harry Hsu et al. “Unsupervised Domain Adaptation With Imbalanced
Cross-Domain Data”. In: Proc. IEEE International Conference on Computer Vi-


[237] Si Si, Dacheng Tao, and Bo Geng. “Bregman divergence-based regularization for transfer subspace learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 22.7 (2010), pp. 929–942.


