DOMAIN ADAPTATION ON GRAPHS BY LEARNING ALIGNED GRAPH BASES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET PİLANCI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

MAY 2018

Approval of the thesis:

DOMAIN ADAPTATION ON GRAPHS BY LEARNING ALIGNED GRAPH BASES

submitted by **MEHMET PİLANCI** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronics Engineering** ______________

Assist. Prof. Dr. Elif Vural
Supervisor, **Electrical and Electronics Engineering Department, METU** ______________

Examining Committee Members:

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering Department, METU ______________

Assist. Prof. Dr. Elif Vural
Electrical and Electronics Engineering Department, METU ______________

Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering Department, METU ______________

Prof. Dr. Çağatay Candan
Electrical and Electronics Engineering Department, METU ______________

Assist. Prof. Dr. Cem Tekin
Electrical and Electronics Engineering Department,
İhsan Doğramacı Bilkent University ______________

Date: ______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MEHMET PİLANCI

Signature :

ABSTRACT

DOMAIN ADAPTATION ON GRAPHS BY LEARNING ALIGNED GRAPH BASES

Pilancı, Mehmet
M.S., Department of Electrical and Electronics Engineering
Supervisor : Assist. Prof. Dr. Elif Vural

May 2018, 70 pages

In this thesis, the domain adaptation problem is studied and a method for domain adaptation on graphs is proposed. Given sufficiently many observations of the label function on a source graph, we study the problem of transferring the label information from the source graph to a target graph for estimating the target label function. Our assumption about the relation between the two domains is that the frequency content of the label function, regarded as a graph signal, has similar characteristics over the source and the target graphs. We propose a method to learn a pair of coherent bases on the two graphs, such that the corresponding source and target graph basis vectors have similar spectral content, while “aligning” the two graphs at the same time so that the reconstructed source and target label functions have similar coefficients over the bases. We formulate the basis learning problem as the learning of a linear transformation between the source and target graph Fourier bases so that each source Fourier basis vector is mapped to a new basis vector in the target graph obtained as a linear combination of the target Fourier basis vectors. One synthetic dataset, two image datasets and one book review dataset are used to test the performance of the proposed algorithm. Besides, baseline machine learning methods and recent domain adaptation algorithms are utilized to compare the performance of the proposed algorithm with the methods in the literature. Experiments on several types of data sets suggest that the proposed method compares quite favorably to reference domain adaptation methods. To the best of our knowledge, our treatment is the first to study the domain adaptation problem in a purely graph-based setting with no need for embedding the data in an ambient space. This feature is particularly convenient for many problems of interest concerning learning on graphs or networks.

Keywords: Domain adaptation, data classification, graph Fourier basis, graph Laplacian, spectrum transfer.

ÖZ

HİZALANMIŞ GRAF TABANLARI ÖĞRENEREK GRAFLAR ÜZERİNDE ALAN UYARLAMA

Pilancı, Mehmet
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Dr. Öğr. Üyesi Elif Vural

Mayıs 2018 , 70 sayfa

Bu tezde, alan uyarlaması problemi üzerine çalışılmıştır ve graflar üzerinde alan uyarlaması için bir metot önerilmiştir. Kaynak graf üzerinde yeterli miktarda gözlem bulunduğunda, hedef etiket fonksiyonunu kestirmek için kaynak graf üzerindeki etiket bilgisini hedef grafa taşıma problemi çalışılmıştır. Graf sinyali olarak adlandırılan etiket fonksiyonunun frekans içeriğinin, kaynak ve hedef alanlarda benzer karakteristiğe sahip olduğuna dair bir varsayımımız bulunmaktadır. Önerilen yöntemde iki graf aynı anda hizalanarak kaynak ve hedef etiket fonksiyonları graf tabanları üzerinde benzer katsayılara sahip olacak şekilde kestirilmektedir. Graf tabanları öğrenilirken birbirlerine karşılık gelen kaynak ve hedef vektörlerin benzer spektral içeriğe sahip olmasına dikkat edilmektedir. Taban öğrenme problemi kaynak ve hedef Fourier tabanları arasında lineer bir dönüşüm olarak formüle edilmiştir. Buradaki formülasyonda her bir kaynak Fourier taban vektörü, hedef Fourier taban vektörlerinin lineer kombinasyonlarından elde edilen yeni bir hedef taban vektörüne eşlenmiştir. Önerilen algoritmanın performansını test etmek için bir sentetik veri kümesi, iki ayrı görüntü veri kümesi ve bir adet kitap yorumu veri kümesi kullanılmıştır. Ayrıca, önerilen algoritmanın performansı temel yapay öğrenme algoritmaları ve güncel alan uyarlama algoritmaları ile karşılaştırılmıştır. Farklı tipteki veri kümeleri üzerinde uygulanan deneyler, önerilen metodun referans alan uyarlama metotlarından daha iyi performans gösterdiğini ortaya koymuştur. Bildiğimiz kadarıyla, bizim yaklaşımımız alan uyarlaması probleminde, verileri bir ortam uzayına yerleştirme ihtiyacı olmadan, tamamen graf tabanlı yapılan ilk çalışmadır. Bu özellik, graflar ve ağlar üzerinde öğrenmeye dayalı problemler için bilhassa uygundur.

Anahtar Kelimeler: Alan uyarlama, veri sınıflandırma, graf Fourier tabanı, graf Laplacian, spektrum aktarma.

To my family

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Assist. Prof. Elif Vural for her valuable guidance, encouragement, understanding and tolerance throughout my thesis study. There is no doubt that her continual support has been a significant factor in the completion of this thesis.

I would like to thank my colleagues for their support throughout this thesis work.

I would like to thank my family for their support and love. The completion of this thesis would not be possible without them.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ML	Machine Learning
TL	Transfer Learning
DA	Domain Adaptation
DASGA	Domain Adaptation via Spectral Graph Alignment
EA	Easy Adapt
DAMA	Domain Adaptation Manifold Alignment
SA	Subspace Alignment
GFK	Geodesic Flow Kernel
NN	Nearest Neighbour
SVM	Support Vector Machine
RMS	Root Mean Square

CHAPTER 1

INTRODUCTION

1.1 Motivation

In the last decades, the amount of available data in multimedia, social network, trade, shopping and many other fields has increased enormously with the help of Internet and mobile technologies. There is a stream of data which is being uploaded on Internet at each second, which consists of images, sound records, videos. Besides, there is another type of data growing on Internet each moment, which is user experience data. People make comments on what they bought, used, experienced in order to inform others.

There are various efforts in the machine learning field so as to extract useful information out of this huge amount of data. In machine learning problems, a large number of training data is needed to learn the intrinsic features of the data. However, the amount of available labeled data is limited in many cases and getting labeled data is an expensive work. Therefore, it is helpful to utilize unlabeled data as well, and domain adaptation comes into play at this stage. In domain adaptation, it is aimed to transfer the information gained in one domain, which is called the source domain, to another domain which is called the target domain. In a typical domain adaptation task, there are many labeled instances in the source domain and there are a few or no labeled instances in the target domain. Domain adaptation is accepted as a sub-field of Transfer Learning whose setting is demonstrated in Figure 1.1

In classical machine learning algorithms, training parameters are used to model the probabilistic distribution models of the data and it is assumed that test data also share

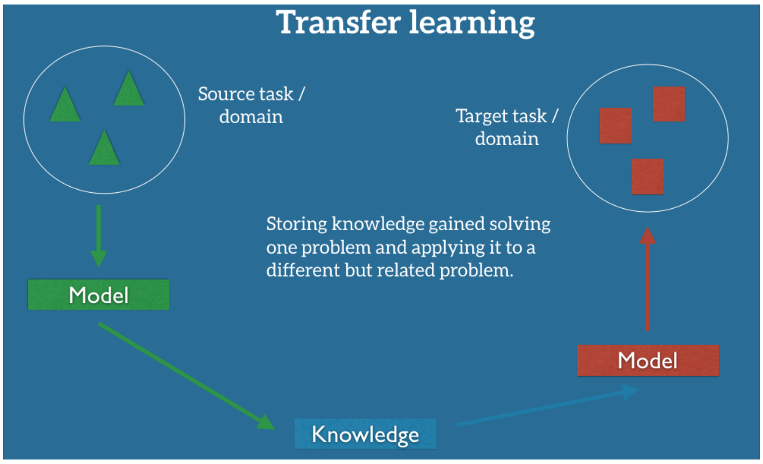
Transfer learning
Source task / domain
Target task / domain
Storing knowledge gained solving one problem and applying it to a different but related problem.
Model
Model
Knowledge

Figure 1.1: Transfer Learning Setting [1]

the same probabilistic distribution. Therefore, if there are a few or no labeled samples in one domain, T, it is not possible to apply traditional machine learning algorithms. However, if there are enough labeled samples in another but related domain, S, domain adaptation techniques can be applied to label the instances in domain T.

A widely known example in domain adaptation is spam filtering in e-mail box. Spam filtering parameters for one specific person are inferred from other public spam e-mailing data. Although public spam e-mail data and the spam e-mails in one specific person's inbox belong to different domains, spam prediction can be applied successfully by using the public data. Another famous example is from natural language processing. Processing text in English can help to process text in German by the help of domain adaptation methods. There are many other domain adaptation applications such as image classification, semantic analysis, etc.

Graph theory models have been used in machine learning and data retrieval researches recently [2]. In machine learning, a graph is used to represent the distribution of data points in a space and how each data point is related to its neighbours. To put it another way, graphs in machine learning provide us an overview of the structure of a dataset. All the points in a graph which was constituted from data points of a dataset have a label. The label change across a graph contains some valuable information. The change characteristics of labels across a graph is retrieved by using graph Fourier transform, which resembles the Fourier Transform in signal processing. That is, graph

Fourier transform is used to identify how fast the label function changes across the graph, which captures the spectral characteristics of the label function.

In this thesis, we propose a new domain adaptation method, where we consider a source graph and a target graph representing the source and target data. We consider the problem of estimating a function, such as a label function, especially on the target graph where very few observations are available. Our assumption about the relation between the source and target domains is that the spectrum, i.e., the frequency content, of the label function has similar characteristics over the source and the target graphs. Then, given the observations of the label function on the source graph, we estimate the label function on the target graph under the prior that its frequency spectrum resembles that of the source graph. Frequency analysis on graph domains is now a well-established framework, thanks to the recent advances in the field of graph signal processing. The convergence of the graph Laplacian operator to the continuous Laplace-Beltrami operator on manifolds has been studied in several previous works [3], [4]. Then, characterizing the Fourier basis vectors as the eigenvectors of the Laplacian operator, the Fourier transform and Fourier bases can be extended to graph domains via the eigenvalue decomposition of the graph Laplacian matrix [5], [6], [7]. In fact, the notion of smoothness, or smoothly-varying functions on graphs has been essential to many dimensionality reduction and semi-supervised learning methods [8], [9], [10] for a long time. Graph-based semi-supervised learning algorithms in a single domain typically rely on the assumption that the label function to be estimated has a smooth variation, i.e., is a slowly changing function, on the graph. This assumption has also been employed in several domain adaptation algorithms benefiting from a graph model, such as in [11], [12], [13]. Meanwhile, the validity of the smoothness assumption is questionable in the general sense. For instance, in Figure 1.2, a generic face manifold is illustrated, where the face images of different individuals may get arbitrarily close to each other due to extreme lighting conditions. It can then be observed that the label function to be estimated has fast variation along certain directions on the data graph, therefore, its spectrum contains some non-negligible high-frequency content as well. While the assumption that the label function should vary slowly on the graph is reasonable especially in a single domain where no information about its spectral content is available, the spectrum can

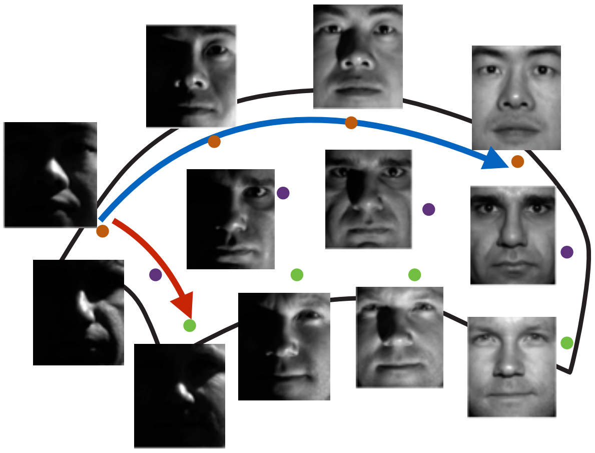

Figure 1.2: Illustration of a generic face manifold. Face images of three different individuals are indicated with different colors. While the class label function varies slowly along the direction shown in blue, it has a relatively fast variation along the red direction. (Face images from Extended Yale face database [14])

actually be learnt in a setting with more than one domain. Our work is then based on the idea of learning the spectral content of the label function from the source graph, and transferring it to the target graph for more accurate estimation.

In this thesis, we propose a novel method for graph domain adaptation that learns a relation between the source and target graphs without assuming any prior information of node correspondences or high similarity between the two graphs. Given a source and a target graph that are independently constructed, we propose to learn a pair of “aligned” bases on the two graphs through which information can be transferred or shared between the two graphs. In particular, the “aligned” source and target bases are such that the coefficients of the source and target label functions when represented in the corresponding bases must be similar. We formulate the basis learning problem as the learning of a linear transformation between the source and target graph Fourier bases so that each source Fourier basis vector is mapped to a new basis vector in the target graph obtained as a linear combination of the target Fourier basis vectors. The learning of this transformation then becomes a key problem of the proposed scheme. In particular, the linear transformation to be learnt must be sufficiently flexible to indeed “align” the two graphs even if they are independently constructed, while retaining the capability of transferring the spectral content of the label function between the two graph bases. In order to achieve this, we impose suitable priors on the linear transformation, and then learn the transformation matrix jointly with the source and target label functions under the constraint that the source and target label functions must have similar coefficients over the learnt bases. The resulting objective function is not jointly convex in the coefficients and the transformation matrix; nevertheless, it is separately convex in one when the other is fixed. We minimize the objective with an alternating optimization procedure.

1.2 Thesis Outline

The goal of this thesis is to study the domain adaptation problem for graph domains, and propose a domain adaptation solution that allows the transfer of knowledge between graph domains.

In Chapter 2, domain adaptation literature is overviewed. The possible solutions to the investigated problems in this work are described. Recent domain adaptation algorithms are introduced and the logic behind them are stated.

In Chapter 3, the technical background for this thesis is provided. At this point, the essential points in graph signal processing are introduced. Firstly, graph Fourier analysis is presented since this is the basics of spectral graph applications. Then, signal representations in the vertex and spectral domains, which are counterparts of the time and frequency domains in traditional signal processing, are described. And then, some useful operators in graph signal processing such as the Laplace operator are defined.

After overviewing the existing solutions in domain adaptation and providing the technical background for graph signal processing in Chapters 2 and 3, our novel algorithm for graph domain adaptation problems is defined in Chapter 4. At this point, the derivation and solution of our optimization problem are stated in details.

In Chapter 5, the performance of our algorithm is tested and compared to other domain adaptation techniques and classical machine learning methods in four different datasets. The first dataset is a synthetic dataset, which is generated for this study in order to examine the baseline performance of the algorithms. Two of the datasets consist of images; one of them aims face recognition and the other one aims object classification. The last dataset consists of user ratings for purchased products in Amazon website.

Lastly, the thesis is summed up in Chapter 6 by stating the important results obtained throughout this study via conducted experiments and the points that can be improved in the future.

CHAPTER 2

RELATED WORK

2.1 Introduction

Firstly, *transfer learning* and *domain adaptation* concepts are introduced and some background information for these fields are provided in this chapter. Since the proposed algorithm in this thesis is based on aligning spectral bases of source and target graph domains, which is a novel approach in the literature, there are no studies which address exactly the same problem as in this thesis. Therefore, a brief overview of the domain adaptation literature is presented in this chapter.

In the machine learning literature, there are some inconsistencies about the use of the *domain adaptation* and the *transfer learning* terms. Some authors even use them interchangeably. However, the prevalent acceptance is used throughout this study as defined in [15] and [16]. A domain $\mathcal{D}$ has a feature space with dimension of d, $\mathcal{X} \subset R^d$ and a marginal probability distribution $P(X)$ where $X = \{x_1, x_2, \ldots, x_n\}$ is a random vector and x_i's are data samples. A task $\mathcal{T}$ is defined on $\mathcal{X}$ by a label space denoted as $\mathcal{Y}$ with the conditional probability function $P(Y|X)$ where $Y = \{y_1, y_2, \ldots, y_n\}$ is a random vector. The main aim of machine learning problems is to find a function, f, which maps each sample to a label in $\mathcal{Y}$. That is, the goal is to find a function which satisfies $f(x_i) = y_i$, $\forall i$.

As mentioned above, there are two domains in domain adaptation problems; the source domain and the target domain. Therefore, the idea in the previous paragraph can be extended to these two domains. Let us denote the source domain as $\mathcal{D}^s = \{\mathcal{X}^s, P(X^s)\}$ and a task on the source domain as $\mathcal{T}^s = \{\mathcal{Y}^s, P(Y^s|X^s)\}$. Similarly, we can define a target domain as $\mathcal{D}^t = \{\mathcal{X}^t, P(X^t)\}$ and a task on the target domain as $\mathcal{T}^t = \{\mathcal{Y}^t, P(Y^t|X^t)\}$. In classical machine learning problems, the source and target domains are accepted to be the same, which implies that $\mathcal{D}^s = \mathcal{D}^t$ and $\mathcal{T}^s = \mathcal{T}^t$.

In the cases where the source domain and the target domain are not the same, $\mathcal{D}^s \neq \mathcal{D}^t$, or the source task and the target task are not the same, $\mathcal{T}^s \neq \mathcal{T}^t$, classical machine learning approaches cannot be applied successfully. In such situations, it may be possible to learn $P(Y^t|X^t)$ by leveraging the information in $\{\mathcal{D}^s,\ \mathcal{T}^s\}$ and this process is called transfer learning (TL) [17].

In domain adaptation, which is classified as a particular application of transfer learning [16], the source and target tasks are assumed to be the same, i.e., $\mathcal{T}^s = \mathcal{T}^t$. However, domain adaptation is expanded to the case where only the $\mathcal{Y}^s = \mathcal{Y}^t$ requirement holds, that is, the second requirement, $P(Y|X^t) = P(Y|X^s)$, is relaxed generally. The taxonomy of transfer learning methods presented by Pan *et al.* is given in Figure 2.1.

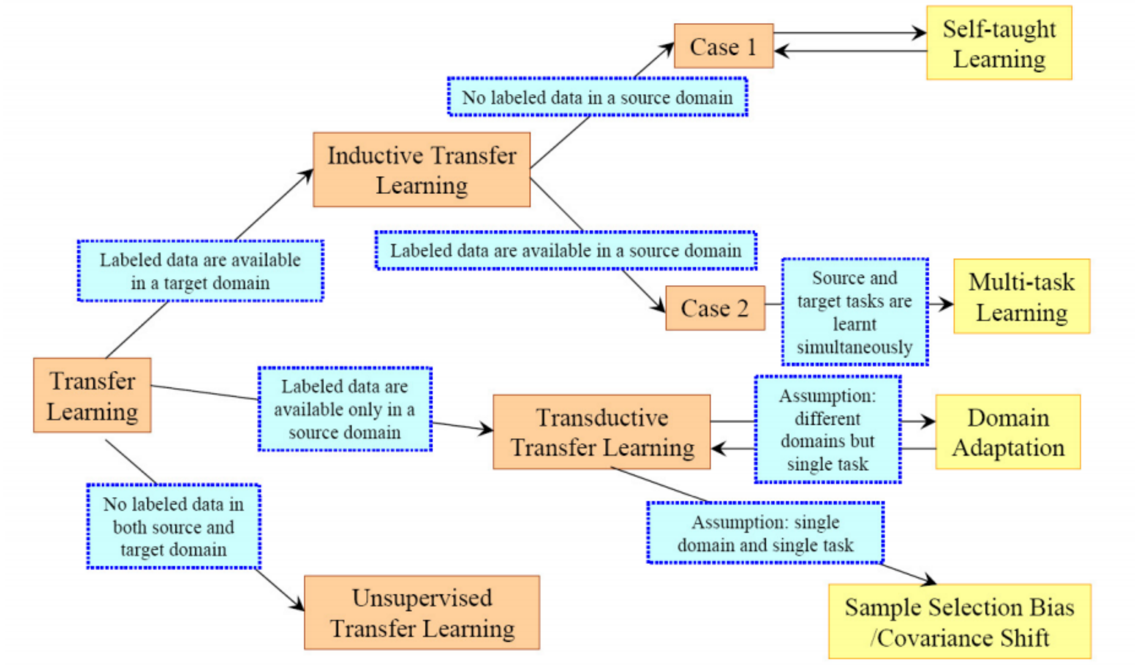
Case 1
Self-taught Learning
No labeled data in a source domain
Inductive Transfer Learning
Labeled data are available in a target domain
Labeled data are available in a source domain
Case 2
Source and target tasks are learnt simultaneously
Multi-task Learning
Transfer Learning
Labeled data are available only in a source domain
Transductive Transfer Learning
Assumption: different domains but single task
Domain Adaptation
No labeled data in both source and target domain
Assumption: single domain and single task
Unsupervised Transfer Learning
Sample Selection Bias /Covariance Shift

Figure 2.1: An overview of transfer learning settings [16]

In domain adaptation literature, the *unsupervised* notion is used for the situation where the labels are only available from the source domain and the *semi-supervised* notion is used for the situation where the labels are available from both the source and the target domains. Note that these concepts are not the same as their usage in traditional machine learning literature.

2.2 Overview of Domain Adaptation Literature

Domain adaptation methods can be discussed under two subtopics; *homogeneous domain adaptation* and *heterogeneous domain adaptation*. In section 2.2.1, homogeneous domain adaptation methods are introduced in which source and target data representations are the same, $\mathcal{X}^s = \mathcal{X}^t$, [18], [19], [20]. In section 2.2.2, heterogeneous domain adaptation methods are stated in which source and target data representations are different, $\mathcal{X}^s \neq \mathcal{X}^t$.

2.2.1 Homogeneous Domain Adaptation Techniques

A group of domain adaptation methods are based on instance re-weighting in which, it is assumed that conditional distributions are shared between the source and target domains, $P(Y|X^s) = P(Y|X^t)$, [18], [21]. Since $P(X^t) \neq P(X^s)$ in many cases, direct application of a source model cannot provide succesful results. One of the methods that addresses this problem is Selective Transfer Machine (STM) which tries to optimize instance weights and classifier parameters jointly [22]. A classification example of the STM algorithm is illustrated in Figure 2.2. Another instance re-weighting method is Adaptive Boosting (AdaBoost) [23] which is proposed to improve the performance of a classifier by increasing the weights of misclassified target samples. AdaBoost is improved by Transfer Adaptive Boosting (TrAdaBoost) [24], which decreases the weights of misclassified source samples so as to lower their effect on the classifier.

Trying to adapt classifier parameters is stated as another approach for domain adaptation [25], [26], [27], [28], [29], [30]. Joachims [25] improved the performance of classical SVM through leveraging information from the target domain in the optimization of SVM. Yang *et al.* proposed a method called Adaptive Support Vector

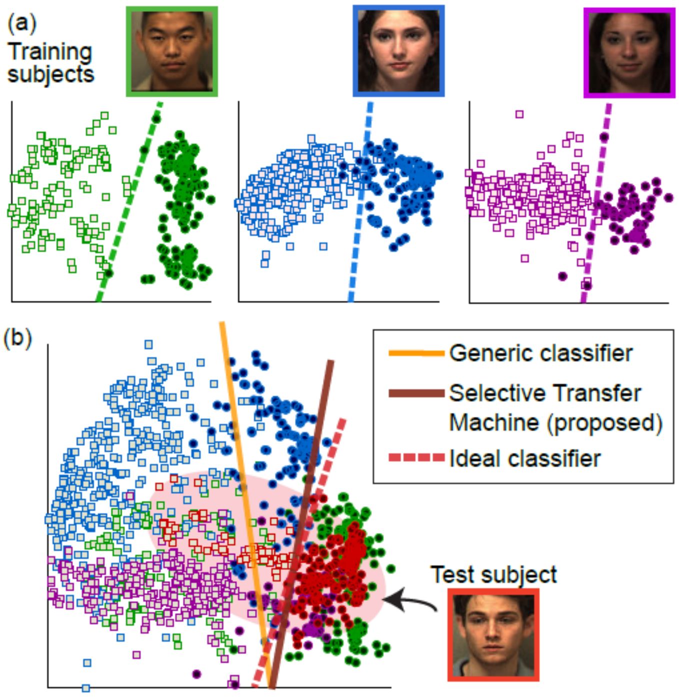

Figure 2.2: Selective Transfer Machine Algorithm Overview [22]

Machines (A-SVMs) to obtain an ensemble classifier from auxiliary classifiers which are optimized for different domains [31]. Jiang *et al.* proposed another approach in order to increase the performance of SVM, which is called Cross-Domain SVM (CDSVM) [29]. In CDSVM, an SVM classifier is found for the source domain, and then, support vectors obtained for the source domain are included in the target domain data. Finally, a new SVM classifier is obtained for the dataset, which consists of target samples and support vectors from source samples.

Feature augmentation methods are also utilized for domain adaptation purposes [32], [33], [34], [35], [36], [37]. In [32], Daumé proposed a practical method called Easy Adapt(EA) to adapt the source and target domains. He augmented source domain features as $\left[\begin{smallmatrix} x^s \\ x^s \\ 0 \end{smallmatrix}\right]$ and target domain features as $\left[\begin{smallmatrix} x^t \\ 0 \\ x^t \end{smallmatrix}\right]$. And then, classical SVM steps are applied on the augmented source and target datasets. EA algorithm is enhanced by Daumé *et al.* with Easy Adapt++ (EA++) in [33]. As an improvement on EA, EA++ utilizes also unlabeled data samples via mapping them as $\left[\begin{smallmatrix} 0 \\ x^u \\ -x^u \end{smallmatrix}\right]$.

Geodesic Flow Sampling (GFS) is another feature augmentation based algorithm, which was proposed in [35], [36]. In GFS, the source and target spaces are initially found by PCA, then, they are viewed as points on a Grassmann manifold. After that, a geodesic path is found out between these two points. The found geodesic path is sampled at some finite points by which intermediate subspaces are obtained. Finally, labeled and unlabeled data are projected on these intermediate subspaces for classification. The GFS algorithm is illustrated in Figure 2.3. The Geodesic Flow Kernel (GFK) algorithm enhanced GFS by sampling an infinite number of points on the geodesic path [37].

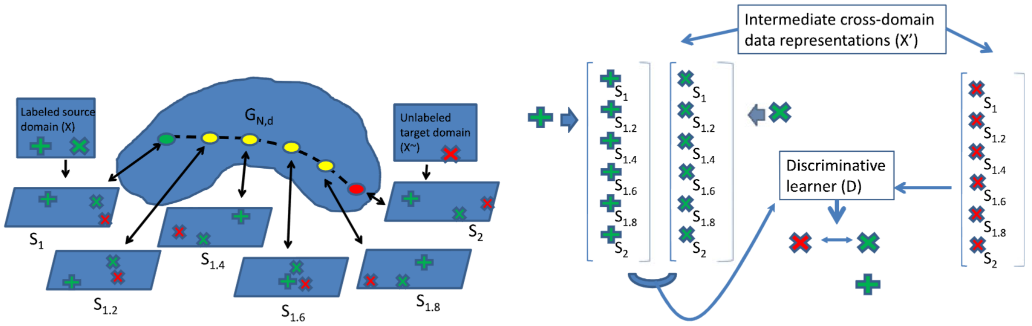
Labeled source domain (X)
$G_{N,d}$
Unlabeled target domain (X~)
S_1
$S_{1.2}$
$S_{1.4}$
$S_{1.6}$
$S_{1.8}$
S_2
Intermediate cross-domain data representations (X')
Discriminative learner (D)

Figure 2.3: Geodesic Flow Sampling Algorithm. Sampling of intermediate points on geodesic path and sample mapping. [36]

Aligning the source and target feature spaces is used as another idea for domain adaptation [38], [39], [40], [41], [42]. Fernando *et al.* proposed the Subspace Alignment (SA) algorithm, which applies feature space alignment, in [42]. In SA, source and target subspaces are computed by a PCA and the subspaces are aligned via learning a transformation matrix. Since projecting each sample to another domain is not necessary in SA, its implementation is very straightforward. Sun *et al.* proposed the Correlation Alignment (CORAL) method which uses the idea of feature space alignment as well [43]. In CORAL, second order statistics of source and target data are utilized. Firstly, source data is whitened with the source covariance matrix and then the whitened source data is re-coloured with target covariance. Sun *et. al* claims that their algorithm is more "frustratingly easy" than Easy Adapt (EA) introduced in [32] since it does not require any labeled data in target domain.

Unsupervised feature transformation is also stated as an idea for domain adaptation [44], [45], [46], [47], [48]. Pan *et al.* proposed to map features instead of subspaces in the Transfer Component Analysis (TCA) method [44]. TCA algorithm tries to discover a latent space in which the marginal distributions of the source and target domains do not change. A mapping from source and target features to the latent space is found out, and then, classical machine learning algorithms can be applied. Long *et al.* proposed the Transfer Joint Matching (TJM) method which applies feature matching in a reproducing kernel Hilbert space and instance reweighting jointly [48].

2.2.2 Heterogeneous Domain Adaptation Techniques

Heterogeneous domain adaptation (HDA) is similar to heterogeneous transfer learning (HTL). However, data from different domains which have different representations are available both in the training and test stages of HDA while one data representation is available in the training stage of HTL and another data representation is used in its test stage. Both HDA and HTL notions are associated with multi-view learning, which enables us to learn better representations from multiple source domains [49], [50]. Multiple domains can come, for instance, from image and text representations [51], [52], [53], or text belonging to different languages [54, 55].

One of the approaches developed for heterogeneous domain adaptation problems is to use auxiliary domains [56], [57], [58], [59], [60], [61]. Tan *et al.* proposed to find an auxiliary domain that contains features of source and target domains in their algorithm Transitive Transfer Learning (TTL) [56]. For instance, if the source domain consists of image data and target domain consists of images, TTL algorithm uses another domain which includes both text and image data, e.g., crawled Web page data, in order to combine the source and target domains. The TTL algorithm is demonstrated in Figure 2.4. Mixed-Transfer method obtains a model of relation between the source and target domains by a joint transition probability graph of mixed instances and features [58]. A text to image heterogeneous transfer learning set-up is illustrated in Figure 2.5.

Symmetric feature transformation is another approach used for heterogeneous domain adaptation [36], [62], [63], [64], [65], [13], [11]. The goal of feature transformation for HDA is to map source and target spaces into a common latent feature space. Duan *et al.* proposed to transform source and target domains into a common latent

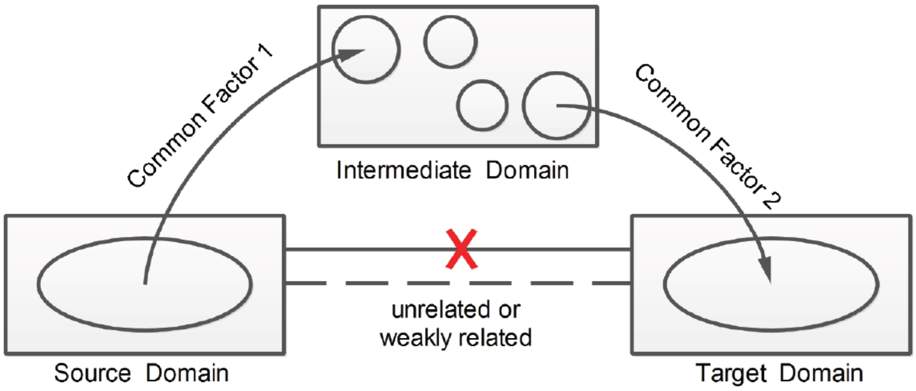

Figure 2.4: Transitive Transfer Learning Algorithm. [56]

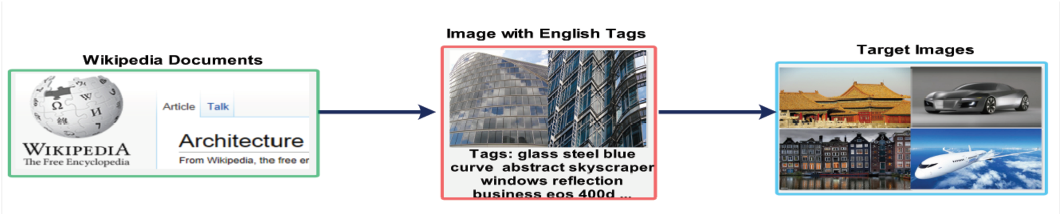
Wikipedia Documents
WIKIPEDIA
The Free Encyclopedia
Article
Talk
Architecture
From Wikipedia, the free er
Image with English Tags
Tags: glass steel blue curve abstract skyscraper windows reflection business eos 400d ...
Target Images

Figure 2.5: Text-to-image heterogeneous learning example. [58]

space, and then apply feature augmentation for the problem of HDA in their algorithm Heterogeneous Feature Augmentation (HFA) [63]. Wang *et al.* stated a method called Domain Adaptation Manifold Alignment (DAMA) in which they map source and target spaces into a latent space such that the underlying structure of each domain is preserved and the samples having the same label are located close [62]. DAMA does not require the domains to have common features, it utilizes the common labels instead. Besides, DAMA can be used for the case where there are multiple source domains. The rationale behind DAMA is illustrated in Figure 2.6.

The last approach for heterogeneous domain adaptation is asymmetric feature transformation in which source features are projected into the target space so as to decrease the distribution difference for data coming from the source and target domains [66], [67], [68]. Kulis *et al.* proposed a method which learns an asymmetric non-linear transformation to map the source domain into the target domain in [67]. Harel *et al.* developed a method called Multiple Outlook MAPping (MOMAP) whose goal is to map a domain with a large number of labeled data to another domain with a small

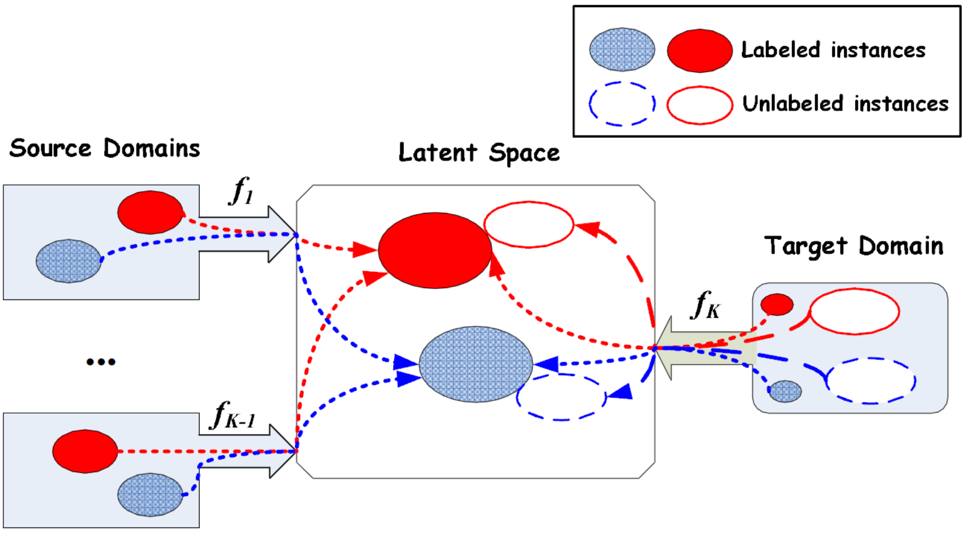

Figure 2.6: DAMA Algorithm. Different colors represent different classes. [36]

number of labeled data [68]. MOMAP learns a transformation from labeled data by the singular value decomposition process which aims to match marginal distributions of classes and preserve the data structure.

CHAPTER 3

AN OVERVIEW OF GRAPH SPECTRAL PROPERTIES

3.1 Introduction

Graphs are frequently used to visualize high dimensional data. They help to understand relational structure of the available data, therefore they are used in many fields such as transportation, social networking, energy and neural networks. A graph consists of vertices and edges, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Vertices present data samples in a dataset and edges define the relationship between two data samples.

Graph signal processing employs techniques on graphs so as to extract intrinsic information of the data. In this chapter, firstly, graph signals are introduced. And then, graph Laplacian operator and Fourier transform on graphs are defined.

3.2 Weighted Graphs and Graph Signals

In weighted graphs, there is a quantity assigned to the edges, which indicates the similarity of the vertices at the two ends of that edge. The main interest of this thesis is undirected and weighted graphs, therefore, the graph is represented by three components: $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, W\}$ where $\mathcal{V} = \{x_i\}_{i=1}^{N}$ denotes vertices, $\mathcal{E}$ denotes edges and $W \in \mathbb{R}^{N \times N}$ denotes the weight matrix of the graph. If there is an edge between the nodes x_i and x_j, then the (i, j)-th element of W_{ij} is the weight of this edge. If the nodes x_i and x_j are not connected with an edge, then $W_{ij} = 0$.

If the edge weights are not defined in the nature of the data, the weights can be computed using Gaussian kernel weighting function:

$$W_{ij} = \begin{cases} \exp\left(-\frac{[dist(i,j)]^2}{2\theta^2}\right) & \text{if } dist(i,j) \leq \mathcal{K} \\ 0 & \text{otherwise} \end{cases}, \tag{3.1}$$

for pre-defined parameters θ and $\mathcal{K}$. $dist(i,j)$ is the distance between two nodes, e.g., Euclidean distance can be used. Moreover, k-nearest neighbours method can be used as well for edge limitation.

A graph signal is a function $f : \mathcal{V} \rightarrow \mathbb{R}$ taking a real value on each vertex of the graph, which can equivalently be represented as an N-dimensional vector $f \in \mathbb{R}^N$. An example of graph function is demonstrated in Figure 3.1.

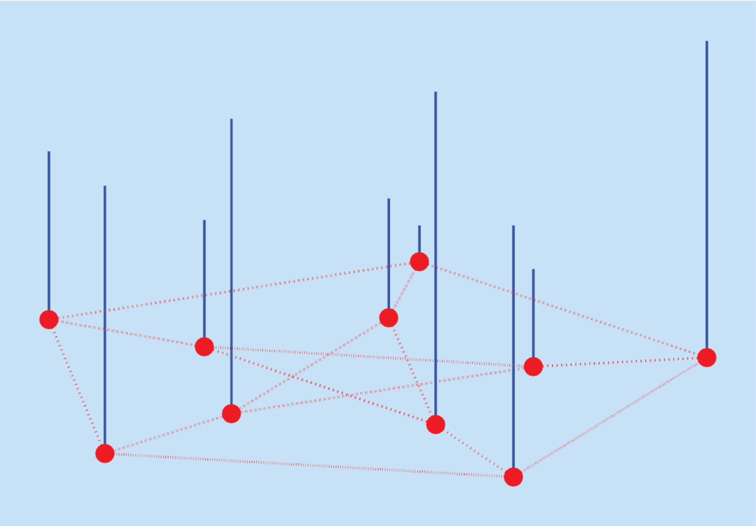

Figure 3.1: A random positive graph signal on the vertices of the Petersen graph. The height of each blue bar represents the signal value at the vertex. [6]

A set $\{v_k\}_{k=1}^N \subset \mathbb{R}^N$ of linearly independent graph signals forms a graph basis, so that any graph signal f can be represented as

$$f = \sum_{k=1}^{N} \alpha_k v_k$$

in terms of the graph basis vectors v_k with coefficients α_k. Representing the basis as a matrix $V = [v_1 \dots v_N] \in \mathbb{R}^{N \times N}$ and the coefficient vector as $\alpha = [\alpha_1 \dots \alpha_N]^T \in \mathbb{R}^N$, the graph signal can be expressed as $f = V\alpha$.

3.3 Graph Laplacian and Graph Fourier Transform

The graph Laplacian matrix is defined as $L = D - W$, where D is the diagonal degree matrix given by $D_{ii} = \sum_j W_{ij}$. The graph Laplacian is an essential element of spectral graph theory, since its application to a graph signal f as an operator via the matrix multiplication

$$(Lf)(x_i) = \sum_{j=1}^{N} W_{ij}(f(x_i) - f(x_j))$$

is the graph equivalent of applying the Laplacian operator to a signal in classical signal processing [3], [4], [6]. This analogy allows the extension of Fourier analysis to graph domains as follows. First recall that the complex exponentials $e^{j\Omega t}$ defining the Fourier transform of one dimensional signals in classical signal processing are given by the eigenfunctions of the Laplacian operator Δ for one-dimensional signals

$$-\Delta(e^{j\Omega t}) = \Omega^2 e^{j\Omega t}. \tag{3.2}$$

The eigenvalue Ω^2 of the Laplacian operator increases with the frequency of the complex exponential $e^{j\Omega t}$. Characterizing the Fourier transform via the eigenfunctions of the Laplacian operator, the graph counterpart of complex exponentials are then the eigenvectors of the graph Laplacian given by

$$Lu_k = \lambda_k u_k.$$

The set of eigenvectors $\{u_k\}_{k=1}^{N}$ of the graph Laplacian corresponding to the eigenvalues $\lambda_1 = 0 \leq \lambda_2 \leq \cdots \leq \lambda_N$ thus defines a graph Fourier basis. In analogy with (3.2), the eigenvalues λ_k bear a notion of frequency over the graph domain. The eigenvectors u_k for increasing values of k indeed have an increasing speed of variation over the graph when regarded as graph signals [6]. This phenomenon can be observed in Figures 3.2 and 3.3. As can be seen in Figure 3.3, as the eigenvalue increases, the number of zero crossings in the associated eigenvectors also increases.

In particular, a common measure for the speed of variation of a graph signal f over the graph is

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{N} W_{ij}(f(x_i) - f(x_j))^2,$$

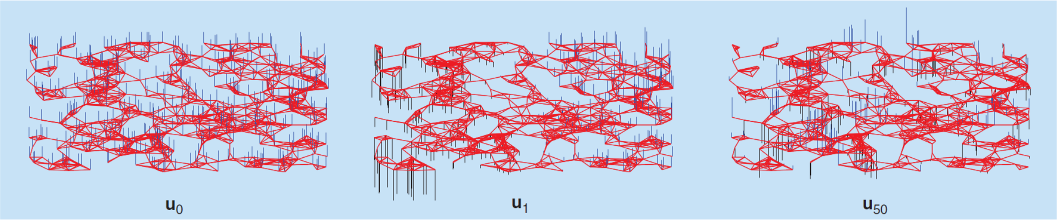

Figure 3.2: Three graph Laplacian eigenvectors of a random sensor network graph. The signals' component values are represented by the blue (positive) and black (negative) bars coming out of the vertices [6]

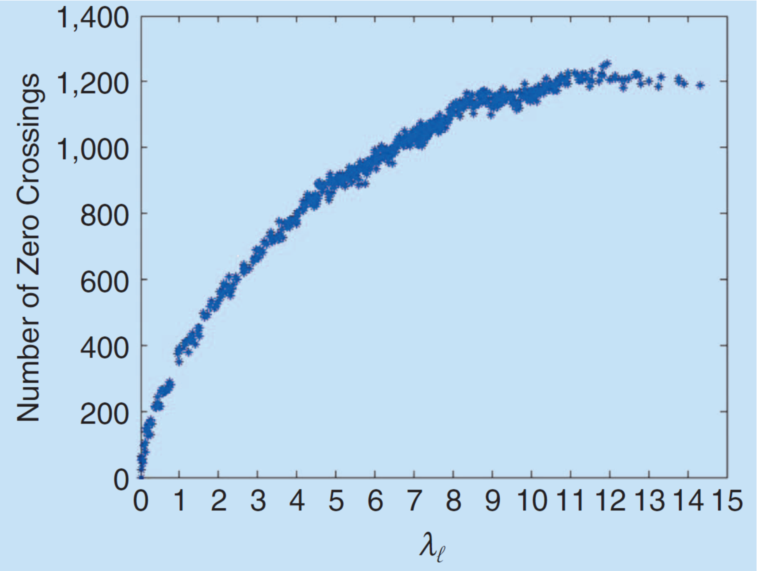

Figure 3.3: The number of zero crossings of the graph Laplacian eigenvectors for the random sensor network graph of Figure 3.2. [6]

which takes larger values if a function f varies more abruptly between neighboring graph vertices. The above term becomes the corresponding eigenvalue λ_k of the graph Laplacian when the graph signal is taken as a Fourier basis vector $f = u_k$

$$u_k^T L u_k = \lambda_k.$$

Once the Fourier basis $\{u_k\}_{k=1}^N$ for graph signals is found, the graph Fourier transform $\hat{f}(\lambda_k)$ of a graph signal f is simply given by its inner product with the basis vectors

$$\hat{f}(\lambda_k) = \langle f, u_k \rangle = \sum_{i=1}^{N} f(x_i) u_k(x_i),$$

or simply as $\hat{f} = U^T f$ in matrix notation, where $\hat{f} = [\hat{f}(\lambda_1) \dots \hat{f}(\lambda_N)]^T$ and $U = [u_1 \dots u_N]$. Here $\hat{f}(\lambda_k)$ is the Fourier coefficient of f corresponding to the basis vector u_k with frequency λ_k. The inverse Fourier transform is then obtained as the reconstruction of the signal from its representation over the Fourier basis as

$$f = \sum_{k=1}^{N} \hat{f}(\lambda_k) u_k = U\hat{f}.$$

3.4 Conclusion

In this chapter, graph operators, which will be essential to conceive the rest of the thesis, are introduced. The structure of a weighted graph, graph Laplacian and graph Fourier transform notions, which are the bases of this thesis, are defined. In fact, there are numerous signal processing methods on graphs such as filtering and convolution which can be observed in [6] in details.

CHAPTER 4

DOMAIN ADAPTATION VIA SPECTRAL GRAPH ALIGNMENT

4.1 Introduction

A novel domain adaptation method called Domain Adaptation via Spectral Graph Alignment (DASGA) is introduced in this chapter. Firstly, the problem formulation for domain adaptation on graphs is given in details. And then, the derivation of DASGA is presented step by step by stating the optimization processes. Finally, complexity analysis of the proposed method is provided in this chapter.

4.2 Problem Formulation for Domain Adaptation on Graphs

In this section, we propose our problem formulation for domain adaptation in graph settings. We consider a source graph $G^s = (\mathcal{V}^s, \mathcal{E}^s, W^s)$ that consists of N_s vertices $\mathcal{V}^s = \{x_i^s\}_{i=1}^{N_s}$ and edges $\mathcal{E}^s$, and a target graph $G^t = (\mathcal{V}^t, \mathcal{E}^t, W^t)$ with N_t vertices $\mathcal{V}^t = \{x_i^t\}_{i=1}^{N_t}$ and edges $\mathcal{E}^t$. The weighted edges of the source and target graphs are respectively represented in the weight matrices W^s, W^t.

Consider a set of available observations $y_i^s = f^s(x_i^s)$ of a function f^s on the source graph for a subset of source data indexed by $i \in I^s \subset \{1, \dots, N_s\}$, and a set of available observations $y_i^t = f^t(x_i^t)$ of a function f^t on the target graph for a subset of target data indexed by $i \in I^t \subset \{1, \dots, N_t\}$. The functions f^s and f^t take discrete label values in a classification problem and continuous values in a regression problem.

Domain adaptation methods often focus on a setting with many labeled samples in the source domain and much fewer labeled samples in the target domain, i.e., $|I^t| \ll |I^s|$.

Let V^s and V^t denote a pair of bases for the functions on the source and target graphs respectively. We can then decompose the label functions f^s and f^t to be predicted in the source and target graphs over the bases V^s and V^t as

$$f^s = \sum_{k=1}^{N_s} \alpha_k^s v_k^s = V^s \alpha^s, \qquad f^t = \sum_{k=1}^{N_t} \alpha_k^t v_k^t = V^t \alpha^t.$$

Here $V^s \in \mathbb{R}^{N_s \times N_s}$ and $V^t \in \mathbb{R}^{N_t \times N_t}$ correspond to the matrix representations of the bases consisting of the basis vectors $\{v_k^s\}$, $\{v_k^t\}$; and $\alpha^s \in \mathbb{R}^{N_s}$ and $\alpha^t \in \mathbb{R}^{N_t}$ are the coefficient vectors.

Domain adaptation methods assume the presence of a relationship between the source and the target domains and aim to transfer the knowledge in the source domain to the target domain in order to better predict the target label function. In the following, we consider a domain adaptation setting where a relationship can be established between the source and target domains via a “coherent” pair of bases V^s, V^t for the space of functions on the source and the target graphs. In particular, if V^s and V^t are a “coherent” pair of bases, then one can transfer the label information from the source graph to the target graph based on the representations of the label functions on these bases. We can then formulate the following problem:

Problem 1.

$$\min_{\alpha^s, \alpha^t} \|S^s V^s \alpha^s - y^s\|^2 + \|S^t V^t \alpha^t - y^t\|^2 + \mu \|\overline{\alpha}^s - \overline{\alpha}^t\|^2 \tag{4.1}$$

Here y^s and y^t are vectors consisting of the available labels $\{y_i^s\}$, $\{y_i^t\}$ in the source and target domains; S^s and S^t are binary selection mask matrices that enforce the label prediction functions f^s, f^t to match the given labels y^s, y^t on the subsets I^s, I^t of labeled data in the source and target domains; and $\mu > 0$ is a weight parameter. The coefficients α^s and α^t of the source and target label functions must be found such that the resulting estimation of the label predictions correspond to the given labels, while α^s and α^t (or their appropriately restricted versions $\overline{\alpha}^s$, $\overline{\alpha}^t$ in the case that the graph sizes are different $N_s \neq N_t$) are close over the source and target graphs.

Then, an important question is what properties a “coherent” pair of bases V^s and V^t should have, and how such bases can be found in practice. If a one-to-one match between the source and target graphs exists, e.g., as in a problem where each source node has a known corresponding target node, then one can simply select the bases as the source and target graph Fourier bases $V^s = U^s$, $V^t = U^t$, so that the spectra of the source and target label functions can be directly matched by solving the problem in (4.1). However, in a realistic setting such a one-to-one match often does not exist. In this work, we propose to learn V^s, V^t relying on the available observations of the label function, in a manner that allows the transfer of the spectral content between the graphs as well. In particular, we propose to choose V^s as the source Fourier basis, and V^t as a target basis expressed as

$$V^s = U^s, \quad V^t = U^t T.$$

Here the matrix $T \in \mathbb{R}^{N_t \times N_t}$ represents a transformation between the target bases U^t and V^t. In the formulation in Problem 1, one can observe that such a transformation matches the source basis vector $v_i^s = u_i^s$ to the following target basis vector

$$v_i^t = \sum_{j=1}^{N_t} T_{ji} u_j^t \tag{4.2}$$

obtained as the linear combination of the target Fourier basis vectors u_j^t via transformation T, where $(\cdot)_{ij}$ denotes the element of a matrix at the i^{th} row and the j^{th} column.

When learning the transformation T, our purpose is to learn a representation that is flexible enough to properly “align” the two individually constructed graphs, while also preserving the spectral relation between the two graphs. The rate of variation of the i-th source Fourier vector $v_i^s = u_i^s$ is proportional to the i-th eigenvalue λ_i^s of the source graph Laplacian L^s. In order to preserve the spectral relation between the graphs, the corresponding target vector v_i^t in (4.2) must have a similar rate of variation on the target graph, so that slowly (or rapidly) varying source label functions are matched to slowly (or rapidly) varying target label functions when solving (4.1). In order to achieve this, we propose to learn T such that the weight T_{ji} of the j-th target Fourier vector u_j^t in the representation of v_i^t is encouraged to be higher for j values close to i, and to decay as j deviates from i. In this way, the source Fourier vector

$u_i^s = v_i^s$ is mapped to a target vector that is mainly composed of the target Fourier vectors u_j^t having frequencies close to that of u_i^s. This can be achieved by penalizing high magnitudes for the entries of T distant from the diagonal, by including a term $\|M \odot T\|^2$ in the overall objective, where $M \in \mathbb{R}^{N_t \times N_t}$ is a symmetric weight matrix of the form

$$M_{ij} = \exp\left(\frac{(i-j)^2}{\sigma^2}\right), \tag{4.3}$$

the scale parameter σ adjusts the width of the window of allowed target frequencies $\{\lambda_j^t\}$ to be matched to a given frequency λ_i^s, and $\odot$ denotes the Hadamard (element-wise) product between two matrices. The overall objective function to minimize then becomes the following:

Problem 2.

$$\begin{aligned} \min_{\alpha^s, \alpha^t, T} & \|S^s U^s \alpha^s - y^s\|^2 + \|S^t U^t T \alpha^t - y^t\|^2 \\ & + \mu_1 \|\alpha^s - \alpha^t\|^2 + \mu_2 \|M \odot T\|_F^2 \\ & \text{subject to } \sum_{i=1}^{N_t} T_{ij}^2 = 1, \text{ for } j = 1, \ldots, N_t. \end{aligned} \tag{4.4}$$

Here $\mu_1 > 0$, $\mu_2 > 0$ are weight parameters, and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. The purpose of the equality constraints in the above problem is to ensure that the columns of the transformation matrix T have unit norm, in order to avoid approaching the trivial solution $T = 0$ which also causes ambiguous solutions for α^t.

While Problem 2 aims to learn a pair of matched full bases on the two graphs, it is often not necessary to use all basis vectors in order to obtain a good reconstruction of the label function: The basis vectors u_i^s, u_i^t with very high frequencies (eigenvalues) λ_i^s, λ_i^t, have a quite rapid variation over the graph, and discarding some of these not only reduces the complexity of the problem, but also serves the important purpose of regularization. For these reasons, it is often useful to select a subset of the basis vectors $\{u_i^s\}_{i=1}^R$, $\{u_i^t\}_{i=1}^R$, corresponding to the smallest R eigenvalues in both domains, where $R < N_s$ and $R < N_t$. Let $\overline{U}^s \in \mathbb{R}^{N_s \times R}$, $\overline{U}^t \in \mathbb{R}^{N_t \times R}$ denote the reduced source and target Fourier bases consisting of the first R basis vectors. When label functions are reconstructed with the reduced bases, we can reformulate the problem in (4.4) as

Problem 3.

$$\begin{aligned}\min_{\overline{\alpha}^s, \overline{\alpha}^t, \overline{T}} & \|S^s \overline{U}^s \overline{\alpha}^s - y^s\|^2 + \|S^t \overline{U}^t \overline{T} \overline{\alpha}^t - y^t\|^2 \\ & + \mu_1 \|\overline{\alpha}^s - \overline{\alpha}^t\|^2 + \mu_2 \|\overline{M} \odot \overline{T}\|_F^2 \\ & \text{subject to } \sum_{i=1}^{R} \overline{T}_{ij}^2 = 1, \text{ for } j = 1, \ldots, R.\end{aligned} \tag{4.5}$$

Here, the matrix $\overline{T} \in \mathbb{R}^{R \times R}$ is the submatrix of T consisting of its first R rows and columns, which match the source vectors $\{u_i^s\}_{i=1}^R$ to linear combinations of $\{u_i^t\}_{i=1}^R$. The reduced weight matrix $\overline{M} \in R^{R \times R}$ has entries as defined in (4.3). The vectors $\overline{\alpha}^s$, $\overline{\alpha}^t$ denote the coordinates of the label functions in the reduced bases $\overline{U}^s$, $\overline{U}^t$ such that the source and target label functions f^s and f^t are reconstructed as

$$f^s = \overline{U}^s \overline{\alpha}^s, \quad f^t = \overline{U}^t \overline{T} \overline{\alpha}^t$$

once the problem in (4.5) is solved. Note that, although the main focus in domain adaptation is to estimate the target labels, the above formulation also allows the estimation of the missing source labels in case of interest.

Estimating the label functions by solving Problem 3, one may then wonder how well the variations of the source and target label functions on the two graphs agree. In the following, we provide an upper bound on the difference between the rates of change of the source and target label functions f^s and f^t. Let $0 = \lambda_1^s \leq \lambda_2^s \leq \cdots \leq \lambda_R^s$ and $0 = \lambda_1^t \leq \lambda_2^t \leq \cdots \leq \lambda_R^t$ respectively denote the smallest R eigenvalues of the source and target graph Laplacians L^s and L^t. Let the similarity of the source and target graph topologies be so that the deviation between the corresponding eigenvalues of the two graph Laplacians are bounded as $|\lambda_i^s - \lambda_i^t| \leq \delta$, for all $i = 1, \ldots, R$. Let us define $\lambda_R = \max(\lambda_R^s, \lambda_R^t)$, which indicates a spectral upper bound (bandwidth) for the frequencies of the first R source and target Fourier basis vectors. Let the coefficients $\overline{\alpha}^s, \overline{\alpha}^t$ and the transformation matrix $\overline{T}$ obtained by solving Problem 3 be such that the difference between the source and target coefficients is bounded as $\|\overline{\alpha}^s - \overline{\alpha}^t\| \leq \Delta_\alpha$, and the deviation between the transformation $\overline{T}$ and the $R \times R$ identity matrix I is bounded as $\|\overline{T} - I\| \leq \Delta_T$, with $\| \cdot \|$ denoting the operator norm for matrices. Finally let C be a bound for the norms of the computed coefficients with $\|\overline{\alpha}^s\|, \|\overline{\alpha}^t\| \leq C$. We then have the following result.

Proposition 1. *Assume that the constants $\lambda_R > 0$, $\delta \geq 0$, $\Delta_T \geq 0$, $\Delta_\alpha \geq 0$, and $C > 0$ are such that the above conditions hold for the solution $\overline{\alpha}^s$, $\overline{\alpha}^t$, $\overline{T}$ of Problem 3. Then, the difference between the rates of variation of the estimated source and target label functions f^s, f^t on the source and target graphs is bounded as*

$$|(f^s)^T L^s f^s - (f^t)^T L^t f^t| \leq C^2\delta + 2C\lambda_R\Delta_\alpha + C^2\lambda_R(2\Delta_T + \Delta_T^2).$$

The proof of Proposition 1 is given in Appendix A. In the light of this theoretical bound, the formulation proposed in Problem 3 can be interpreted as follows. In the considered setting, due to the assumption of the similarity of their spectra, the source and target label functions must have similar rates of variation over the two graphs. The bound in Proposition 1 shows that the source and target label functions have similar rates of variation if the constants δ, λ_R, Δ_α, Δ_T are sufficiently small. The constant δ depends on the topological similarity between the two graphs and cannot be controlled by the learning algorithm. Meanwhile, the constant λ_R in the above bound suggests that preventing λ_R from taking very large values should have a positive effect on the learning. This is in line with the choice of representing the label functions with a relatively small number R of basis vectors in Problem 3, in contrast to Problem 2. Then, another objective of Problem 3 is to minimize the difference between the coefficient vectors $\overline{\alpha}^s$ and $\overline{\alpha}^t$, which reduces Δ_α. Finally, the term $\|\overline{M} \odot \overline{T}\|_F^2$ in the learning objective aiming to discourage large off-diagonal entries will eventually help reduce the constant Δ_T in the above bound. Note, however, that we deliberately avoid imposing $\overline{T} \approx I$ in Problem 3, which would restrict the flexibility of the learnt bases in aligning the two graphs to account for the differences in the graph topologies. This is discussed in more detail in Section 4.3.3.

4.3 Proposed Method: Domain Adaptation via Spectral Graph Alignment

In this section, we present the proposed domain adaptation method, which we call Domain Adaptation via Spectral Graph Alignment (DASGA). Our algorithm aims to learn a pair of "aligned" bases on the source and target graphs based on Problem 3.

The problem in (4.5) is not jointly convex in all optimization variables $\overline{\alpha}^s$, $\overline{\alpha}^t$, $\overline{T}$. Nevertheless, it is convex separately in the overall coefficient vector $\overline{\alpha} = [(\overline{\alpha}^s)^T(\overline{\alpha}^t)^T]^T$, and the transformation matrix $\overline{T}$. Hence, we propose to minimize the objective (4.5) with an iterative and alternating optimization approach, by first fixing $\overline{T}$ and optimizing $\overline{\alpha}^s$, $\overline{\alpha}^t$; and then fixing the coefficient vectors $\overline{\alpha}^s$, $\overline{\alpha}^t$ and optimizing $\overline{T}$ in each iteration. We describe these two optimization steps in the sequel.

4.3.1 Optimization of the Coefficient Vectors

In the first step of an iteration, the transformation matrix $\overline{T}$ is fixed, and the coefficient vectors $\overline{\alpha}^s$ and $\overline{\alpha}^t$ are optimized. Fixing $\overline{T}$, the optimization problem in (4.5) becomes the following unconstrained problem in $\overline{\alpha}^s$ and $\overline{\alpha}^t$

$$\begin{aligned}\min_{\overline{\alpha}^s,\overline{\alpha}^t} G(\overline{\alpha}^s,\overline{\alpha}^t) = \min_{\overline{\alpha}^s,\overline{\alpha}^t} \quad & \|S^s\overline{U}^s\overline{\alpha}^s - y^s\|^2 \\ & + \|S^t\overline{U}^t\overline{T}\overline{\alpha}^t - y^t\|^2 + \mu_1\,\|\overline{\alpha}^s - \overline{\alpha}^t\|^2.\end{aligned} \tag{4.6}$$

The function $G(\overline{\alpha}^s,\overline{\alpha}^t)$ is convex in the coefficients $\overline{\alpha}^s$ and $\overline{\alpha}^t$ and its global minimum can be found by setting its derivatives to 0:

$$\begin{aligned}\frac{\partial G(\overline{\alpha}^s,\overline{\alpha}^t)}{\partial\overline{\alpha}^s} &= 2A^s\overline{\alpha}^s - 2B^sy^s + 2\mu_1\overline{\alpha}^s - 2\mu_1\overline{\alpha}^t = 0 \\ \frac{\partial G(\overline{\alpha}^s,\overline{\alpha}^t)}{\partial\overline{\alpha}^t} &= 2A^t\overline{\alpha}^t - 2B^ty^t + 2\mu_1\overline{\alpha}^t - 2\mu_1\overline{\alpha}^s = 0\end{aligned}$$

where

$$\begin{aligned}A^s &= (\overline{U}^s)^T(S^s)^TS^s\overline{U}^s, \qquad B^s = (\overline{U}^s)^T(S^s)^T \\ A^t &= (\overline{U}^t\overline{T})^T(S^t)^TS^t\overline{U}^t\overline{T}, \qquad B^t = (\overline{U}^t\overline{T})^T(S^t)^T.\end{aligned}$$

This gives the coefficient vectors as

$$\begin{aligned}&\overline{\alpha}^s = (\mu_1^{-1}A^tA^s + A^t + A^s)^{-1}(\mu_1^{-1}A^tB^sy^s + B^sy^s + B^ty^t) \\ &\overline{\alpha}^t = (\mu_1^{-1}A^s\overline{\alpha}^s + \overline{\alpha}^s - \mu_1^{-1}B^sy^s).\end{aligned}$$

4.3.2 Optimization of the Transformation Matrix

In the second step of an iteration, the coefficient vectors $\overline{\alpha}^s$ and $\overline{\alpha}^t$ are fixed and the transformation matrix $\overline{T}$ is optimized. Then the minimization of the objective in (4.5) becomes equivalent to the following problem

$$\begin{aligned} \min_{\overline{T}} H(\overline{T}) = &\min_{\overline{T}} \quad \|S^t \overline{U}^t \overline{T} \overline{\alpha}^t - y^t\|^2 + \mu_2 \, \|\overline{M} \odot \overline{T}\|_F^2 \\ &\text{subject to } \sum_{i=1}^{R} \overline{T}_{ij}^2 = 1, \text{ for } j = 1, \ldots, R. \end{aligned} \tag{4.7}$$

The above problem involves the minimization of a quadratic convex function $H(\overline{T})$ in $\overline{T}$ subject to R equality constraints that are also quadratic and convex in $\overline{T}$. We solve the problem in (4.7) using the Sequential Quadratic Programming (SQP) algorithm [69], which is a method to numerically solve constrained nonlinear optimization problems. The SQP algorithm is based on iteratively approximating the original problem with a Quadratic Programming problem, where the objective function is replaced with its local quadratic approximation, and the equality and inequality constraints are replaced with their local affine approximations. In our problem (4.7), the objective function $H(\overline{T})$ is already a quadratic function of $\overline{T}$ and we only have equality constraints.

The first and second order derivatives to be used in the solution of (4.7) are found as follows. Let $\bar{t} \in \mathbb{R}^{R^2}$ denote the column-wise vectorized form of the matrix $\overline{T}$, such that its k-th entry is given by $\bar{t}_k = T_{ij}$, with $k = (j-1)R + i$, for $i, j = 1, \ldots, R$. We denote by $h(\bar{t}) = H(\overline{T})$ the objective in (4.7) when considered as a function of $\bar{t}$. The objective function $h(\bar{t}) = H(\overline{T})$ can then be rewritten in terms of $\bar{t}$ as

$$h(\bar{t}) = \|A\bar{t} - y^t\|^2 + \mu_2 \|F\bar{t}\|^2. \tag{4.8}$$

Here $A \in \mathbb{R}^{L_t \times R^2}$ is a matrix with entries given by $A_{lk} = (S^t \overline{U}^t)_{li} \, \overline{\alpha}_j^t$ and $F \in \mathbb{R}^{R^2 \times R^2}$ is a diagonal matrix with entries given by $F_{kk} = \overline{M}_{ij}$, where $l = 1, \ldots, L_t$ and $k = R(j-1) + i$, for $i, j = 1, \ldots, R$. The variable L_t here is the number of labeled target samples. Next, the j-th equality constraint of the problem (4.7) can be written in terms of $\bar{t}$ as

$$g_j(\bar{t}) = \sum_{i=1}^{R} \overline{T}_{ij}^2 - 1 = 0 \tag{4.9}$$

for $j = 1, \ldots, R$.

The problem (4.7) is then solved by forming the Lagrangian function

$$(\bar{t}, \eta) = h(\bar{t}) - g(\bar{t}, \eta)$$

where

$$g(\bar{t}, \eta) = \sum_{j=1}^{R} \eta_j g_j(\bar{t}), \tag{4.10}$$

$\eta_j > 0$ are the Lagrange multipliers, and $\eta = [\eta_1 \ldots \eta_R]^T$. From (4.8), we obtain the gradient of the objective $h(\bar{t})$ as

$$\nabla_{\bar{t}} h = 2(A^T A + \mu_2 F^T F)\bar{t} \tag{4.11}$$

and its Hessian as

$$\nabla^2_{\overline{tt}} h(\bar{t}) = 2(A^T A + \mu_2 F^T F). \tag{4.12}$$

Next, from (4.9), the k-th entry of the gradient of $g_j(\bar{t})$ is found as

$$\left(\nabla_{\bar{t}}\, g_j\right)_k = \begin{cases} 2\bar{t}_k\,, & \text{if } (j-1)R + 1 \leq k \leq jR \\ 0, & \text{otherwise} \end{cases} \tag{4.13}$$

for $k = 1, \ldots, R^2$. From (4.13), the Hessian $\nabla^2_{\overline{tt}}\, g(\bar{t}, \eta)$ of the second term $g(\bar{t}, \eta)$ of the Lagrangian in (4.10) is obtained as a diagonal matrix with entries given by

$$[\nabla^2_{\overline{tt}}\, g(\bar{t}, \eta)]_{kk} = 2\eta_j \tag{4.14}$$

for $R(j-1) + 1 \leq k \leq Rj$. Putting (4.12) and (4.14) together, we get the Hessian of the Lagrangian as

$$\nabla^2_{\overline{tt}}(\bar{t}, \eta) = \nabla^2_{\overline{tt}} h(\bar{t}) - \nabla^2_{\overline{tt}} g(\bar{t}, \eta).$$

The SQP algorithm optimizes objectives with equality constraints by iteratively updating the solution $(\bar{t}, \eta)$, where a linear system representing the approximate solution of the KKT conditions with the Newton's method is solved in each iteration [69, Algorithm 18.1]. The linear system is constructed from the objective $h(\bar{t})$, the constraints $g_j(\bar{t})$, their gradients, and the Hessian of the Lagrangian.

4.3.3 Overall Optimization Procedure

We now overview the overall optimization procedure employed in the proposed DASGA method. First, the optimization variables $\overline{T}$, $\overline{\alpha}^s$, and $\overline{\alpha}^t$ are initialized as follows. Since the objective in Problem 3 aims to find a transformation that aligns the source and target Fourier bases, a natural choice would be to initialize $\overline{T}$ as the identity matrix, so that each source vector u_i^s is mapped to the target vector u_i^t. However, even in a simple scenario where the source and target graphs are very similar, as the eigenvalue decomposition determines eigenvectors up to a sign, mapping each u_i^s to u_i^t might in fact constitute a bad initialization; e.g., consider the very simple case where the source and target graphs are identical but $u_i^t = -u_i^s$. An unfavorable initialization of the transformation matrix may consequently influence the estimates of the coefficient vectors $\overline{\alpha}^s$, $\overline{\alpha}^t$ and affect the overall solution of the alternating optimization procedure.

In order to obtain a more favorable initialization, we propose to set the initial $\overline{T}$ matrix with a strategy that corrects the sign of each target vector according to its best match among the source basis vectors. This strategy is based on the method presented in our work [70], where the best match of a target vector u_i^t among the source vectors is determined by finding

$$\max_j |\langle \tilde{u}_j^s, \tilde{u}_i^t \rangle|. \tag{4.15}$$

Here $\tilde{u}_j^s$, $\tilde{u}_i^t$ are subvectors of the basis vectors u_j^s, u_i^t obtained by restricting them to a subset of their entries indexed by some $\{s_i\}_{i=1}^K$ and $\{t_i\}_{i=1}^K$. It is difficult to directly compare the vectors u_j^s, u_i^t as the nodes of the source and target graphs are ordered arbitrarily and independently of each other. If a set of corresponding source and target node pairs $\mathcal{N} = \{(x_{s_i}^s, x_{t_i}^t)\}_{i=1}^K$ is known, then this set can be used for the restriction of the basis vectors to a subset of their entries in the problem (4.15), so that the vectors u_j^s, u_i^t can be compared throughout their chosen entries. However, in our method we do not rely on the availability of a set of corresponding node pairs and propose to form the set $\mathcal{N} = \{(x_{s_i}^s, x_{t_i}^t)\}_{i=1}^K$ based on the class labels, such that each pair of matched nodes $(x_{s_i}^s, x_{t_i}^t)$ is formed randomly among the source and target nodes having the same class labels. We then compare the vectors u_j^s, u_i^t over their entries $\tilde{u}_j^s$, $\tilde{u}_i^t$ corresponding to these nodes. Although very few labeled target nodes are typically available in a domain adaptation application, we have observed that only a few pairs is often sufficient to determine the correct signs for initializating $\overline{T}$, which is next done as follows

$$\overline{T}_{ii} = \mathrm{sgn}(\langle \tilde{u}_{J_i}^s, \tilde{u}_i^t \rangle), \qquad J_i = \arg\max_j |\langle \tilde{u}_j^s, \tilde{u}_i^t \rangle|. \tag{4.16}$$

Here sgn denotes the sign function and $\overline{T}$ is initialized as a diagonal matrix with -1's or 1's on the diagonals that matches the sign of each target vector u_i^t to the source vector u_j^s best corresponding to it. Note that this initialization respects the normalization constraint on the entries of the $\overline{T}$ matrix in (4.5).

Once the transformation matrix $\overline{T}$ is initialized in this way, the alternating optimization procedure starts, where the coefficient vectors $\overline{\alpha}^s$ and $\overline{\alpha}^t$ are computed by fixing $\overline{T}$ first, and then $\overline{T}$ is optimized by fixing $\overline{\alpha}^s$ and $\overline{\alpha}^t$ in each iteration, as described in Sections 4.3.1 and 4.3.2. In each iteration, both the updates on $\overline{\alpha}^s$ and $\overline{\alpha}^t$, and the update on $\overline{T}$ either reduce or retain the value of the objective function in (4.5). Since the objective function is nonnegative and thus bounded from below, it converges throughout the proposed iterative alternating optimization process. We continue the iterations until the convergence of the objective function. The proposed Domain Adaptation via Spectral Graph Alignment (DASGA) algorithm is summarized in Algorithm 1.

4.3.4 Complexity Analysis

We now present the complexity analysis of the proposed method. The overall complexity is mainly determined by the complexity of Steps 4 and 5 of Algorithm 1 executed iteratively until convergence. Let L_s and L_t denote the number of labeled samples respectively in the source and the target domains.

We first derive the complexity of Step 4. In the solution of (4.6), the matrices B^s and A^s are respectively computed with $O(L_s N_s R)$ and $O(L_s N_s R + L_s R^2)$ operations. Meanwhile, these are constant matrices that do not depend on $\overline{T}$ and they are computed only once; hence, we may ignore their calculation in the overall complexity. Next, $O(N_t R^2 + L_t N_t R)$ and $O(N_t R^2 + L_t N_t R + L_t R^2)$ operations are needed

Algorithm 1 Domain Adaptation via Spectral Graph Alignment (DASGA)

```
1: Input:
   $W^s$, $W^t$: Source and target graph weight matrices
   $y^s$, $y^t$: Available source and target labels
2: Initialization:
   Set the transformation matrix $\overline{T}$ as in (4.16).
3: repeat
4:     Update coefficients $\overline{\alpha}^s$, $\overline{\alpha}^t$ by solving (4.6).
5:     Update transformation matrix $\overline{T}$ by solving (4.7).
6: until the objective function (4.5) converges
7: Output:
   $f^t = \overline{U}^t\overline{T}\overline{\alpha}^t$: Estimated target label function
   $f^s = \overline{U}^s\overline{\alpha}^s$: Estimated source label function
```

to compute the matrices B^t and A^t respectively. The matrices $\mu_1^{-1}A^tA^s + A^t + A^s$ and $\mu_1^{-1}A^tB^sy^s + B^sy^s + B^ty^t$ in the expression of $\overline{\alpha}^s$ are computed respectively with $O(R^3)$ and $O(L_sR^2 + L_tR)$ operations. Considering also the matrix inversion in its expression, $\overline{\alpha}^s$ is computed with $O(R^3)$ operations. The target coefficients $\overline{\alpha}^t$ are then obtained from $\overline{\alpha}^s$ with $O(R^2)$ operations. From the complexities of all these computations, we get the overall complexity of Step 4 of Algorithm 1 as $O(R^3 + (L_s + N_t)R^2 + L_tN_tR)$.

Next, we examine the complexity of executing Step 5 with the SQP algorithm. The complexity of the evaluation of $h(\overline{t})$ in (4.8) is of $O(L_tR^2 + R^4)$. From (4.11), we observe that the gradient $\nabla_{\overline{t}}h$ is computed with $O(R^4)$ operations as well. Finally, since the Hessian $\nabla^2_{\overline{t}\overline{t}}h(\overline{t})$ of the objective in (4.12) is a constant matrix that does not depend on $\overline{t}$, we can exclude it from the complexity of the iterative SQP algorithm. Next, from (4.9), the complexity of computing all R gradients is obtained as $O(R^2)$. From (4.13) and (4.14), we observe that the gradients $\nabla_{\overline{t}}g_j(\overline{t})$ of the constraints and the Hessian $\nabla^2_{\overline{t}\overline{t}}\, g(\overline{t}, \eta)$ are obtained directly from $\overline{t}$ and η without any operations. We thus conclude that the Hessian $\nabla^2_{\overline{t}\overline{t}}(\overline{t}, \eta)$ of the Lagrangian can also be obtained with negligible complexity. Finally, the optimization variables are updated by solving the linear system given in [69, Algorithm 18.1] with $O(R^6)$ operations in a single iteration of the SQP algorithm. Putting together the complexities of all these operations, we conclude that the complexity of solving Step 5 with the SQP algorithm is of $O(R^6 + L_t R^2)$.

Finally, considering together the Steps 4 and 5 of Algorithm 1, we get the overall complexity of the DASGA algorithm as $O(R^6 + (L_s + N_t)R^2 + L_t N_t R)$.

CHAPTER 5

EXPERIMENTAL RESULTS

In the following, we first evaluate the performance of the proposed method with comparative experiments. We then study the behavior of the algorithm throughout the iterative optimization procedure and examine its sensitivity to the choice of the algorithm parameters.

5.1 Evaluation of the Algorithm Performance

The proposed algorithm is tested on several real and synthetic datasets. The performance of the proposed DASGA method is compared to the domain adaptation methods Heterogeneous Domain Adaptation using Manifold Alignment (DAMA) [62], Easy Adapt++ (EA++) [71], Subspace Alignment (SA) [42] and Geodesic Flow Kernel for Unsupervised Domain Adaptation (GFK) [37]; as well as the baseline classifiers Support Vector Machine (SVM), Nearest-Neighbor classification (NN), and the graph-based Semi-Supervised Learning with Gaussian fields (SSL) algorithm [10]. The baseline classifiers are evaluated under the “source+target” setting, using the labeled samples from both the source and the target domains as the training set, which has been observed to give better results than the “source only” and “target only” settings in general due to the limited number of target labels. When using the SA and GFK algorithms, once the source and target domains are aligned in an unsupervised way as proposed in [42] and [37], the known source and target labels are both used in the final classification of test samples. In the application of DASGA algorithm, the weight matrices W^s, W^t are constructed with a Gaussian kernel using the local scaling strategy introduced in [72]. It is proposed to calculate a local scaling parameter instead of using a single scaling parameter for the entire graph in this local scaling strategy. In each of the following experiments, the source labels are assumed to be known and the ratio of known target labels are varied gradually. The class labels of the unlabeled target samples are then estimated with the tested algorithms and the classification performances are compared.

5.1.1 Experiments on synthetic data sets

The first set of experiments are conducted on synthetic data sets with two classes. In the source domain, 100 samples are drawn for each class from a normal distribution in $\mathbb{R}^3$, with different means for the two classes. The target domain samples are then obtained by rotating the source domain samples by $90°$ around the x-axis. The three data sets shown in Figures 5.1, 5.2 and 5.3 are generated by varying the variances of the normal distributions, where the variance gradually increases from synthetic dataset-1 to synthetic dataset-3. The difficulty of classification increases with the variance of the distribution.

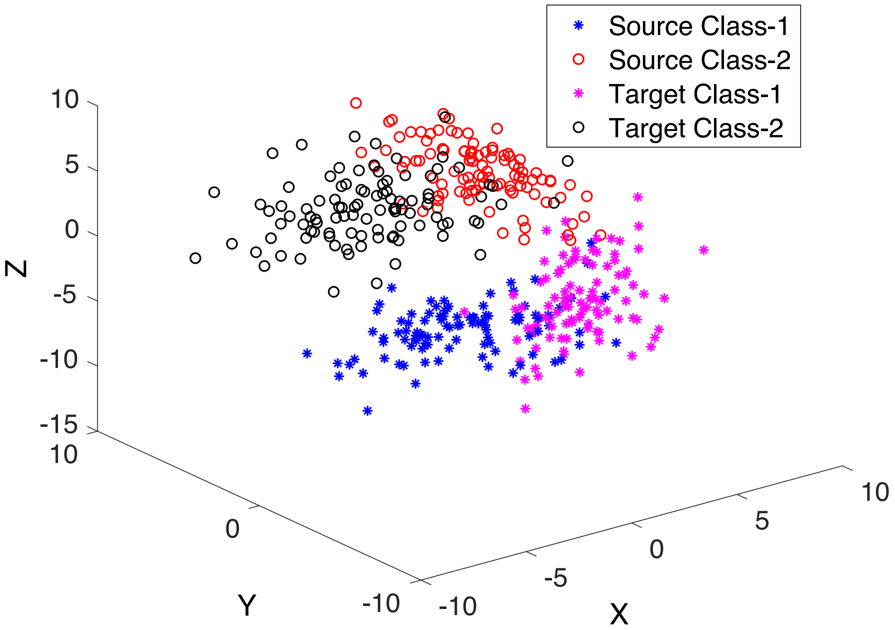

Figure 5.1: Synthetic dataset-1

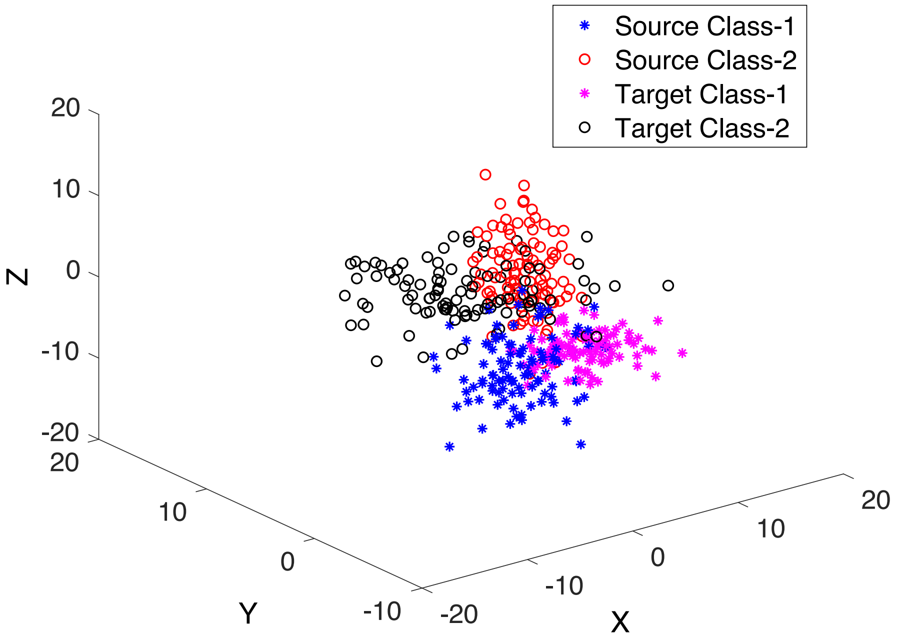

Figure 5.2: Synthetic dataset-2

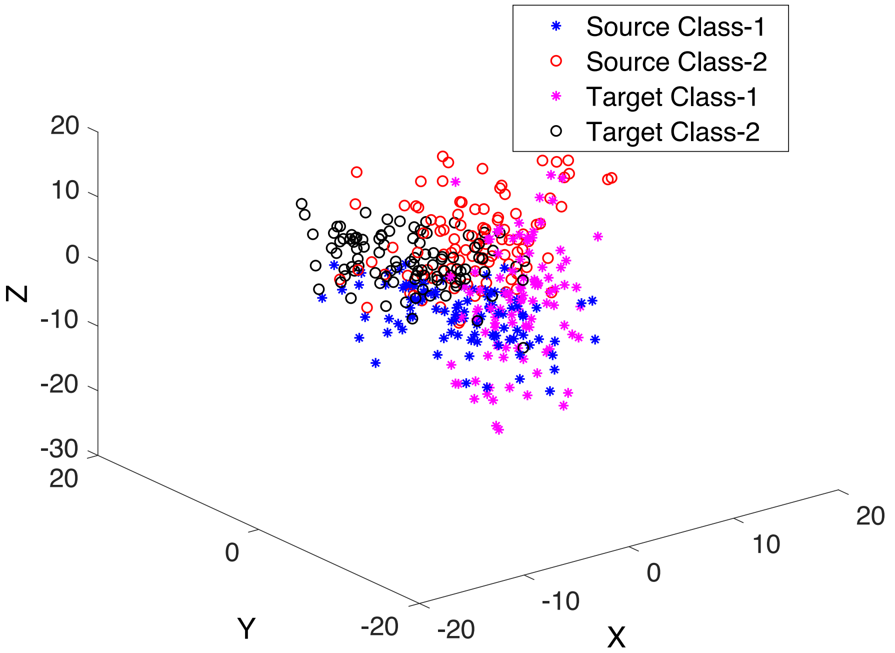

Figure 5.3: Synthetic dataset-3

As can be seen in Figure 5.1, the variances of data are low enough so that a straight plane can separate the data classes. As demonstrated in Figures 5.2 and 5.3, these two datasets are more challenging.

The proposed DASGA algorithm is used with the parameters $\mu_1 = 0.001$, $\mu_2 = 1$, $R = 10$. In the graphs constructed for DASGA algorithm, each vertex corresponds to a different data sample, i.e., a different feature vector. The source and target graphs are constructed by connecting each data sample to their 20 nearest neighbors, and a Gaussian kernel is used for forming the weight matrices.

In Figures 5.4, 5.5 and 5.6 the misclassification rates of unlabeled target samples in percentage are plotted with respect to the ratio of labeled target samples in percentage for the three synthetic data sets. The results are averaged over 100 repetitions of the experiment with random selections of the labeled samples. As expected, the misclassification rates of the algorithms have the general tendency to decrease as the ratio of known target labels increases.

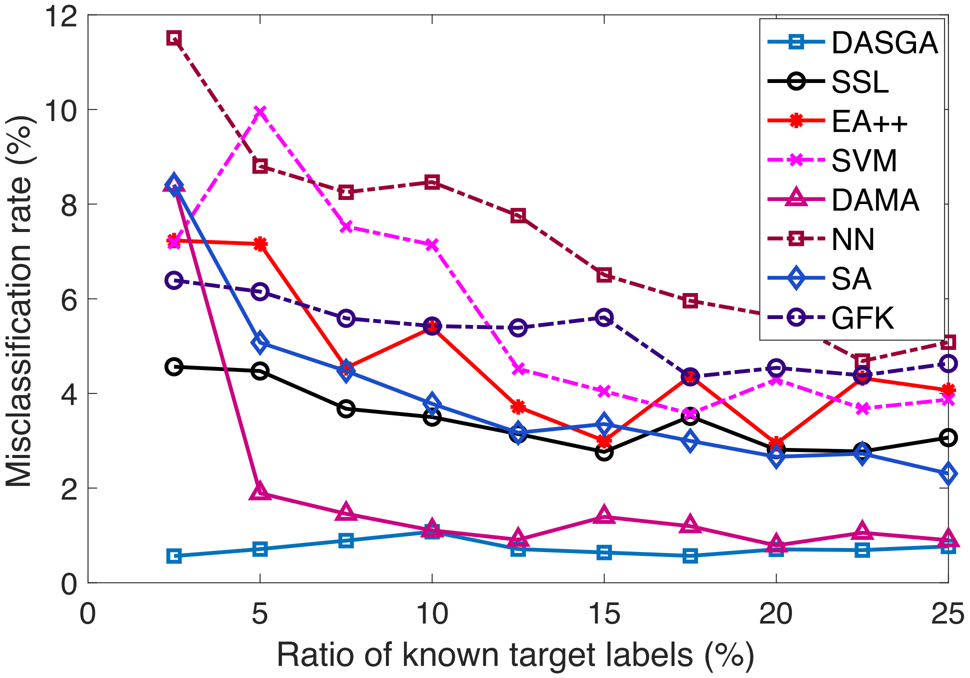

Figure 5.4: Synthetic Dataset-1 Errors

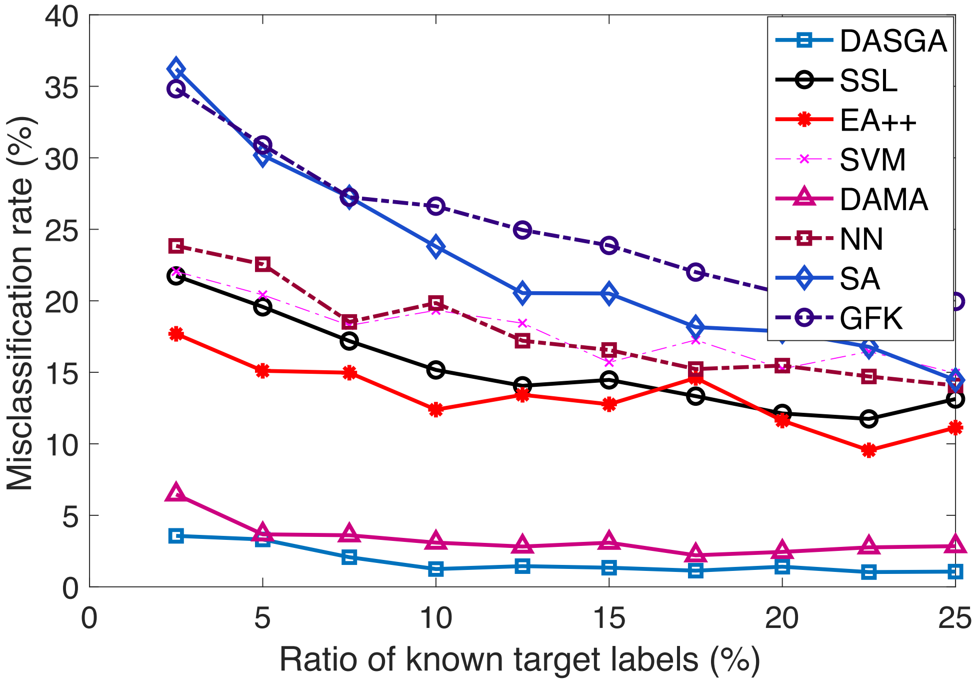

Figure 5.5: Synthetic Dataset-2 Errors

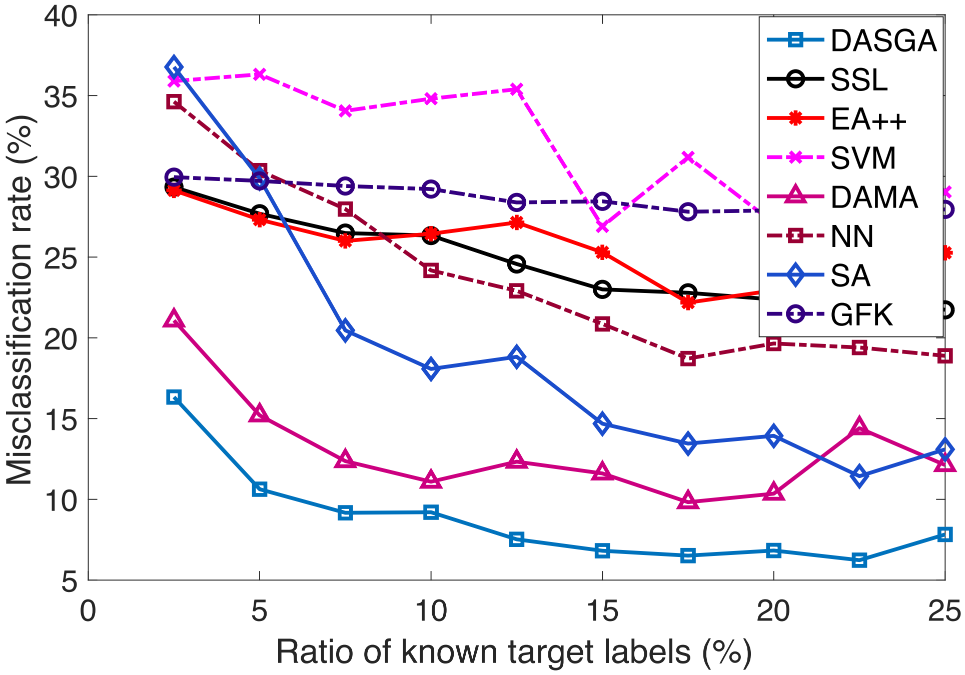

Figure 5.6: Synthetic Dataset-3 Errors

The proposed DASGA algorithm is observed to outperform the compared methods in all three experiments. As the variance of the distributions increases from Figure 5.4 to 5.6, the performance gap between DASGA and the other methods increases. The baseline SVM, NN, and SSL classifiers give a relatively small error (less than 10%) in the Synthetic dataset-1, where the two classes are better separated from each other due to the small variance of the distributions. However, the performances of these baseline classifiers degrade in the Synthetic datasets 2 and 3 where the data variance increases.

The domain adaptation methods tend to perform better than the baseline classifiers in general. In particular, the DAMA algorithm [62] follows the proposed DASGA algorithm in all experiments. DAMA is a supervised method aiming to preserve the topology of the data set via a graph model when learning a discriminative projection. This feature of DAMA seems to bring an advantage over the SA and GFK methods, which align the two domains in an unsupervised way. The proposed DASGA method is the least affected by the challenges in the data distributions such as large variance and poor separation between the classes. As DASGA is purely based on a graph representation of data, it detaches the ambient space properties of data from its representation to some extent. The misclassification rate of the graph-based DASGA method degrades in Figure 5.6 compared to Figure 5.4 by around 15%, whereas the degradation in the misclassification rates of the subspace-alignment-based SA and GFK methods, or the feature-augmentation-based EA++ method is around 25%.

5.1.2 Experiments on image data sets

We next evaluate the performance of the proposed algorithm on two image data sets. The first set of experiments are done on the MIT-CBCL face recognition database [73]. The data set consists of a total of 3240 face images rendered from the 3D head models of 10 subjects under varying illumination and poses. The images of each subject are rendered under 9 different poses varying from the frontal view (Pose 1) to a nearly profile view (Pose 9), and 36 illumination conditions at each pose. Some sample images are shown in Figure 5.7. We downsample the images to a resolution of 100×100 pixels. In our experiments, we consider the images taken under each

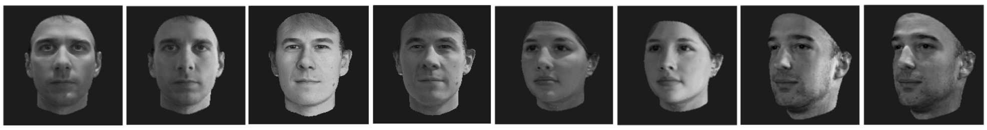

Figure 5.7: Sample images from the MIT-CBCL face data set for three different subjects [73]. Leftmost two, middle two, and rightmost two images are rendered respectively under poses 1, 2, 5, and 9 for various illumination conditions.

pose as samples from a different domain. That is, the experiments are conducted by selecting one pose as the source domain and another pose as the target domain. Hence, each domain consists of the images of all 10 subjects rendered under varying illumination conditions at a certain pose.

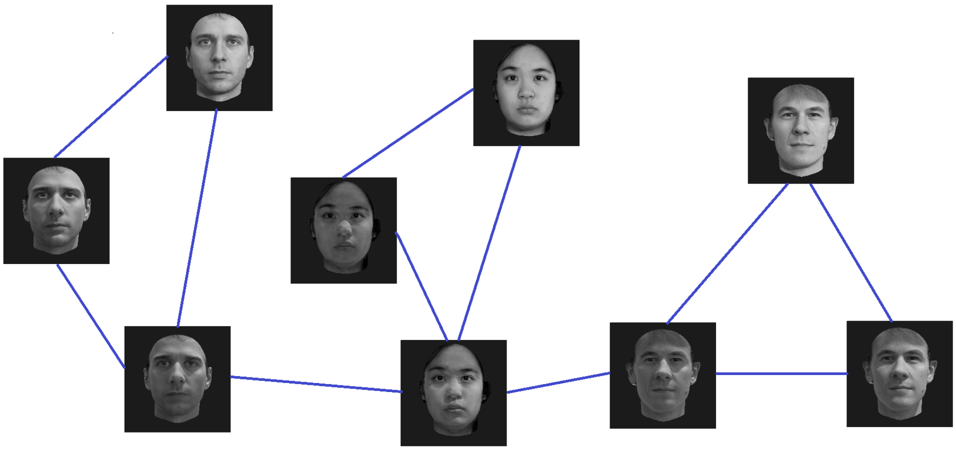

Figure 5.8: A face graph consisting of 9 images from 3 people

Three experiments are conducted by taking source domain as Pose 1. The target domain is taken as Pose 2 in the first experiment, Pose 5 in the second experiment and Pose 9 in the third experiment. In the construction of graphs for DASGA algorithm, the images are regarded as nodes of the graphs as illustrated in Figure 5.8. Source and target data graphs are constructed independently in the source and target domains, by connecting each image to its nearest 38 neighbors with respect to the Euclidean distance. The parameters of the proposed DASGA method are set as $\mu_1 = 0.01$, $\mu_2 = 0.85$, and $R = 9$, which are selected based on trials over the images from the other poses in the data set (6 and 8) that are not used in the experiments. The experiment is repeated over 50 realizations with random selections of the labeled

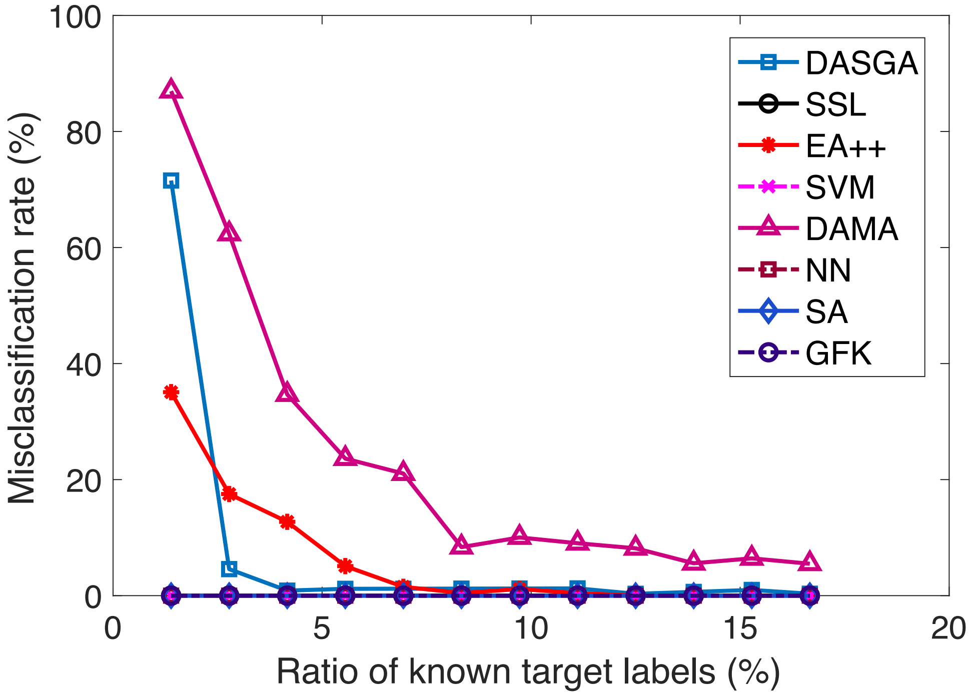

Figure 5.9: Misclassification rates obtained with the proposed SDA and reference methods. Source domain: Pose 1, Target domain: Pose 2

samples and the results are averaged.

The misclassification rates of the unlabeled target images are plotted with respect to the ratio of labeled target images in Figures 5.9, 5.10 and 5.11, where the target domain is respectively taken as Pose 2, Pose 5 and Pose 9. The misclassification errors of all algorithms are seen to be larger in Figure 5.11 compared to Figure 5.10, which is due to the fact that the similarity between the source and target domains is weaker in Figure 5.11 as the source and target poses differ more significantly. Misclassification rates in Figure 5.9 are generally observed to be lower compared to 5.10 due to the same reason. However, a noticeable difference in Figure 5.9 is that domain adaptation methods DASGA, DAMA and EA++ performs worse than baseline classifiers such as SVM and NN especially when the ratio of known target labels is lower than 5%. The reason for this difference is that Pose 1 and Pose 2 images are very similar, which enables SVM and NN to use pyhsical coordinates of images efficiently. In Figures 5.10 and 5.11, the misclassification rate of the proposed DASGA method is relatively high when the ratio of known target labels is below 5%, which quickly approaches 0, when at least 5% of the target samples are labeled. The only methods

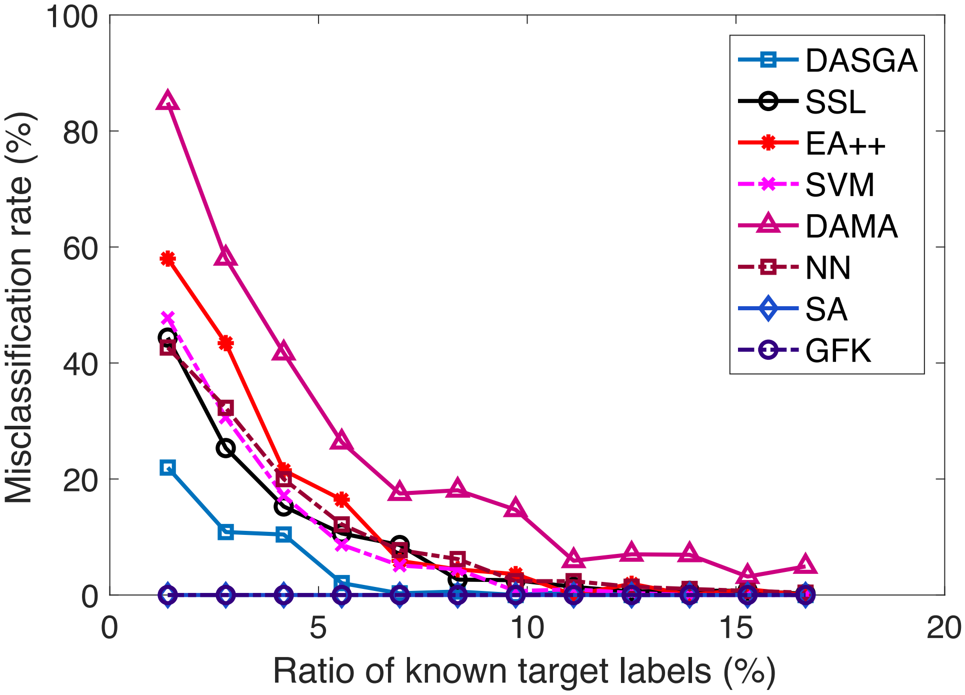

Figure 5.10: Misclassification rates obtained with the proposed SDA and reference methods. Source domain: Pose 1, Target domain: Pose 5

that outperform the proposed DASGA algorithm are the GFK and the SA domain adaptation methods. The performance of these two algorithms is particularly good in this experiment. The idea underlying these unsupervised methods is to align the low-dimensional subspaces approximating the source and target domains via geometric transformations. This approach is particularly appropriate for this face data set, as the PCA basis vectors of the face images of the same subjects captured from different poses can be easily aligned. The proposed graph-based DASGA algorithm does not use the pixel intensity values of image data samples once the source and target graphs are constructed, hence, it does not employ the same type of information as the GFK and SA methods. Nevertheless, its performance catches up with those of GFK and SA much quicker than the other methods in comparison as the number of known target labels increases.

The second image data set that is used in the evaluation of the proposed method is the COIL-20 object database [74]. The dataset consists of a total of 1440 images of 20 objects. Each object has 72 images taken from different viewpoints rotating around

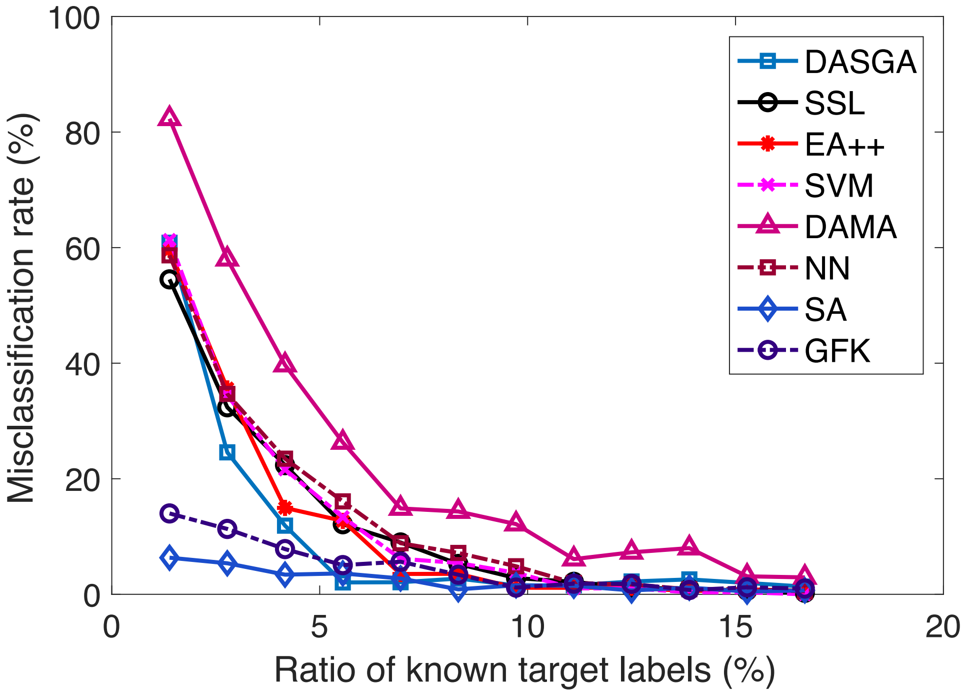

Figure 5.11: Misclassification rates obtained with the proposed SDA and reference methods. Source domain: Pose 1, Target domain: Pose 9

it. In this setup, images are initially masked with base mask images provided in the database such that the backgrounds of the images are set to black. This is a significant step because the distance between the images are needed to construct the graph and the background information can give misleading information. Original COIL-20 data set has images in 128×128 resolution but we downsample the images to a resolution of 32×32 pixels in our set-up. This reduction in the resolution is critical because algorithms such as SVM could consume an enormous amount of time when the data dimension is too high. Finally, the images are normalized in order to get rid of lighting effects. This is because the aim in this setup is to focus on the objects.

In order to build a transfer learning set-up with this database, a fictional scenario must be created because different domains do not exist in the original data set naturally. Therefore, we focus on a transfer learning scenario by dividing the 20 objects in the data set into two groups and matching each object in the first group to another object in the second group. The grouping is done by maximizing the similarity of the object pairs matched across the two groups based on the pairwise distances between the image samples of the objects. The distance of all objects in source domain to

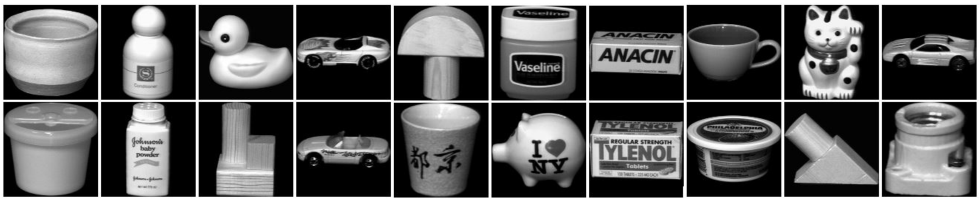

Figure 5.12: Sample images from the COIL-20 data set. The upper and lower rows show the objects respectively in the source domain and the target domain. Each source domain object is matched to the target domain object right below it. Matched object pairs are considered to have the same class label in the experiments.

the objects in the target domain are calculated in order to find the similarity between source and target objects. Let A and B be two classes and $a_1, a_2, \ldots, a_N \in A$ and $b_1, b_2, \ldots, b_N \in B$ be data samples drawn from these classes. The closest distance of all $a_i's$ to $b_i's$ are calculated and then summed to find the total distance between class A and class B. The most similar 2 objects are found and they are matched to each other. In this matching, the label of the source object is directly assigned as the label of target object, that is, the matched objects are regarded as belonging to the same class for the rest of the experiment. After obtaining a matched pair, the current pair is removed from the set of objects and the most similar objects from different domains are found in the remaining set by applying the same procedure. This process is repeated until all objects have a corresponding object in the other domain. At the end of this process, the matches given in Figure 5.12 are obtained.

The parameters of the proposed DASGA algorithm are set as $\mu_1 = 0.01$, $\mu_2 = 1$ and $R = 10$, in accordance with the typical values used in the previous experiments. In this experiment, images are regarded as nodes of the graphs which are constructed for DASGA algorithm as in the previous experiment. The source and target graphs are constructed by connecting each sample to its 3 nearest neighbors. This small value is chosen deliberately to be coherent with the small intrinsic dimension of the data set as the images are formed by rotating the camera around each object in only one direction. The class labels are represented with multidimensional one-hot label vectors.

The misclassification rates of the algorithms are plotted with respect to the ratio of known target labels in Figure 5.13. The proposed DASGA method is observed to

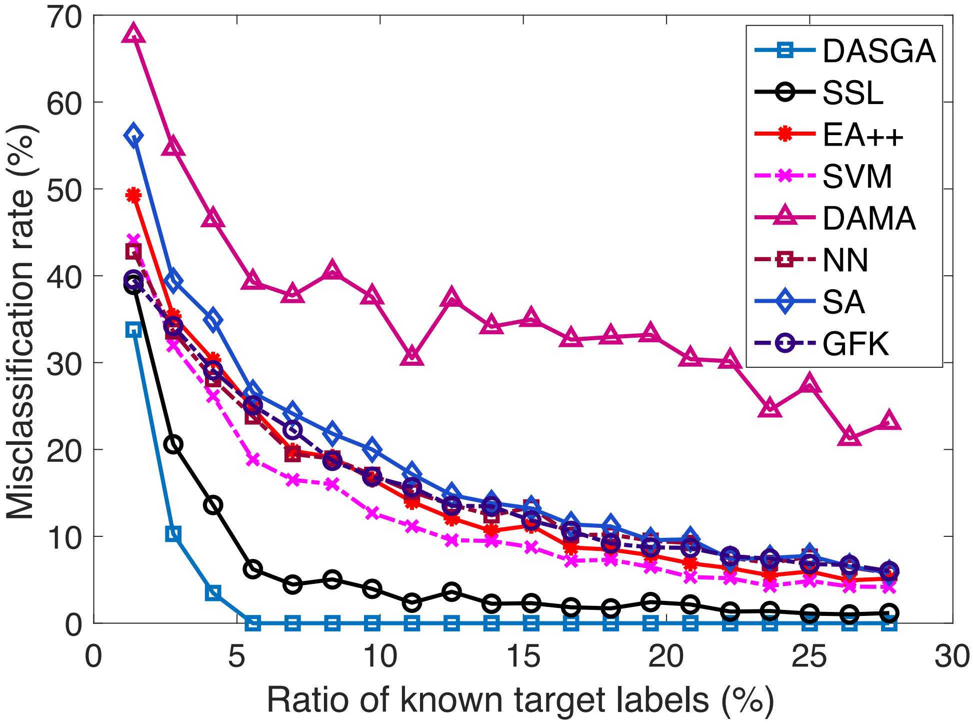

Figure 5.13: Misclassification rates of target samples for the COIL-20 object data

yield the best classification performance. The misclassification rate of the proposed algorithm reaches zero when about 5% of the samples are labeled in the target domain. The graph-based semi-supervised learning algorithm SSL follows the proposed method. The regular sampling of the images on the image manifold in this data set allows the construction of well-organized graphs, which can be successfully exploited by graph-based learning methods. The SVM algorithm also performs relatively well in this setup. Although the number of labeled samples in the target domain is limited, SVM can successfully make use of the labeled data samples in the source domain. Setting the object matches so as to minimize the pairwise distances causes the source and target domain samples from the same class to have relatively small distance, which contributes positively to the performance of SVM. The performances of the domain adaptation methods SA, GFK, and DAMA fall behind that of the baseline classifiers in this experiment. Relying on the alignment of the source and target domains via transformations or projections, these methods fail in the transfer learning problem considered in this experiment as the source and target images belong to different objects and hence they are difficult to align via projections or transformations.

5.1.3 Experiments on online book ratings data

The proposed algorithm is finally evaluated on the Amazon product ratings data set [75] for the prediction of the user ratings on books. The data set contains the scores from users who purchased a book from Amazon, where the scores are integers in the range $[1, 5]$. The experiment is conducted on the first 150000 ratings in the data set. The users who rated less than three books are excluded from the experiment.

In each repetition of the experiment, two bestsellers are chosen from the book catalogue of Amazon. The source graph consists of the users who read the first bestseller, and the target graph consists of the users who read the second bestseller. Each graph node corresponds to a user, and the scores of the users for the first and second bestsellers are regarded as signals (label functions), respectively on the source and the target graphs. The purpose of the experiment is to predict the user scores for the second bestseller on the target graph. The source and target graphs are constructed with respect to the similarities between the users, where two users are considered similar if their past reading records agree. Thus, if two users have read books in common, they are connected with an edge in the graphs. The edge weights are determined as inversely proportional to the average difference of the scores the users assigned to the same books, in order to capture the similarity of their literary preferences.

Given the scores on the source bestseller, and the available scores on the target bestseller, we estimate the unavailable scores on the target bestseller with the compared algorithms. The parameters of the proposed DASGA algorithm are set as $\mu_1 = 0.001$, $\mu_2 = 0.8$, $R = 10$, which are selected by trials on a test setup with two arbitrarily chosen bestsellers that are not used in the experiments. Being a purely graph-based method, the proposed DASGA algorithm requires only the source and the target user graphs and the available ratings. Meanwhile, the other algorithms in comparison require as input the coordinates of the data samples; thus, need an embedding of the data in an ambient space. Unlike the image data and synthetic data used in the previous experiments, the data samples do not have a physical embedding in this experiment. One could possibly regard the user ratings given to previously read books as feature vectors. However, due to the very large number of books in the Amazon catalogue and the small number of books users typically read, such feature vectors are very sparse in a very high-dimensional ambient space. This increases the complexity and impairs the performance and feasibility of most of the compared methods. In order to test the compared methods, we follow an alternative approach and embed the source and target graphs into an Euclidean domain of optimal dimension using the Multidimensional Scaling Algorithm (MDS) [76]. The coordinates learnt for each user with MDS are then used as training features by the compared algorithms.

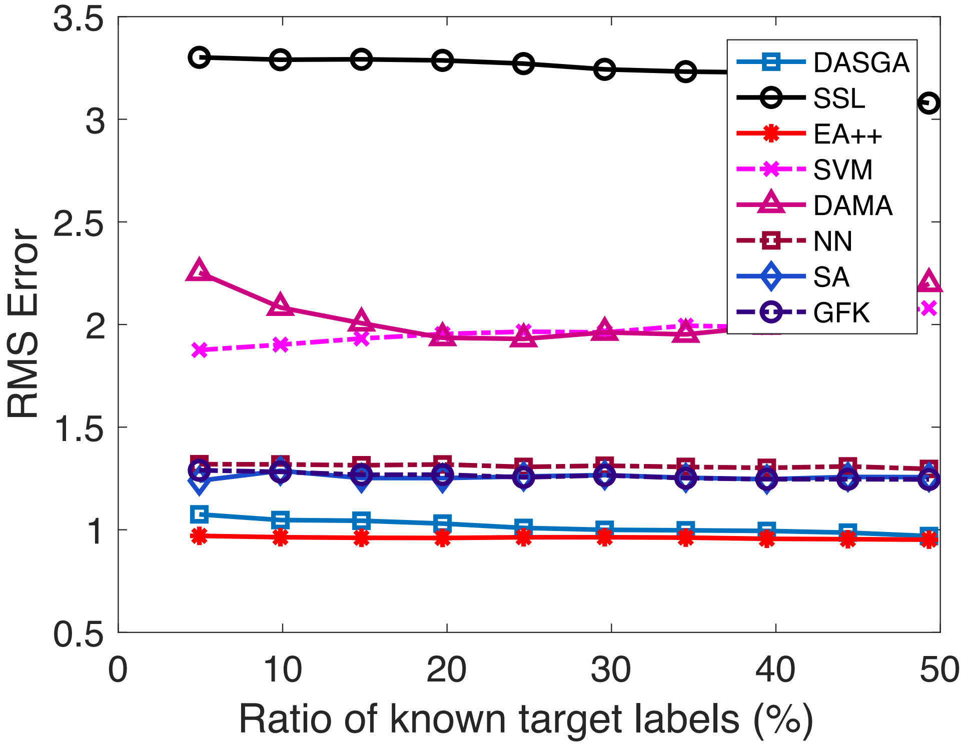

Figure 5.14: RMS errors of target user score predictions for Amazon book ratings

The experiment is conducted over 10 different pairs of source and target bestsellers, with 10 repetitions of the experiment for each bestseller pair by randomly selecting the labeled nodes. Figure 5.14 shows the root mean square (RMS) error of the predictions of user scores on the target bestseller, with respect to the ratio of available scores for the target bestseller. The misclassification rates of the score predictions (considering each score from 1 to 5 as a different class label) are also plotted in Figure 5.15. The errors are averaged over all experiments. The results in Figure 5.14 show that the proposed DASGA method provides the smallest RMS prediction er-

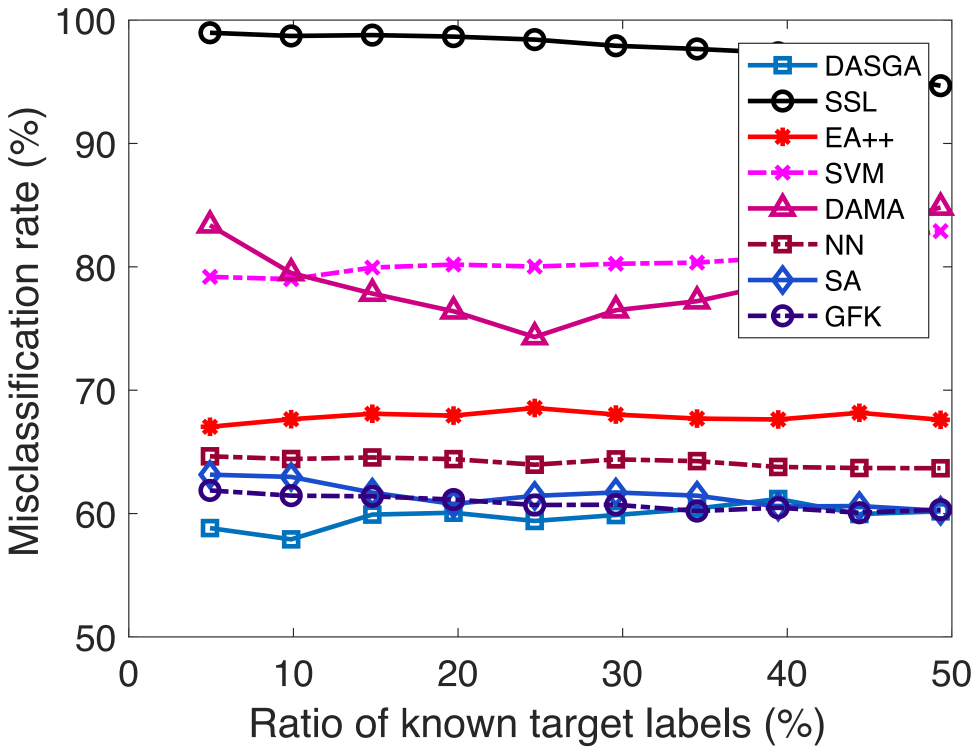

Figure 5.15: Misclassification rates of target user score predictions for Amazon book ratings

ror among the compared algorithms, except for the EA++ algorithm. On the other hand, the misclassification rates in Figure 5.15 show that the EA++ method gives a considerably higher misclassification rate than most domain adaptation methods in comparison. The reason for this discrepancy between the RMS error and the misclassification rate is that, in this data set, users tend to assign scores to books within a rather limited range, where most scores vary within 3 and 5. For this reason, although an algorithm does not predict the score labels correctly, its RMS prediction error may remain relatively small. Hence, the results in Figures 5.14 and 5.15 should be considered together when assessing the performances of the algorithms. The overall results suggest that the performance of the DASGA method is quite satisfactory compared to the other algorithms when both the RMS error and the misclassification rate are taken into account. The RMS prediction error of DASGA is seen to decrease at a very slow rate with the increase in the known target labels. This behavior is somewhat different from that observed in the previous experiments, and might possibly be explained with the properties of the data set. Due to the small number of ratings each user provides within a large book catalogue, there are relatively few pairs of users who read sufficiently many books in common. This causes the source and target graphs to have a sparse topology with a limited number of edges in this experiment; thus, the utility of the information of the known target labels in the prediction of the unavailable target labels is more limited compared to the denser graph topologies considered in the previous experiments on synthetic and image data. A significant amount of information regarding the spectral content of the label function is readily transferred from the source domain to the target domain via the proposed algorithm. This already allows the prediction of the target scores with a certain performance level even when there are few target labels, which does not improve significantly with the increasing availability of target labels.

5.2 Stabilization and Sensitivity Analysis of the Proposed Algorithm

We now study the behavior of the proposed DASGA algorithm throughout the iterative optimization procedure, as well as its sensitivity to the choice of the algorithm parameters.

We first examine the variations of the objective function and the misclassification rate of unlabeled target samples in percentage during the iterations. The value of the objective function (4.5) is evaluated in each iteration of the alternating optimization procedure, as well as the misclassification rate given by the solution computed in each iteration. The evolutions of the objective function and the misclassification rate are shown for the COIL-20 in Figures 5.16 and 5.17, and for the MIT-CBCL data sets in Figures 5.18 and 5.19. The results confirm that the objective function decreases monotonically throughout the iterations and converges as discussed in Section 4.3.3. The misclassification rate also has the general tendency to decrease during the iterations. The rate of decrease of the misclassification error follows closely that of the objective function in both data sets. This suggests that the objective function (4.5) underlying the proposed method captures well the actual performance of classification.

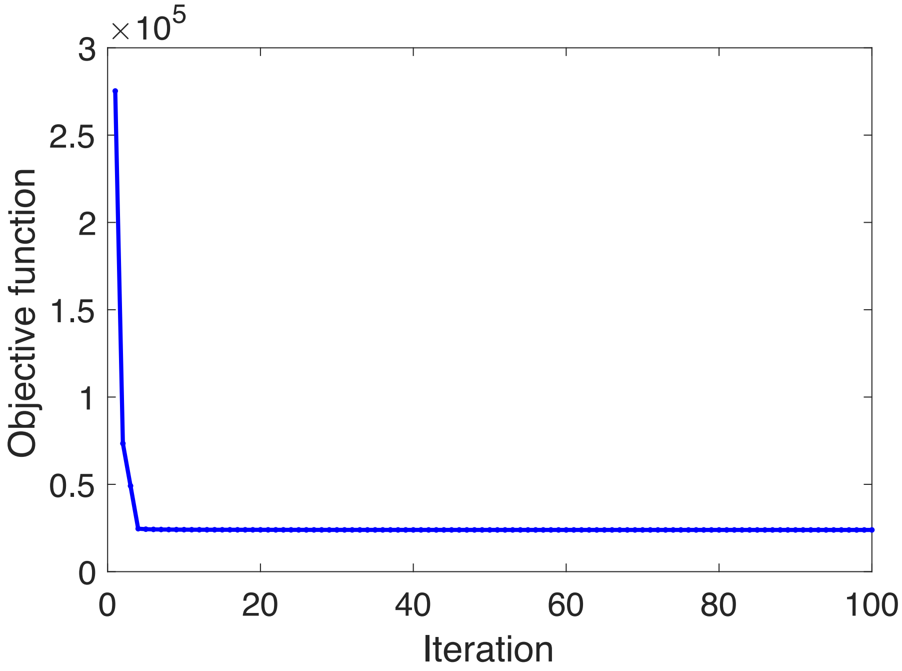

Figure 5.16: Evolution of the objective function throughout the iterations for the COIL-20 data

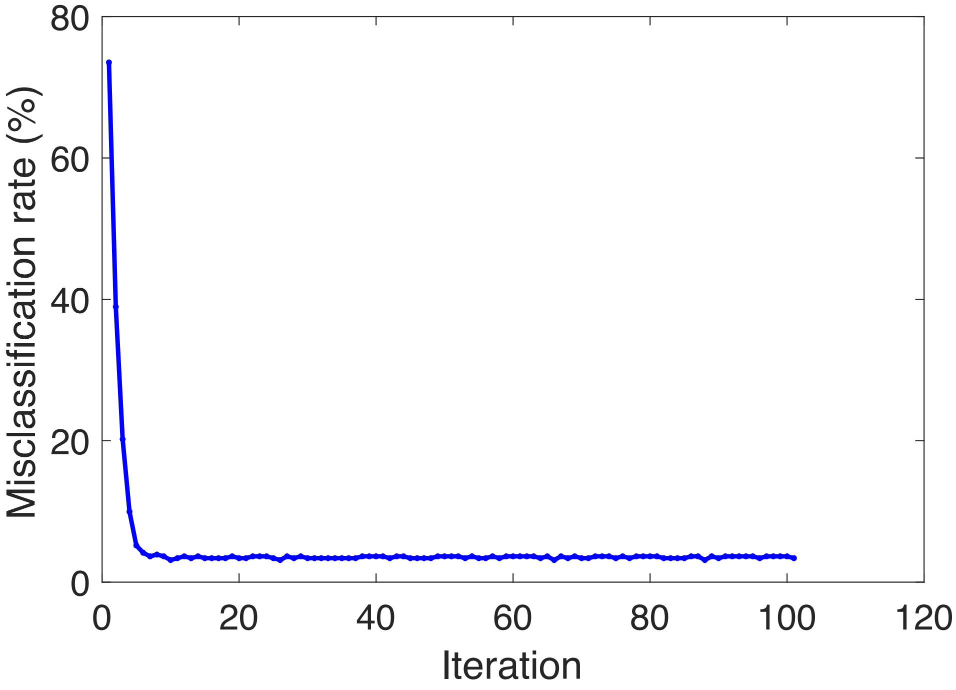

Figure 5.17: Evolution of the misclassification rate throughout the iterations for the COIL-20 data

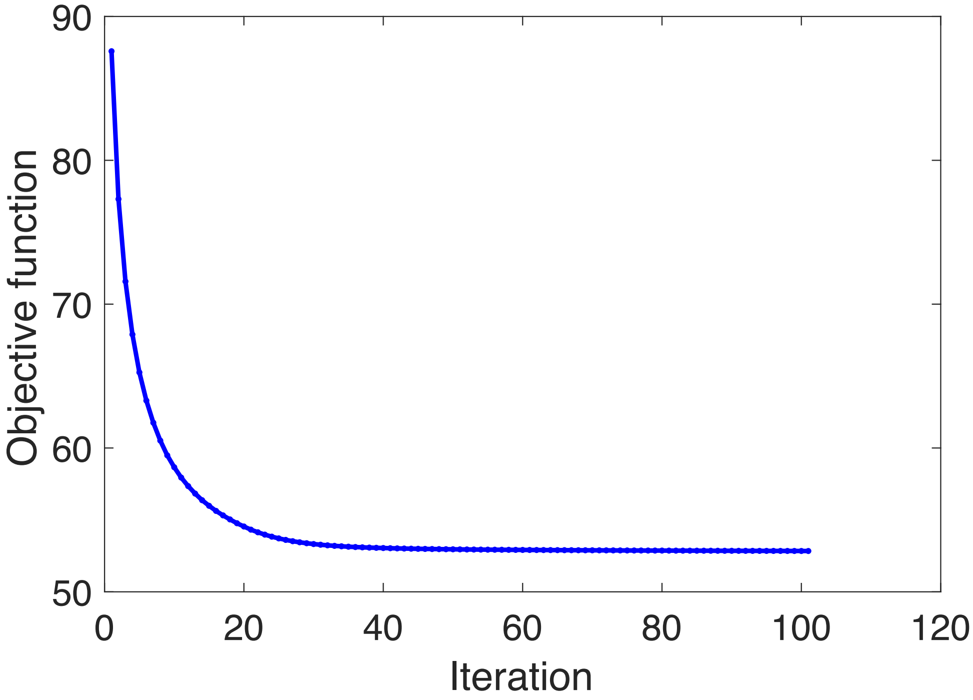

Figure 5.18: Evolution of the objective function throughout the iterations for the MIT-CBCL data

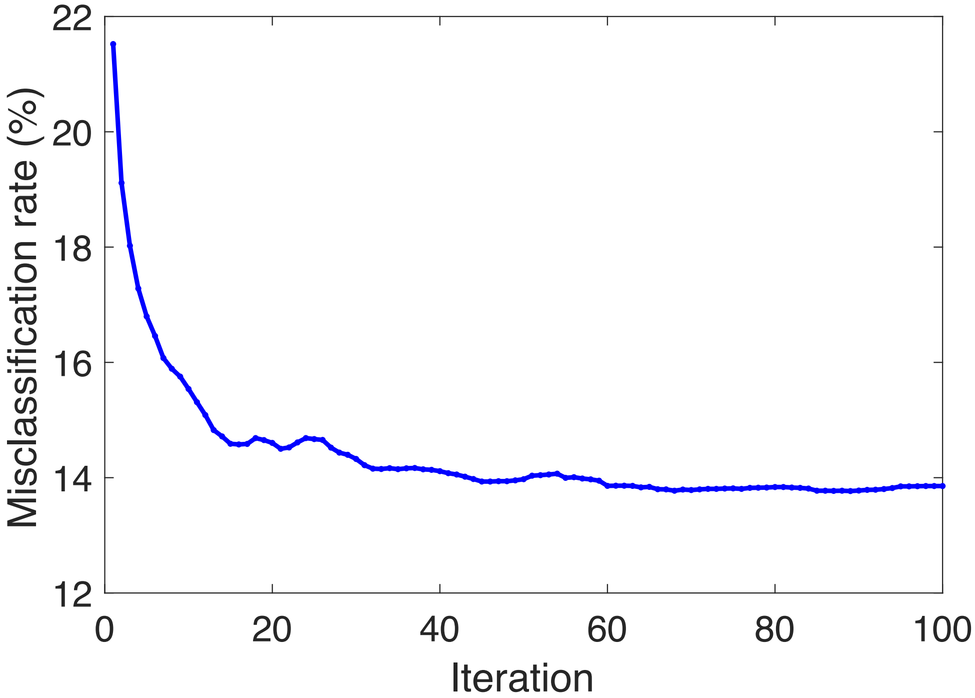

Figure 5.19: Evolution of the misclassification rate throughout the iterations for the MIT-CBCL data

Next, we study the sensitivity of the proposed method to the choice of the algorithm parameters. The experiments on the different types of data sets in Section 5.1 suggest that choosing the weight parameter μ_1 as around $0.001 - 0.01$ and μ_2 as around 1 yields reasonable performance in general. Here we focus on the other algorithm parameters that might also have an influence on the algorithm performance; namely, the number of nearest neighbors K used when constructing the source and target graphs, and the number of graph basis vectors R used in the objective (4.5). The variations of the misclassification rate of unlabeled target samples with the number of nearest neighbors K and the number of basis vectors R are shown respectively in Figures 5.20 and 5.21 for the synthetic data sets of Section 5.1.1, and in Figures 5.22 and 5.23 for the COIL-20 data set.

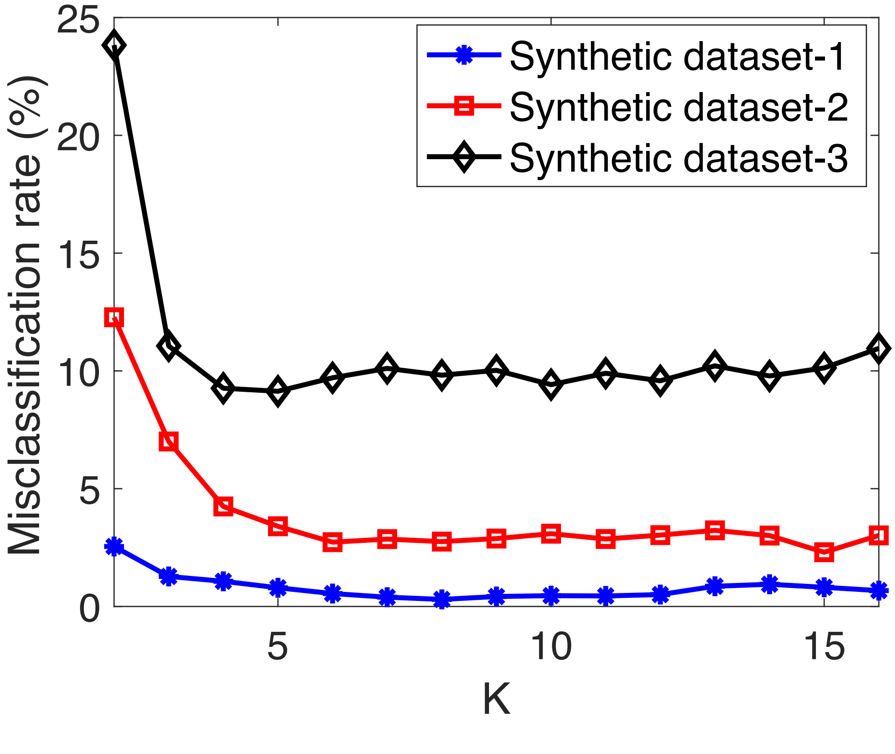

Figure 5.20: Variation of the misclassification rates of target samples with the number of neighbors K for the synthetic data set

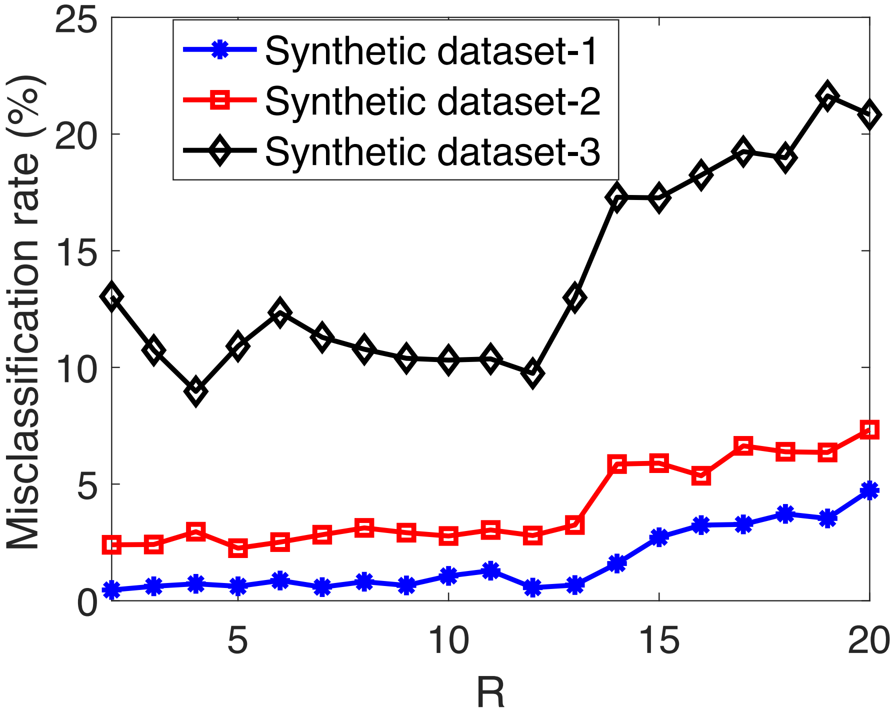

Figure 5.21: Variation of the misclassification rates of target samples with the number of basis vectors R for the synthetic data set

In Figure 5.20, the algorithm performance is seen to be stable over a relatively wide range of K values for the synthetic data set. It can be observed that the proposed method tends to favor smaller K values for the Synthetic dataset-3, compared to the other synthetic data sets. This may be explained with the fact that samples from the two classes are closer to each other in Synthetic dataset-3 due to the high variance of the normal distribution. This causes a larger portion of the nearest neigbors of a sample to be from the other class when K is high, which has a negative effect on the classification performance. Meanwhile, the results on the COIL-20 data set given in Figure 5.22 show that the proposed method is more sensitive to the choice of the K parameter in this data set. In particular, the optimal value of K is quite small and around $3-4$. In fact, this result is quite in line with the intrinsic geometric properties of this data set: As the images of the objects are taken by rotating the camera around the object by varying a single camera angle parameter, the intrinsic dimension of this data set is quite low. The best performance is then achieved when the graph is constructed with a small number of neighbors, which conforms to the geometric structure of data.

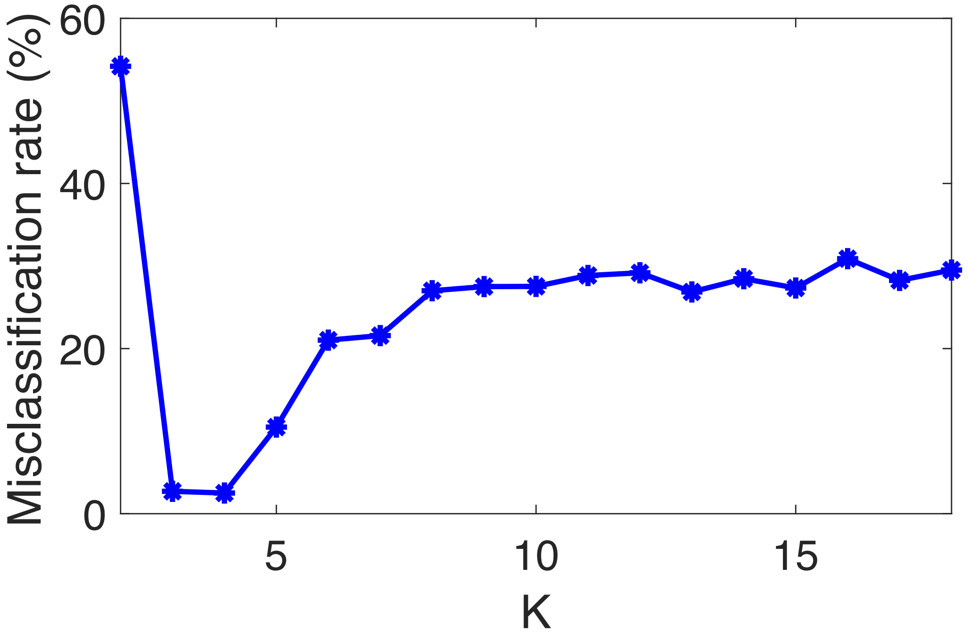

Figure 5.22: Variation of the misclassification rates of target samples with the number of neighbors K for the COIL-20 data set

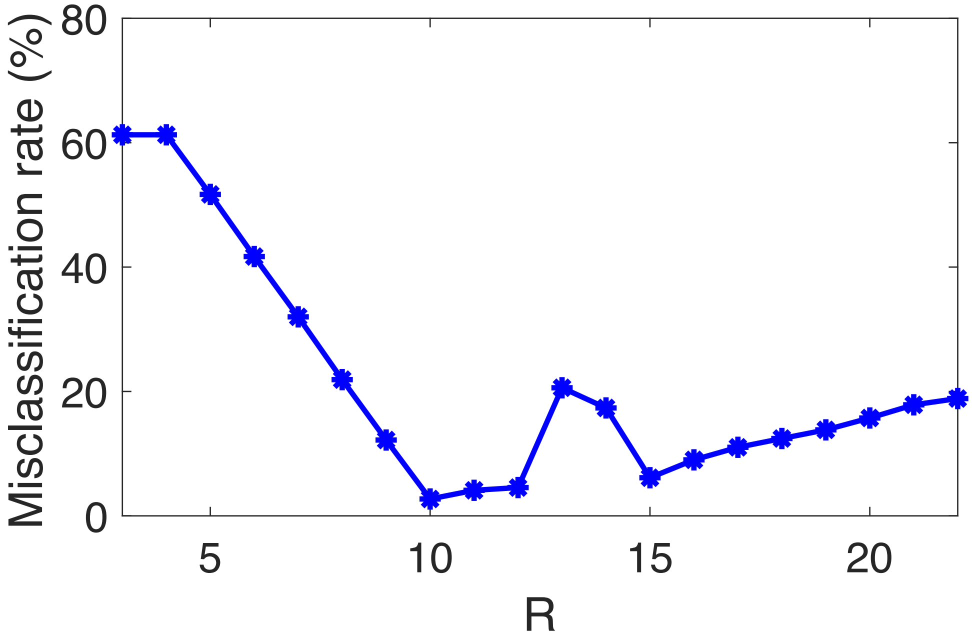

Figure 5.23: Variation of the misclassification rates of target samples with the number of basis vectors R for the COIL-20 data set

The results in Figure 5.21 suggest that the variation of the misclassification rate is quite stable over a relatively large range of R values for the Synthetic dataset-1 and the Synthetic dataset-2. The performance is observed to be higher when a smaller number of basis vectors is used. Since the variance of the normal distributions is rather small in these two synthetic data sets, the nearest neighbors of samples in the constructed source and target graphs tend to belong to the same class. Then, the label function has a slow variation on the graph, and consequently, a small number of low-frequency Fourier basis vectors approximate the label function well. The optimal value of R is seen to be higher for the Synthetic dataset-3, where the number of neighboring samples from different classes increases due to the high variance of the distributions. Then the label function has stronger high-frequency components compared to the first two synthetic data sets, so that a slightly higher R value is preferable. Comparing these results to those on the COIL-20 data given in Figure 5.23, one can see that the optimal number of basis vectors R is higher for the COIL-20 data set. The misclassification rate decreases as R increases for small R values, which is due to increase in the capability of representing the label function when more basis vectors are used. The optimal value of R is around 10-12, and the performance tends to degrade when R is increased beyond these values. This is because increasing R too much results in poor regularization and increases the misclassification error, which is also consistent with the theoretical bound in Proposition 1. This observation is also confirmed on the other real data sets by the results in Section 5.1, where it has been seen that setting R around 10 yields reasonable performance in general.

CHAPTER 6

CONCLUSION

Extracting useful information from available data is at the focus of popular research fields such as machine learning and data mining. In this thesis, we have addressed the problem of domain adaptation for learning with graph data, whose aim is to extract information from one domain so as to leverage it in another domain.

Domain adaptation and transfer learning terms are introduced and domain adaptation literature is reviewed in Chapter 2. Homogeneous domain adaptation, where source and target data representations are the same, and heterogeneous domain adaptation, where source and target data representations are different, are defined. Moreover, homogeneous and heterogeneous domain adaptation algorithms proposed in the literature are introduced.

Graph signal processing is reviewed in Chapter 3. Graph signal processing notions such as graph label function, graph Laplacian and graph Fourier transform are defined and spectral properties of graphs are introduced. This chapter is a critical part of this study because the thesis is based on spectral properties of graphs. As the Fourier transform in traditional signal processing gives information about how fast a signal varies with time, the Fourier transform in graphs gives information about how fast a graph function changes over the vertices of the graph. In this thesis, it is proposed that the spectral content of a graph signal acquired using the graph Fourier transform, can be utilized to obtain useful information about a similar graph signal on another graph. Moreover, it is shown that the eigenvectors of the graph Laplacians of two different graphs can be used as basis vectors and information from one graph can be transferred to the other one using these basis vectors.

In order to deal with the aforementioned domain adaptation problem, a novel algorithm, which is based on graphs, is proposed in this thesis. The motivation and derivation of our algorithm, DASGA, is presented in Chapter 4. Given a source graph with sufficiently many labeled nodes and a target graph, our graph-based domain adaptation algorithm estimates a label function on the target graph, relying on the assumption that the frequency content of the source and target label functions have similar characteristics. Our method is based on the idea of learning a pair of coherent bases on the source and target graphs not only resembling in terms of their spectral content, but also "aligning" the two graphs such that the label functions over the two graphs can be reconstructed with similar coefficients. The proposed domain adaptation algorithm is completely graph-based and is particularly applicable in learning problems defined purely on graph domains where no physical embedding of data samples is available. The proposed method can potentially be applied to many machine learning problems of interest concerning graph domains.

Four data sets consisting of synthetic data, COIL-20, MIT-CBCL face recognition database and Amazon book reviews are used to test the performance of our algorithm in Chapter 5. The performance of DASGA is compared to the performance of DAMA, SVM, Nearest Neighbor, EA++, SA and GFK algorithms. DASGA showed a notable performance in all experiments. DASGA performed best in the COIL-20 data set since the data consists of rotating images of objects, which makes graph-based methods advantageous. Moreover, algorithms using physical coordinates of the data are not successful in COIL-20 since matched objects in source and target domains are not the same object. This suggests that the proposed algorithm may have some potential in transfer learning applications where the relation between the tasks is weaker. DASGA also performs best in synthetic datasets and its performance gets superior to other algorithms as the data gets complicated. In MIT-CBCL dataset, DASGA becomes the third successful algorithm behind SA and GFK when the ratio of known target labels is under 5%. DASGA catches up their performance when the ratio of known target labels is higher than 5%. This is because GFK and SA are unsupervised algorithms and they are based on finding low dimensional subspaces approximating the source and target domains, which is appropriate for the face dataset. In Amazon book reviews dataset, DASGA performs best when RMS and misclassification errors are taken into account. Consequently, DASGA reveals its best performance when the graph models of source and target domains are related only although source and target samples are not similar in terms of the values of their features. That is, the performance of DASGA is superior to other methods when there cannot be found direct matches between source and target domain samples but there is a similarity between domains in terms of graph models as in the COIL-20 dataset. Besides, another advantage of DASGA algorithm is that it can be utilized for the datasets which do not have physical coordinates, e.g., Amazon book reviews dataset, since constructing consistent graphs in both domains are sufficient for DASGA without the need of physical coordinates of data.

Finally, the methodology proposed in this thesis uses Fourier basis in order to align source and target domains. The performance of DASGA may be improved by using various bases instead of the Fourier basis. Moreover, the construction of the graphs have a significant effect in the performance of the algorithm since the only information DASGA uses is the topology of the graph. In this thesis, Euclidean distance is used to determine the distances between data samples and Gaussian kernel weighting function is used to assign a similarity value for adjacent vertices in the graph with the K-NN approach. Different distance functions such as Manhattan distance and Canberra distance can be utilized to observe their effect on the performance of the algorithm. Another important parameter of the proposed algorithm is K in the K-NN method because it also has a significant effect in the topology of the resultant graph. Therefore, some effort might be spent in future studies in order to assign an optimal K value as an input to the algorithm. Besides, the performance of the algorithm can be improved by proposing new methods to determine the number of eigenvectors to be used, R, in the algorithm depending on the dataset. Lastly, the effect of having sparser graphs on the performance of DASGA may be investigated in the future works.

REFERENCES

[1] S. Ruder. Transfer learning - machine learning's next frontier. `http://ruder.io/transfer-learning/index.html#adaptingtonewdomains`. Accessed: 2018-01-01.

[2] Y. Liu. Graph-based learning models for information retrieval: A survey. 2006.

[3] M. Hein, J. Audibert, and U. von Luxburg. From graphs to manifolds - weak and strong pointwise consistency of graph laplacians. pages 470–485. Max-Planck-Gesellschaft, 2005.

[4] A. Singer. From graph to manifold laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128 – 134, 2006. Special Issue: Diffusion Maps and Wavelets.

[5] F. R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, December 1996.

[6] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.*, 30(3):83–98, 2013.

[7] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129 – 150, 2011.

[8] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003.

[9] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328, 2003.

[10] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. Twentieth Int. Conf. Machine Learning*, pages 912–919, 2003.

[11] T. Yao, Y. Pan, C. Ngo, H. Li, and T. Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, Boston, MA, USA, June 7-12, 2015*, pages 2142–2150, 2015.

[12] L. Cheng and S. J. Pan. Semi-supervised domain adaptation on manifolds. *IEEE Trans. Neural Netw. Learning Syst.*, 25(12):2240–2249, 2014.

[13] M. Xiao and Y. Guo. Semi-supervised subspace co-projection for multi-class heterogeneous domain adaptation. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II*, pages 525–540, 2015.

[14] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[15] K. R. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *J. Big Data*, 3:9, 2016.

[16] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

[17] G. Csurka. A comprehensive survey on domain adaptation for visual applications. In *Domain Adaptation in Computer Vision Applications.*, pages 1–35. 2017.

[18] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Proc. Advances in Neural Information Processing Systems 19*, pages 601–608, 2006.

[19] W. Dai, G. Xue, Q. Yang, and Y. Yu. Transferring naive Bayes classifiers for text classification. In *Proc. Twenty-Second AAAI Conference on Artificial Intelligence*, pages 540–545, 2007.

[20] J. Blitzer, R. T. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proc. Conf. Empirical Methods in Natural Language Processing*, pages 120–128, 2006.

[21] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye. A two-stage weighting framework for multi-source domain adaptation. In *Proc. Advances in Neural Information Processing Systems 24*, pages 505–513, 2011.

[22] W. Chu, F. De la Torre, and J. F. Cohn. Selective transfer machine for personalized facial action unit detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 3515–3522, 2013.

[23] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

[24] W. Dai, Q. Yang, G. Xue, and Y. Yu. Boosting for transfer learning. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, pages 193–200, 2007.

[25] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pages 200–209, 1999.

[26] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank. Domain transfer SVM for video concept detection. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1375–1381, 2009.

[27] L. Bruzzone and M. Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(5):770–787, 2010.

[28] Y. Chen, G. Wang, and S. Dong. Learning with progressive transductive support vector machine. *Pattern Recognition Letters*, 24(12):1845–1855, 2003.

[29] W. Jiang, E. Zavesky, S. Chang, and A. C. Loui. Cross-domain learning methods for high-level visual concept classification. In *Proceedings of the International Conference on Image Processing, ICIP, 2008, October 12-15, 2008, San Diego, California, USA*, pages 161–164, 2008.

[30] H. Cheng, P. Tan, and R. Jin. Localized support vector machine and its efficient algorithm. In *Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA*, pages 461–466, 2007.

[31] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th International Conference on Multimedia 2007, Augsburg, Germany, September 24-29, 2007*, pages 188–197, 2007.

[32] H. Daumé III. Frustratingly easy domain adaptation. *CoRR*, abs/0907.1815, 2009.

[33] H. Daumé III, A. Kumar, and A. Saha. Frustratingly easy semi-supervised domain adaptation. In *2010 Workshop on Domain Adaptation for Natural Language Processing, ACL 2010, Uppsala, Sweden, 15 July 2010.*, pages 53–59, 2010.

[34] H. Daumé III, A. Kumar, and A. Saha. Co-regularization based semi-supervised domain adaptation. In *Proc. Advances in Neural Information Processing Systems 23*, pages 478–486, 2010.

[35] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 999–1006, 2011.

[36] R. Gopalan, R. Li, and R. Chellappa. Unsupervised adaptation across domain shifts by generating intermediate data representations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(11):2288–2302, 2014.

[37] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2066–2073, 2012.

[38] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *Proc. Twenty-Third AAAI Conference on Artificial Intelligence*, pages 677–682, 2008.

[39] Z. Fang and Z. Zhang. Discriminative transfer learning on manifold. In *Proc. 13th SIAM Int. Conf. Data Mining*, pages 539–547, 2013.

[40] C. Wang and S. Mahadevan. Manifold alignment using procrustes analysis. In *Proc. 25th Int. Conf. Machine Learning*, pages 1120–1127, 2008.

[41] C. Wang. *A geometric framework for transfer learning using manifold alignment*. PhD thesis, 2010.

[42] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 2960–2967, Washington, DC, USA, 2013. IEEE Computer Society.

[43] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2058–2065, 2016.

[44] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 1187–1192, 2009.

[45] M. Baktashmotlagh, M. Tafazzoli Harandi, B. C. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 769–776, 2013.

[46] M. Baktashmotlagh, M. Tafazzoli Harandi, B. C. Lovell, and M. Salzmann. Domain adaptation on the statistical manifold. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 2481–2488, 2014.

[47] M. Long, G. Ding, J. Wang, J. Sun, Y. Guo, and P. S. Yu. Transfer sparse coding for robust image representation. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 407–414, 2013.

[48] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer joint matching for unsupervised domain adaptation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1410–1417, 2014.

[49] C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *CoRR*, abs/1304.5634, 2013.

[50] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes. On deep multi-view representation learning. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1083–1092, 2015.

[51] D. R. Hardoon, S. Szedmák, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[52] R. Socher and F. Li. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 966–973, 2010.

[53] F. Yan and K. Mikolajczyk. Deep correlation for matching images and text. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, Boston, MA, USA, June 7-12, 2015*, pages 3441–3450, 2015.

[54] A. Vinokourov, J. Shawe-Taylor, and N. Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, pages 1473–1480, 2002.

[55] M. Faruqui and C. Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 462–471, 2014.

[56] B. Tan, Y. Song, E. Zhong, and Q. Yang. Transitive transfer learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 1155–1164, 2015.

[57] Y. Zhu, Y. Chen, Z. Lu, S. J. Pan, G. Xue, Y. Yu, and Q. Yang. Heterogeneous transfer learning for image classification. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011.

[58] B. Tan, E. Zhong, M. K. Ng, and Q. Yang. Mixed-transfer: Transfer learning over mixed graphs. In *Proceedings of the 2014 SIAM International Conference on Data Mining, Philadelphia, Pennsylvania, USA, April 24-26, 2014*, pages 208–216, 2014.

[59] G. Qi, C. C. Aggarwal, and T. S. Huang. Towards semantic knowledge propagation from text corpus to web images. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, pages 297–306, 2011.

[60] L. Yang, L. Jing, J. Yu, and M. K. Ng. Learning transferred weights from cooccurrence data for heterogeneous transfer learning. *IEEE Trans. Neural Netw. Learning Syst.*, 27(11):2187–2200, 2016.

[61] Y. Yan, Q. Wu, M. Tan, and H. Min. Online heterogeneous transfer learning by weighted offline and online classifiers. In *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III*, pages 467–474, 2016.

[62] C. Wang and S. Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 1541–1546, 2011.

[63] L. Duan, D. Xu, and I. W. Tsang. Learning with augmented features for heterogeneous domain adaptation. *CoRR*, abs/1206.4660, 2012.

[64] W. Li, L. Duan, D. Xu, and I. W. Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(6):1134–1148, 2014.

[65] X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu. Transfer learning on heterogenous feature spaces via spectral transformation. *2010 IEEE International Conference on Data Mining*, pages 1049–1054, 2010.

[66] J. T. Zhou, I. W. Tsang, S. J. Pan, and M. Tan. Heterogeneous domain adaptation for multiple classes. In *Proceedings of the Seventeenth International Confer-*

ence on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014, pages 1095–1103, 2014.

[67] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 1785–1792, 2011.

[68] M. Harel and S. Mannor. Learning from multiple outlooks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 401–408, 2011.

[69] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

[70] M. Pilancı and E. Vural. Domain adaptation via transferring spectral properties of label functions on graphs. In *IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop*, pages 1–5, 2016.

[71] H. Daumé, III, A. Kumar, and A. Saha. Frustratingly easy semi-supervised domain adaptation. In *Proc. 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 53–59, 2010.

[72] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Proc. Advances in Neural Information Processing Systems 17*, pages 1601–1608. MIT Press, 2005.

[73] MIT-CBCL face recognition database. Available: http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html.

[74] S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-20). Technical report, Feb 1996.

[75] J. J. McAuley, R. Pandey, and J. Leskovec. Inferring networks of substitutable and complementary products. In *Proc. 21th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pages 785–794, 2015.

[76] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, Dec 1952.

APPENDIX A

PROOF OF PROPOSITION 1

Proof. The solution $\overline{\alpha}^s, \overline{\alpha}^t, \overline{T}$ of Problem 3 gives the estimated source and target label functions as $f^s = \overline{U}^s \overline{\alpha}^s$ and $f^t = \overline{U}^t \overline{T} \overline{\alpha}^t$. The rates of variation of f^s and f^t on the source and target graphs are given by

$$
\begin{aligned}
(f^s)^T L^s f^s &= (\overline{\alpha}^s)^T (\overline{U}^s)^T L^s \overline{U}^s \overline{\alpha}^s = (\overline{\alpha}^s)^T \Lambda^s \overline{\alpha}^s \\
(f^t)^T L^t f^t &= (\overline{T}\overline{\alpha}^t)^T (\overline{U}^t)^T L^t \overline{U}^t \overline{T} \overline{\alpha}^t = (\overline{T}\overline{\alpha}^t)^T \Lambda^t \overline{T} \overline{\alpha}^t
\end{aligned}
$$

where Λ^s and Λ^t are the diagonal matrices consisting of the R smallest eigenvalues of respectively L^s and L^t, such that $\Lambda^s_{ii} = \lambda^s_i$ and $\Lambda^t_{ii} = \lambda^t_i$, for $i = 1, \ldots, R$.

The difference between the rates of variations of f^s and f^t can then be bounded as

$$
\begin{aligned}
&|(f^s)^T L^s f^s - (f^t)^T L^t f^t| = |(\overline{\alpha}^s)^T \Lambda^s \overline{\alpha}^s - (\overline{T}\overline{\alpha}^t)^T \Lambda^t \overline{T}\overline{\alpha}^t| \\
&= |(\overline{\alpha}^s)^T \Lambda^s \overline{\alpha}^s - (\overline{\alpha}^s)^T \Lambda^t \overline{\alpha}^s + (\overline{\alpha}^s)^T \Lambda^t \overline{\alpha}^s - (\overline{\alpha}^t)^T \Lambda^t \overline{\alpha}^t + (\overline{\alpha}^t)^T \Lambda^t \overline{\alpha}^t - (\overline{T}\overline{\alpha}^t)^T \Lambda^t \overline{T}\overline{\alpha}^t| \\
&\leq |(\overline{\alpha}^s)^T (\Lambda^s - \Lambda^t) \overline{\alpha}^s| + |(\overline{\alpha}^s)^T \Lambda^t \overline{\alpha}^s - (\overline{\alpha}^t)^T \Lambda^t \overline{\alpha}^t| + |(\overline{\alpha}^t)^T \Lambda^t \overline{\alpha}^t - (\overline{T}\overline{\alpha}^t)^T \Lambda^t \overline{T}\overline{\alpha}^t|.
\end{aligned}
\tag{A.1}
$$

In the following, we derive an upper bound for each one of the three terms at the right hand side of the inequality in (A.1). The first term is bounded as

$$
|(\overline{\alpha}^s)^T (\Lambda^s - \Lambda^t) \overline{\alpha}^s| \leq \|\overline{\alpha}^s\|^2 \|\Lambda^s - \Lambda^t\| \leq C^2 \delta.
$$

Here the first inequality is due to the Cauchy-Schwarz inequality, and the second inequality follows from the fact that the operator norm of the matrix $\Lambda^s - \Lambda^t$ is given by the magnitude of its largest eigenvalue, which cannot exceed δ due to the assumption $|\lambda^s_i - \lambda^t_i| \leq \delta$ for all i.

Next, we bound the second term in (A.1) as

$$\begin{aligned}|(\overline{\alpha}^s)^T\Lambda^t\overline{\alpha}^s-(\overline{\alpha}^t)^T\Lambda^t\overline{\alpha}^t| &= |(\overline{\alpha}^s)^T\Lambda^t\overline{\alpha}^s-(\overline{\alpha}^s)^T\Lambda^t\overline{\alpha}^t+(\overline{\alpha}^s)^T\Lambda^t\overline{\alpha}^t-(\overline{\alpha}^t)^T\Lambda^t\overline{\alpha}^t|\\ &\le |(\overline{\alpha}^s)^T\Lambda^t(\overline{\alpha}^s-\overline{\alpha}^t)|+|(\overline{\alpha}^s-\overline{\alpha}^t)^T\Lambda^t\overline{\alpha}^t|\\ &\le \|\overline{\alpha}^s\|\|\Lambda^t\|\|\overline{\alpha}^s-\overline{\alpha}^t\|+\|\overline{\alpha}^s-\overline{\alpha}^t\|\|\Lambda^t\|\|\overline{\alpha}^t\|\le 2C\lambda_R\Delta_\alpha\end{aligned}$$

where the last equality follows from the fact that the matrix norm $\|\Lambda^t\|$ is bounded by the largest eigenvalue of Λ^t, which is smaller than λ_R by our assumption.

Lastly, the third term in (A.1) can be bounded as

$$\begin{aligned}|(\overline{\alpha}^t)^T\Lambda^t\overline{\alpha}^t-(\overline{T}\overline{\alpha}^t)^T\Lambda^t\overline{T}\overline{\alpha}^t| &\le |(\overline{\alpha}^t)^T\Lambda^t\overline{\alpha}^t-(\overline{\alpha}^t)^T\Lambda^t\overline{T}\overline{\alpha}^t\quad+(\overline{\alpha}^t)^T\Lambda^t\overline{T}\overline{\alpha}^t-(\overline{T}\overline{\alpha}^t)^T\Lambda^t\overline{T}\overline{\alpha}^t|\\ &\le |(\overline{\alpha}^t)^T\Lambda^t(\overline{\alpha}^t-\overline{T}\overline{\alpha}^t)|+|(\overline{\alpha}^t-\overline{T}\overline{\alpha}^t)^T\Lambda^t\overline{T}\overline{\alpha}^t|\\ &\le \|\overline{\alpha}^t\|^2\|\Lambda^t\|\|I-\overline{T}\|+\|\overline{\alpha}^t\|^2\|I-\overline{T}\|\|\Lambda^t\|\|\overline{T}\|.\end{aligned} \tag{A.2}$$

Bounding the norm of $\overline{T}$ as

$$\|\overline{T}\|=\|I+\overline{T}-I\|\le\|I\|+\|\overline{T}-I\|\le 1+\Delta_T$$

and using also the assumption $\|\overline{T}-I\|\le\Delta_T$ in (A.2), we get

$$|(\overline{\alpha}^t)^T\Lambda^t\overline{\alpha}^t-(\overline{T}\overline{\alpha}^t)^T\Lambda^t\overline{T}\overline{\alpha}^t|\le C^2\lambda_R\Delta_T+C^2\lambda_R\Delta_T(1+\Delta_T).$$

Finally, putting together the upper bounds for all the three terms in (A.1), we get the stated result

$$|(f^s)^TL^sf^s-(f^t)^TL^tf^t|\le C^2\delta+2C\lambda_R\Delta_\alpha+C^2\lambda_R(2\Delta_T+\Delta_T^2).$$

□