Domain Adaptation on Graphs by Learning Aligned Graph Bases

Mehmet Pilancı and Elif Vural

Abstract—A common assumption in semi-supervised learning with graph models is that the class label function varies smoothly on the data graph, resulting in the rather strict prior that the label function has low-frequency content. Meanwhile, in many classification problems, the label function may vary abruptly in certain graph regions, resulting in high-frequency components. Although the semi-supervised estimation of class labels is an ill-posed problem in general, in several applications it is possible to find a source graph on which the label function has similar frequency content to that on the target graph where the actual classification problem is defined. In this paper, we propose a method for domain adaptation on graphs motivated by these observations. Our algorithm is based on learning the spectrum of the label function in a source graph with many labeled nodes, and transferring the information of the spectrum to the target graph with fewer labeled nodes. While the frequency content of the class label function can be identified through the graph Fourier transform, it is not easy to transfer the Fourier coefficients directly between the two graphs, since no one-to-one match exists between the Fourier basis vectors of independently constructed graphs in the domain adaptation setting. We solve this problem by learning a transformation between the Fourier bases of the two graphs that flexibly “aligns” them. The unknown class label function on the target graph is then reconstructed such that its spectrum matches that on the source graph while also ensuring the consistency with the available labels. The proposed method is tested in the classification of image, online product review, and social network data sets. Comparative experiments suggest that the proposed algorithm performs better than recent domain adaptation methods in the literature in most settings.

Index Terms—Domain adaptation, data classification, graph Fourier transform, graph Laplacian, spectrum transfer.

1 INTRODUCTION

Most classification algorithms rely on the assumption that the labeled and unlabeled data samples at hand are drawn from the same distribution. However, in many practical data classification problems, the labeled training samples and the unlabeled test samples may have different statistics [1]. Domain adaptation methods make use of the class labels sufficiently available in a source domain in order to infer the label information in a target domain where labeled data are much more scarce. In order to be able to “transfer” the information from one domain to another, some inherent relation must exist between the two domains. In this work, we focus on a setting where the source and the target data are represented with a graph in each domain. We consider that the source and the target graphs are related in such a way that the spectra of the source and the target class label functions on the two graphs share similar characteristics. We then propose a method that makes use of this relation in order to estimate the missing labels in the target domain based on the sufficiently available label information in the source domain.

The domain adaptation problem has attracted much attention in the recent years. Each domain adaptation solution is based on a certain assumption about how the source and the target domains are related. Some methods assume that the data samples from different domains can be aligned via projections and transformations [2], [3], while some try to establish a relation between their distributions [4], [5], or learn joint feature representations [6]. Meanwhile, what is common between all these methods is that they are strictly based on the assumption that the data samples reside in an ambient space such as an Euclidean domain, hence they have physical coordinates. Although this may be true in various settings, there are also many data classification problems where the source and the target data are defined or described solely through the pairwise affinities or the relations between data samples. Some examples are social networks [7], where no physical coordinates are associated with a user but relations or links between different users define the network; or sensor networks [8], where the pairwise similarities between different sensors are identifiable via their geographical or other kinds of proximities. Graph models provide very convenient tools for such problems. For instance, in a social network each user can be represented as a graph node and relationships between users can be captured with edges. One can then consider an inference problem on the graph, e.g., whether a user is likely to be interested in a product or not. Similarly, in a sensor network one may infer the missing data at a broken sensor based on the data obtained from the other sensors.

In this work, we propose a new domain adaptation method that uses a source graph and a target graph representing the source and the target data. We consider the problem of estimating a label function on the target graph where very few labels are available. Depending on the application, the label function we consider can be any function defined on a graph domain, whose missing values are to be inferred from the available values. In particular, in a data classification problem, which is the main application area of our work, the values of the label function are class labels.
Our assumption about the relation between the source and the 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. 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 on the source graph.

Harmonic analysis on graph domains, which permits the definition of the Fourier transform on graphs, has been an active and popular research topic of the recent years [9], [10]. However, the notion of smoothness or smoothly-varying functions on graphs has actually been essential to many dimensionality reduction and semi-supervised learning methods since a long time [11], [12], [13]. Graph-based semi-supervised learning algorithms in a single domain typically rely on the assumption that the label function has a smooth variation on the graph [14], [15]. Meanwhile, the validity of the smoothness assumption is questionable in the general sense. For instance, in Figure 1 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. Consequently, the label function has fast variation along certain directions on the data graph and its spectrum contains some non-negligible high-frequency content. Although 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 actually be learnt in a setting with more than one domain. Our method 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 as illustrated in Figure 2.

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. In particular, the “aligned” source and target bases are such that the coefficients of the source and target label functions are similar when represented in the corresponding bases. We formulate the basis learning problem as the learning of a linear transformation between the source and the 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 the target label functions under the constraint that the source and the 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 thus minimize the objective function with an alternating optimization procedure. The output of the algorithm is the estimated label function on the target graph, which provides the class labels of the initially unlabeled data samples. To the best of our knowledge, our treatment is the first to study the domain adaptation problem based on explicitly analyzing the spectrum of label functions in a pure graph setting. Our proposed method is applicable not only to data analysis problems defined purely on graphs, but also to data embedded in an ambient space via the construction of graphs with respect to, e.g., nearest-neighborhoods. We demonstrate the usage of the algorithm in several data classification and regression applications. Classification results on synthetic data, face and object images, and social network data as well as regression results for the prediction of product ratings of users show that the proposed algorithm often outperforms traditional classifiers and reference domain adaptation methods in comparison.

The paper is organized as follows. In Section 2 we overview the related literature. In Section 3 we present a brief introduction to frequency analysis on graphs and formulate the problem of graph domain adaptation. In Section 4 we describe the proposed algorithm for domain adaptation on graphs via spectral graph alignment. In Section 5 we evaluate the performance of the algorithm with comparative experiments. Finally, we conclude in Section 6.

2 RELATED WORK

The domain adaptation problem has been treated in several settings and under different assumptions so far [1]. Some works focus on a problem where the source and the target distributions are defined on the same data space [17], [18], [19]. In the case that the conditional distributions of labels remain unchanged and only the marginal distributions of
data coordinates vary between the source and target domains, the domain adaptation problem is referred to as co-
variate shift or sample selection bias, where solutions based on sample reweighting are applicable [17], [20]. Daumé III et al. and Duan et al. have proposed to map the source and the target features to a higher dimensional domain via feature augmentation, where a common classifier can be learnt [21], [22], [23], [24]. In settings with multiple sources domains, a common approach is to learn the target hypothesis based on a weighted combination of the source hypotheses [25], [26].

Another domain adaptation solution consists of learning a transformation or a projection that aligns the source and the target data [2], [3], [27], [28], [29], [30], [31]. In fact, the idea of aligning the source and the target domains by mapping them to an intermediate space through a transformation has been at the core of many domain adaptation algorithms, some of which can also be applied to problems where the source and the target samples reside in different ambient spaces [32]. Several authors have proposed to reduce the distance between the samples from different domains by learning a transformation that maps the source and the target domains via copula functions or transformations. A metric adapted to the domain adaptation problem is learnt in [42], [43]. In some works, a classifier is learnt in a joint manner with the mapping [44], [45], or directly in the original data domain based on a self-training principle [46].

Deep networks have also gained popularity in domain adaptation applications in the recent years. These methods are typically based on the extraction of domain-invariant features that are shared between [6], [47] or adapted specifically [48] to the source and the target domains. Domain classifier layers aiming to reduce the distribution discrepancy are often learnt along with the label predictors in an adversarial manner [48], [49].

While all of the above domain adaptation methods rely strictly on the availability of a representation of the data in an ambient space, in this study, we focus on a setting where the data does not need to have a physical embedding and the problem may be directly defined over an abstract data 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 works [50], [51], which provides a foundation for graph signal processing. 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 [9], [10], [52].

The idea of matching graph bases with transformations or pairwise correspondences has been explored before in the previous works [53], [54], [55]; however, in different settings related to unsupervised clustering or 3D shape analysis problems. Note that, several previous methods have already incorporated manifold models or graph models in domain adaptation. The algorithm in [32] employs a manifold model and learns projections by preserving the topology of the data set while achieving discrimination between different classes. The works in [45], [56], [57] similarly impose priors on the smoothness of the label function over the data graph. The K-NN graphs used in [57] are iteratively refined with the aid of a supervised metric learner. A pair of source and target graphs are constructed in [58], which is followed by a graph matching stage to map source classes to target clusters for multispectral image classification.

Finally, a preliminary version of our work has been presented in [59], where the idea of transferring the label spectrum between a source and a target graph has been explored for the first time. However, a major limitation of the algorithm in [59] is that it relies on a one-to-one match between the graph Fourier basis vectors. This restricts its applicability to settings where the source and the target graphs are highly similar so as to admit a direct match between the two graph Fourier bases. This limitation is circumvented in the current study by learning a transformation between the two Fourier bases.

3 DOMAIN ADAPTATION ON GRAPHS

In this section, we first give an overview of the extension of classical frequency analysis techniques to graph domains [10]. Then, we propose a problem formulation for domain adaptation on graphs. In the following, matrices are represented with uppercase letters, and vectors are denoted with lowercase or Greek letters. Vectors are considered as column vectors unless stated otherwise. Aij stands for the (i, j)-th entry of a matrix A, and |·| denotes the cardinality of a set.

3.1 Overview of Frequency Analysis on Graphs

In graph-based methods, a data set with N data samples is typically represented with a graph with N vertices, such that each vertex corresponds to a data sample. Let G = (V, E, W) be a weighted graph with N vertices (nodes), where $V = \{v_i\}_{i=1}^{N}$ is the set of vertices, $E$ is the set of edges, and $W \in \mathbb{R}^{N \times N}$ is the weight matrix. If there is an edge between the nodes $x_i$ and $x_j$, then $W_{ij}$ consists of the weight of this edge. If the nodes $x_i$ and $x_j$ are not connected with an edge, then $W_{ij} = 0$.

A graph signal is a function $f : V \rightarrow \mathbb{R}$ taking a real value on each graph vertex, which can equivalently be represented as an N-dimensional vector $f \in \mathbb{R}^{N}$. A set $\{v_k\}_{k=1}^{N} \subset \mathbb{R}^{N}$ of linearly independent graph signals form a graph basis, so that any graph signal $f$ can be represented as

$$f = \sum_{k=1}^{N} \alpha_{k}v_{k}$$  \hspace{1cm} (1)$$

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

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

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

(2)

is the graph equivalent of applying the Laplacian operator to a signal in classical signal processing [10], [50], [51]. This analogy allows the extension of classical Fourier analysis to graph domains as follows. First recall that for one-dimensional signals, the complex exponentials \( e^{j\Omega k} \) defining the Fourier transform are given by the eigenfunctions of the Laplacian operator \( \Delta \)

\[
-\Delta(e^{j\Omega k}) = \Omega^2 e^{j\Omega k}.
\]  

(3)

The eigenvalue \( \Omega^2 \) of the Laplacian operator increases with the frequency of the complex exponential \( e^{j\Omega k} \). Characterizing the Fourier transform via the eigenfunctions of the Laplacian operator, the graph counterparts of complex exponentials are then the eigenvectors \( \{u_k\}_{k=1}^{N} \) of the graph Laplacian given by

\[
Lu_k = \lambda_k u_k.
\]  

(4)

The set of eigenvectors \( \{u_k\}_{k=1}^{N} \subset \mathbb{R}^N \) of the graph Laplacian corresponding to the eigenvalues \( \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N \) thus defines a graph Fourier basis. In analogy with (3), the eigenvalues \( \lambda_k \) bear a notion of frequency in a graph. The eigenvectors \( u_k \) for increasing values of \( k \) indeed have an increasing speed of variation over the graph when regarded as graph signals [10]. In particular, a common measure for the speed of variation of a graph signal \( f \) over the graph is

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

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

\[
u_k^T Lu_k = \lambda_k.
\]  

(6)

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

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

(7)

This can be equivalently written as \( \hat{f} = U^T f \in \mathbb{R}^N \) in matrix notation, where \( \hat{f} = [\hat{f}(\lambda_1) \ldots \hat{f}(\lambda_N)]^T \) and \( U = [u_1 \ldots u_N] \in \mathbb{R}^{N \times N} \). Here \( \hat{f}(\lambda_k) \) is the \( k \)-th Fourier coefficient of \( f \) corresponding to the basis vector \( u_k \) with frequency \( \lambda_k \). The inverse Fourier transform is then obtained as the reconstruction of the signal from its representation over the Fourier basis as

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

(8)

### 3.2 Problem Formulation for Graph Domain Adaptation

We now propose our problem formulation for domain adaptation in graph settings. We consider a source graph \( G^s = (V^s, E^s, W^s) \) that consists of \( N_s \) vertices \( V^s = \{x^s_i\}_{i=1}^{N_s} \) and edges \( E^s \), and a target graph \( G^t = (V^t, E^t, W^t) \) with \( N_t \) vertices \( V^t = \{x^t_i\}_{i=1}^{N_t} \) and edges \( E^t \). The weighted edges of the source and the target graphs are respectively represented in the weight matrices \( W^s \), \( W^t \). Let \( U^s \in \mathbb{R}^{N_s \times N_s} \) and \( U^t \in \mathbb{R}^{N_t \times N_t} \) denote the matrices containing the Fourier basis vectors, respectively on the source and the target graphs. These are computed using the eigenvalue decompositions of the respective graph Laplacians \( L^s \in \mathbb{R}^{N_s \times N_s} \) and \( L^t \in \mathbb{R}^{N_t \times N_t} \) as explained in [4].

Consider a label function \( f^s \in \mathbb{R}^{N_s} \) on the source graph and a label function \( f^t \in \mathbb{R}^{N_t} \) on the target graph. We assume that the labels of some of the nodes are available. We denote the known labels as \( y^s_i = f^s(x^s_i) \) on the source graph (for labeled \( x^s_i \)), and as \( y^t_i = f^t(x^t_i) \) on the target graph (for labeled \( x^t_i \)). The sets containing the indices of the labeled data samples are denoted as \( S^s \subset \{1, \ldots, N_s\} \) and \( T^t \subset \{1, \ldots, N_t\} \) in the source and the target domains. The label functions \( f^s \) and \( f^t \) take discrete values in a classification problem and continuous values in a regression problem. For instance, in a classification problem with two classes, one can set \( y^s_i \) as equal to 1 if the labeled data sample \( x^s_i \) belongs to the first class and as 0 if it is from the second class. The problem is then to compute the labels of all unlabeled data samples, which is done by estimating the label vector \( f^t \). Domain adaptation methods often assume a setting with many labeled samples in the source domain and much fewer labeled samples in the target domain, i.e., \( |S^s| \ll |T^t| \).

Let \( V^s \in \mathbb{R}^{N_s \times N_s} \) and \( V^t \in \mathbb{R}^{N_t \times N_t} \) denote a pair of bases for the functions respectively on the source and the target graphs. 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

\[
\begin{align*}
 f^s &= \sum_{k=1}^{N_s} \alpha_k^s v_k^s, \\
 f^t &= \sum_{k=1}^{N_t} \alpha_k^t v_k^t.
\end{align*}
\]  

(9)

Here \( V^s \) and \( V^t \) contain respectively the basis vectors \( \{v_k^s\} \) and \( \{v_k^t\} \) in their columns; and \( \alpha^s \in \mathbb{R}^{N_s} \) and \( \alpha^t \in \mathbb{R}^{N_t} \) are 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 the 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.**

\[
\min_{\alpha^s, \alpha^t} \|S^s V^s \alpha^s - y^s\|^2 + \|S^t V^t \alpha^t - y^t\|^2 + \mu \|\alpha^s - \alpha^t\|^2
\]  

(10)
Here $y^s \in \mathbb{R}^{K_s}$ and $y^t \in \mathbb{R}^{K_t}$ are vectors consisting respectively of the available labels $\{y^s_i\}$ and $\{y^t_i\}$ in the source and the target domains where $K_s = |I^s|$ and $K_t = |I^t|$ are the number of known labels. The matrices $S^s \in \mathbb{R}^{K_s \times N_s}$ and $S^t \in \mathbb{R}^{K_t \times N_t}$ are binary selection mask matrices consisting of 0's and 1's, which enforce the label prediction functions $f^s$, $f^t$ to match the available labels $y^s$, $y^t$ over the subsets $I^s$, $I^t$ of labeled data; and $\mu > 0$ is a weight parameter. The coefficients $\alpha^s$ and $\alpha^t$ of the source and target label functions must be found such that the resulting estimation and the target domains where

\begin{equation}
\{ x^t_i \} \subseteq \mathbb{R}^{N_t} \quad \text{are close over the source and the target graphs.}
\end{equation}

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 exists between the source and the target graphs, 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 Fourier bases $V^s = U^s$, $V^t = U^t$, so that the spectra of the source and the target label functions can be directly matched by solving the problem in (10). However, in a realistic setting such a one-to-one match often does not exist. For instance, the experiments reported in Section 5.1, Figure 9 study the frequency content of the label function on the source and the target graphs. The results in Figure 9 indicate that the general shape (envelope) of the spectrum resembles between the two graphs; however, corresponding Fourier coefficients across the two graphs are not always the same. This suggests that although it would be too restrictive to transmit the exact Fourier coefficients, it is possible to exploit the similarity between the shapes of the source and the target spectra. Based on these observations, 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. In particular, we propose to choose

\begin{equation}
V^s = U^s, \quad V^t = U^t T
\end{equation}

where $U^s$ and $U^t$ are the Fourier bases, and the matrix $T \in \mathbb{R}^{N_t \times N_t}$ represents a transformation between $U^t$ and $V^t$. From Problem 1, one can observe that $T$ matches the source basis vector $v^s_i = u^s_i$ to the target basis vector

\begin{equation}
v^t_i = \sum_{j=1}^{N_t} T_{ij} u^s_j
\end{equation}

obtained as a linear combination of the Fourier vectors $u^s_j$.

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 $v^s_i = u^s_i$ is proportional to the $i$-th eigenvalue $\lambda_i^s$ of the source graph Laplacian $L^s$. In order to preserve the spectral relation between the graphs, the corresponding target vector $v^t_i$ in (12) 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. For this reason, we propose to learn $T$ such that

\begin{equation}
\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 \\
&\quad + \mu_1 \| \alpha^s \|^2 + \mu_2 \| M \odot T \|^2_F \\
\text{subject to } &\sum_{i=1}^{N_t} T_{ij}^2 = 1, \text{ for } j = 1, \ldots, N_t.
\end{aligned}
\end{equation}

Here $\mu_1 > 0$, $\mu_2 > 0$ are weight parameters, and $\| \cdot \|_F$ denotes the Frobenius norm of a matrix. The equality constraints ensure that the columns of the transformation matrix $T$ have unit norm, in order not to approach the trivial solution $T = 0$.

While Problem 2 aims to learn a pair of complete bases on the two graphs, it is often not necessary to use all basis vectors for obtaining a good reconstruction of the label function: Fourier basis vectors $u^s_i$, $u^t_i$ with very high frequencies (eigenvalues) $\lambda_i^s$, $\lambda_i^t$ 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, we select a subset of the basis vectors $\{u^s_i\}_{i=1}^R$, $\{u^t_i\}_{i=1}^R$, corresponding to the $R$ smallest frequencies in both domains, where $R < N_s$ and $R < N_t$. Let $U^s \in \mathbb{R}^{N_s \times R}$, $U^t \in \mathbb{R}^{N_t \times R}$ denote the reduced source and target Fourier bases consisting of the
functions have similar rates of variation if the constants \( \delta, \lambda_R, \Delta_\alpha, \Delta_T \) are sufficiently small. The constant \( \delta \) depends on the topological similarity between the two graphs and cannot be controlled by the learning algorithm. Meanwhile, the constant \( \lambda_R \) in the above bound suggests that preventing \( \lambda_R \) 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 \( \pi^s \) and \( \pi^t \), which reduces \( \Delta_\alpha \). Finally, the term \( \| \mathbf{M} \circ \mathbf{T} \|_F \) in the learning objective aiming to discourage large off-diagonal entries will eventually help reduce the constant \( \Delta_T \) in the above bound. Note, however, that we deliberately avoid imposing \( \mathbf{T} \approx \mathbf{I} \) 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.

### 4 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 (15) is not jointly convex in all optimization variables \( \pi^s, \pi^t, T \). Nevertheless, it is convex separately in the overall coefficient vector \( \pi = [(\pi^s)^T (\pi^t)^T]^T \), and the transformation matrix \( T \). Hence, we propose to minimize the objective (15) with an iterative and alternating optimization approach, by first fixing \( T \) and optimizing \( \pi^s, \pi^t \); and then fixing the coefficient vectors \( \pi^s, \pi^t \) and optimizing \( T \) in each iteration. We describe these two optimization steps in the sequel.

#### 4.1 Optimization of the Coefficient Vectors

In the first step of an iteration, the transformation matrix \( T \) is fixed, and the coefficient vectors \( \pi^s \) and \( \pi^t \) are optimized. Fixing \( T \), the optimization problem in (15) becomes the following unconstrained problem in \( \pi^s \) and \( \pi^t \):

\[
\min_{\pi^s, \pi^t} G(\pi^s, \pi^t) = \min_{\pi^s, \pi^t} \| S^s \mathbf{U}^T \pi^s - y^s \|^2 \\
+ \| S^t \mathbf{U}^T \pi^t - y^t \|^2 + \mu_1 \| \pi^s - \pi^t \|^2.
\]

(17)

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

\[
\frac{\partial G(\pi^s, \pi^t)}{\partial \pi^s} = 2 A^s \pi^s - 2 B^s y^s + 2 \mu_1 \pi^s - 2 \mu_1 \pi^t = 0
\]

(18)

\[
\frac{\partial G(\pi^s, \pi^t)}{\partial \pi^t} = 2 A^t \pi^t - 2 B^t y^t + 2 \mu_1 \pi^t - 2 \mu_1 \pi^s = 0
\]

where

\[
A^s = (\mathbf{U}^s)^T (S^s)^T S^s \mathbf{U}^s, \quad B^s = (\mathbf{U}^s)^T (S^s)^T A^t = (\mathbf{U}^t)^T (S^t)^T S^t \mathbf{U}^t, \quad B^t = (\mathbf{U}^t)^T (S^t)^T A^t.
\]

(19)
This gives the coefficient vectors as
\[ \alpha^s = (\mu_1^{-1} A^T \alpha^s + A^T + A^s)^{-1}(\mu_1^{-1} A^T B^s y^s + B^s y^s + B^T y^t) \]
\[ \alpha^t = (\mu_1^{-1} A^s \alpha^s + \alpha^s - \mu_1^{-1} B^s y^s). \]
(20)

4.2 Optimization of the Transformation Matrix

In the second step of an iteration, the coefficient vectors \( \alpha^s \) and \( \alpha^t \) are fixed and the transformation matrix \( T \) is optimized. Then the minimization of the objective in (15) becomes equivalent to the following problem

\[
\min_{T} H(T) = \min_{T} \| S^T U^T T \alpha^t - y^t \|^2 + \mu_2 \| M \odot T \|^2_F
\]

subject to \( \sum_{i=1}^{R} T^2_{ij} = 1 \), for \( j = 1, \ldots, R \).

(21)

The above problem involves the minimization of a quadratic convex function \( H(T) \) in \( T \) subject to \( R \) equality constraints that are also quadratic and convex in \( T \). We solve the problem in (21) using the Sequential Quadratic Programming (SQP) algorithm [60], 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 (21), the objective function \( H(T) \) is already a quadratic function of \( T \) and we only have equality constraints.

The first and second order derivatives to be used in the solution of (21) are found as follows. Let \( \bar{T} \in \mathbb{R}^{R^2} \) denote the column-wise vectorized form of the matrix \( T \), such that its \( k \)-th entry is given by \( \bar{T}_{j} = T_{ij} \), with \( k = (j-1)R + i \), for \( i, j = 1, \ldots, R \). We denote by \( h(\bar{T}) = H(T) \) the objective function (21) when considered as a function of \( \bar{T} \). The objective function \( h(T) = H(T) \) can then be rewritten in terms of \( \bar{T} \) as

\[
h(\bar{T}) = \| A \bar{T} - y^t \|^2 + \mu_2 \| F \bar{T} \|^2_F.
\]
(22)

Here \( A \in \mathbb{R}^{L_i \times R^2} \) is a matrix with entries given by \( A_{lk} = (S^T U^T)_{li} \alpha_j^t \) and \( F \in \mathbb{R}^{R^2 \times R^2} \) is a diagonal matrix with entries given by \( F_{kk} = M_{ij} \), where \( l = 1, \ldots, L_i \) and \( k = (j-1) + i \), for \( i, j = 1, \ldots, R \). The variable \( L_i \) here is the number of labeled target samples. Next, the \( j \)-th equality constraint of the problem (21) can be written in terms of \( \bar{T} \) as

\[
g_j(\bar{T}) = \sum_{i=1}^{R} T_{ij}^2 - 1 = 0
\]
(23)

for \( j = 1, \ldots, R \).

The problem (21) is then solved by forming the Lagrangian function

\[
L(\bar{T}, \eta) = h(\bar{T}) - g(\bar{T}, \eta)
\]
(24)

where

\[
g(\bar{T}, \eta) = \sum_{j=1}^{R} \eta_j g_j(\bar{T}),
\]
(25)

\( \eta_j > 0 \) are the Lagrange multipliers, and \( \eta = [\eta_1 \ldots \eta_R]^T \).

From (22), we obtain the gradient of the objective \( h(\bar{T}) \) as

\[
\nabla_{\bar{T}} h = 2(A^T A + \mu_2 F^T F) \bar{T}
\]
(26)

and its Hessian as

\[
\nabla_{\bar{T}}^2 h(\bar{T}) = 2(A^T A + \mu_2 F^T F).
\]
(27)

Next, from (23), the \( k \)-th entry of the gradient of \( g_j(\bar{T}) \) is found as

\[
(\nabla_{\bar{T}} g_j)_{k} = \begin{cases} 2\bar{T}_{kj}, & \text{if } (j-1)R + 1 \leq k \leq jR \\ 0, & \text{otherwise} \end{cases}
\]
(28)

for \( k = 1, \ldots, R^2 \). From (25), the Hessian \( \nabla_{\bar{T}}^2 \eta g(\bar{T}, \eta) \) of the second term \( g(\bar{T}, \eta) \) of the Lagrangian in (25) is obtained as a diagonal matrix with entries given by

\[
[\nabla_{\bar{T}}^2 \eta g(\bar{T}, \eta)]_{kk} = 2\eta_j
\]
(29)

for \( R(j-1) + 1 \leq k \leq Rj \). Putting (27) and (29) together, we get the Hessian of the Lagrangian as

\[
\nabla_{\bar{T}}^2 L(\bar{T}, \eta) = \nabla_{\bar{T}}^2 h(\bar{T}) - \nabla_{\bar{T}}^2 \eta g(\bar{T}, \eta).
\]
(30)

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 [60] 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.

Remark: Although the SQP algorithm often converges to a solution in practice, it is not easy to establish a general theoretical convergence guarantee. For our problem, the convergence can be theoretically guaranteed under certain conditions: Let the algorithm parameter \( \mu_2 \) be chosen such that there exists a local solution \( (\bar{T}^*, \eta^*) \) to the Lagrangian function \( L(\bar{T}, \eta) \) of the constrained problem (21) such that \( \mu_2 > \eta_{j}^* \) for all \( j = 1, \ldots, R \). Then, if the initialization \( (\bar{T}, \eta) \) of the SQP algorithm is sufficiently close to \( (\bar{T}^*, \eta^*) \), the algorithm converges to \( (\bar{T}^*, \eta^*) \). The details of this convergence analysis are provided in Appendix B.

4.3 Overall Optimization Procedure

We now overview the overall optimization procedure employed in the proposed DASGA method. First, the optimization variables \( T, \alpha^s \), and \( \alpha^t \) 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 \( T \) 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 \( \alpha^s \), \( \alpha^t \) and affect the overall solution of the alternating optimization procedure.

In order to obtain a more favorable initialization, we propose to set the initial \( T \) 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 previous work [59], where the best match of a target vector \( u^t_i \) among the source vectors is determined by finding

\[
\max_j | \langle \tilde{u}^s_j, \tilde{u}^t_i \rangle |.
\]  

(31)

Here \( \tilde{u}^s_j, \tilde{u}^t_i \) are subvectors of the basis vectors \( u^s_j, u^t_i \) obtained by restricting them to a subset of their entries indexed by \( \{ s_i \}_{i=1}^{K} \) and \( \{ t_i \}_{i=1}^{K} \). It is difficult to directly compare the vectors \( u^s_j, u^t_i \) 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 \( \mathcal{N} = \{ (x^s_i, x^t_i) \}_{i=1}^{K} \) is known, then this set can be used for the restriction of the basis vectors to a subset of their entries in the problem [51], so that the vectors \( u^s_j, u^t_i \) 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 \( \mathcal{N} = \{ (x^s_i, x^t_i) \}_{i=1}^{K} \) based on the class labels, such that each pair of matched nodes \( (x^s_i, x^t_i) \) is formed randomly among the source and target nodes having the same class labels. We then compare the vectors \( u^s_j, u^t_i \) over their entries \( \tilde{u}^s_j, \tilde{u}^t_i \) 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 initializing \( \mathcal{T} \), which is next done as follows

\[
\mathcal{T}_{ii} = \text{sgn}(\langle \tilde{u}^s_j, \tilde{u}^t_i \rangle), \quad J_i = \arg \max_j | \langle \tilde{u}^s_j, \tilde{u}^t_i \rangle |.
\]

(32)

Here \( \text{sgn} \) denotes the sign function and \( \mathcal{T} \) is initialized as a diagonal matrix with \(-1\)’s or \(1\)’s on the diagonals that matches the sign of each target vector \( u^t_i \) to the source vector \( u^s_j \) best corresponding to it. This initialization respects the normalization constraint [15] on the entries of \( \mathcal{T} \).

Once the transformation matrix \( \mathcal{T} \) is initialized in this way, the alternating optimization procedure starts, where the coefficient vectors \( \tilde{\pi}^s \) and \( \tilde{\pi}^t \) are computed by fixing \( \mathcal{T} \) first, and then \( \mathcal{T} \) is optimized by fixing \( \tilde{\pi}^s \) and \( \tilde{\pi}^t \) in each iteration, as described in Sections 4.1 and 4.2. In each iteration, both the updates on \( \tilde{\pi}^s \) and \( \tilde{\pi}^t \), and the update on \( \mathcal{T} \) either reduce or retain the value of the objective function in (15). 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.

### 4.4 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 (17), 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 \( \mathcal{T} \) and they are computed only once; hence, we may ignore their calculation in the overall complexity. Next, \( O(N_s R^2 + L_s N_s R + L_s R^2) \) operations are needed to compute the matrices \( B^t \) and \( A^t \) respectively. The matrices \( \mu_1^{-1} A^t B^s y^s + A^t B^t y^t \) and \( \mu_1^{-1} A^t B^s y^s + B^t y^t \) in the expression of \( \tilde{\pi}^s \) are computed respectively with \( O(R^3) \) and \( O(L_s R^2 + L_s R) \) operations. Considering also the matrix inversion in its expression, \( \tilde{\pi}^s \) is computed with \( O(R^3) \) operations. The target coefficients \( \tilde{\pi}^t \) are then obtained from \( \tilde{\pi}^s \) with \( O(R^2) \) operations. From the complexities of all these computations, we get the overall complexity of Step 4 of Algorithm 1 as \( O(R^3 + (L_s + N_s)R^2 + L_s N_s R) \).

Next, we examine the complexity of executing Step 5 with the SQP algorithm. The complexity of the evaluation of \( h(\tilde{t}) \) in (22) is of \( O(L_s R^2 + R^2) \). From (26), we observe that the gradient \( \nabla_{\tilde{t}} h(\tilde{t}) \) is computed with \( O(R^2) \) operations as well. Finally, since the Hessian \( \nabla_{\tilde{t}}^2 h(\tilde{t}) \) of the objective in (27) is a constant matrix that does not depend on \( \tilde{t} \), we can exclude it from the complexity of the iterative SQP algorithm. Next, from (23), the complexity of computing all \( R \) gradients is obtained as \( O(R^2) \). From (28) and (29), we observe that the gradients \( \nabla_{\eta} g(\tilde{t}, \eta) \) and the Hessian \( \nabla_{\tilde{t}}^2 g(\tilde{t}, \eta) \) are obtained directly from \( \tilde{t} \) and \( \eta \) without any operations. We thus conclude that the Hessian \( \nabla_{\eta}^2 L(\tilde{t}, \eta) \) of the Lagrangian can also be obtained with negligible complexity. Finally, the optimization variables are updated by solving the linear system given in Algorithm 18.1 with \( O(R^2) \) 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 \( O(R^6 + L_s R^2) \).

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

### 5 Experimental results

In the following, we first introduce the datasets and then evaluate the performance of the proposed method with comparative experiments. Next, we study the behavior of the algorithm throughout the iterative optimization pro-

---

**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 \( \mathcal{T} \) as in (32).
3. **repeat**
4.   - Update coefficients \( \tilde{\pi}^s, \tilde{\pi}^t \) by solving (17).
5.   - Update transformation matrix \( \mathcal{T} \) by solving (21).
6. **until** the objective function (15) converges
7. **Output:**
   - \( f^t = U^T \mathcal{T} \tilde{\pi}^t \): Estimated target label function
   - \( f^s = U^T \tilde{\pi}^s \): Estimated source label function
The following data sets are used in the experiments.

5.1 Data sets

The following data sets are used in the experiments.

Synthetic data sets. The two synthetic data sets shown in Figures 4(a) and 4(b) are generated by drawing 100 samples for each class from a normal distribution in \( \mathbb{R}^3 \), with different means for the two classes. The means of the source classes and the corresponding target classes are symmetric along the \( y \)-direction. The variance of the distribution is higher in Synthetic dataset-2 compared to Synthetic dataset-1; hence the difficulty level of the classification task is higher. The source and the target graphs are constructed by connecting each data sample to their 25 nearest neighbors. The edge weights are computed with a Gaussian kernel as \( w_{ij}^s = \exp(-\|x_i^s - x_j^s\|^2/\sigma^2) \) in the source graph and similarly in the target graph, where \( x_i^s \) and \( x_j^s \) are the data sample coordinates and the scale parameter \( \sigma \) is chosen proportionally to the typical distance between neighboring samples.

MIT-CBCL face image data set. The MIT-CBCL face recognition database [61] consists of a total of 3240 face images rendered from the 3D head models of 10 participants under varying illumination and poses. The images of each participant 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. We downsample the images to a resolution of 100 \( \times \) 100 pixels and consider the images taken under each pose as samples from a different domain. Raw features consisting of pixel intensities are used in the experiments. Two settings are considered, where the source domain is taken as Pose 1 in both settings. The target domain is taken as Pose 5 in the first setting, and as Pose 9 in the second setting. Source and target data graphs are constructed independently in the source and the target domains, by connecting each image to its nearest 38 neighbors with respect to the Euclidean distance. The weight matrices \( W^s, W^t \) are constructed with a Gaussian kernel as in the synthetic data sets.

COIL object image data set. The COIL-20 object database [62] consists of a total of 1440 images of 20 objects. Each object has 72 images taken from different viewpoints rotating around it. We downsample the images to a resolution of 32 \( \times \) 32 pixels. We consider a transfer learning setting 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 with respect to their similarity computed via pairwise distances. The experiments are then done by considering each group of 10 objects as a different domain, and regarding the images of the matched objects across the two domains as having the same class label. The two groups and the matched object pairs are shown in Figure 6.

The source and the target graphs are constructed by connecting each sample to its 3 nearest neighbors and the weights are set with a Gaussian kernel. A small number of neighbors 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.

Amazon product ratings data set. The Amazon data set [63] is used in the task of predicting user ratings on books. The data set contains 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 represents a user, and the scores that the users gave to the first and the second bestsellers are regarded as signals (label functions), respectively on the source and the target graphs. The source and the 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 consider the task of predicting the unavailable scores on the target bestseller.

Facebook data set. The Facebook data set [64] consists of various communities (friend circles) extracted from the Facebook network. Graph nodes and edges respectively represent Facebook users and their friendship relations. In our experiments two different communities are chosen as the source graph and the target graph. Isolated users or user cliques are removed and the weights of all edges are set to the constant value 1. The gender of the Facebook users is taken as the binary label function to be predicted. The source and the target graphs, consisting respectively of 157 and 40 users, are shown in Figure 7, where the values of the label function are represented with two different colors.

In the following, we first verify the validity of the main assumption of the proposed method that the frequency content of the label function is similar on the source and the target graphs. The Fourier coefficients of the source and the target label functions are plotted in Figures 8-12 for all data sets. Heavily concentrated at low frequencies, the source and the target label functions seem to have similar frequency contents for the two synthetic data sets in Figure 8. In Figure 9(a) the source and the target Fourier coefficients at the same frequency have quite similar magnitudes due to the high similarity between the source and the target images captured under nearby camera angles. On the other hand, in Figure 9(b) where the two domains bear smaller resemblance, the source and the target Fourier coefficients at the same frequency do not always have similar magnitudes. Nevertheless, the shape of the spectrum is similar between the source and the target graphs, with similar amplitudes at nearby frequencies. The results on the COIL-20 and Amazon data sets in Figures 10 and 11 can be interpreted similarly, as the spectra of the source and the target label functions decay with the frequency and have similar characteristics in both data sets. The label function is seen to have a rather flat spectrum in Figure 10 for the Facebook data set, on both the source and the target graphs (note that the two graphs have different sizes and all Fourier coefficients are plotted in the figure). This is due to the highly irregular nature of the label function, which is observable in Figure 7. The experiments on all of the data sets lead to the common conclusion that the assumption that the label function has similar frequency content on the source and the target graphs is realistic in practice.

5.2 Evaluation of the Algorithm Performance
The performance of the proposed DASGA method is compared to the domain adaptation methods Heterogeneous Domain Adaptation using Manifold Alignment (DAMA) [32], Easy Adapt++ (EA++) [23], Subspace Alignment (SA) [2], Geodesic Flow Kernel for Unsupervised Domain Adaptation (GFK) [3], Scatter Component Analysis (SCA) [40], LDA-Inspired Domain Adaptation (LDADA) [46], Joint Geometrical and Statistical Alignment (JGSA) [31], 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 [13]. The baseline classifiers SVM and NN are eval-
5.2.1 Experiments on synthetic data sets

The proposed DASGA algorithm is used with the parameters $\mu_1 = 0.1$, $\mu_2 = 1$, and $R = 9$ in the experiments with synthetic data sets. In Figure 13, the misclassification rates of unlabeled target samples in percentage are plotted with respect to the ratio of labeled target samples in percentage. The results are averaged over 50 repetitions of the experiment with random selections of the labeled samples. As expected, the misclassification rates of the algorithms tend to decrease as the ratio of known target labels increases. The proposed DASGA algorithm is observed to outperform the compared methods in both data sets. The performance gap between DASGA and the other methods is larger in Synthetic dataset-2, which is a more challenging data set consisting of normally distributed data. Among the domain adaptation methods, DAMA [32] and LDADA [46] give the closest performance to the proposed DASGA method. The approach in both of these methods is to learn supervised projections, which is relatively successful in this synthetic data set. However, the proposed graph-based DASGA algorithm does not use the pixel intensity values of image data samples. Instead, it relies on a pure graph representation of data, therefore its performance is less affected by the ambient space properties of the data. This feature is seen to provide some robustness against the challenges such as large variance and poor separation between the classes.

5.2.2 Experiments on image data sets

We next evaluate the performance of the proposed algorithm on the image data sets. In the experiments with the MIT-CBCL face image data set, the parameters of the proposed DASGA method are set as $\mu_1 = 0.1$, $\mu_2 = 0.85$, and $R = 9$. The experiment is repeated over 10 realizations with random selections of the labeled 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 14(a) and 14(b), where the source and the target domains are similar, the proposed DASGA method is seen to outperform the domain adaptation methods EA++ and JGSA, as well as the SA, GFK, and LDADA methods which yield almost zero error. Capturing the face images of the same participants from nearby poses in a clean and controlled environment, the two domains in this experiment are quite convenient to align with methods using projections and geometric transformations, which explains the success of these algorithms. On the other hand, the performance of DASGA algorithm does not use the pixel intensity values of image data samples. Hence, it does not employ the same type of information as these methods. Nevertheless, in Figure 14(b), where the source and target images differ more significantly, the performance of DASGA catches up with the other methods when the ratio of known target labels reaches 7%.

In the experiments with the COIL-20 image data set, the parameters of the proposed method are set as $\mu_1 = 1$, $\mu_2 = 1$ and $R = 10$. The misclassification rates of the algorithms are plotted with respect to the ratio of known target labels in Figure 15. The proposed DASGA method is observed to often yield the best classification performance. The misclassification rate of the proposed algorithm falls to zero when about 7% of the samples are labeled in the target domain.
domain. The graph-based semi-supervised learning algorithm SSL also performs well in this experiment. 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 performances of the domain adaptation methods SA, DAMA, and LDADA fall behind that of the simple NN classifier in this experiment. Relying on the alignment of the source and the target domains via transformations, these methods fail in the transfer learning problem considered in this experiment. The source and the target images belong to different objects; hence, they are difficult to align via linear projections or transformations. It is also interesting to note that the relatively more sophisticated SCA method based on nonlinear kernel transformations, is more successful in this challenging data set compared to the previous data sets of simpler structure.

5.2.3 Experiments on the Amazon book ratings data set

In the experiments with the Amazon book ratings data, the parameters of DASGA are set as $\mu_1 = 0.001$, $\mu_2 = 0.8$, and $R = 10$, 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 the coordinates of the data samples; thus, need an embedding of the data in an ambient space. Unlike the image data and the 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. Another solution could be to represent graph nodes using graph-theoretic features as in [65], [66]; however, such features should be selected and used carefully. In order to test the compared methods, we follow an alternative approach and embed the source and the target graphs into an Euclidean domain of optimal dimension using the Multidimensional Scaling (MDS) algorithm [67]. The coordinates learnt for each user with MDS are then used as training features by the compared algorithms.

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. The average misclassification rates of the score predictions (considering each score from 1 to 5 as a different class label) are plotted in Figure 16(a) and Figure 16(b) shows the root mean square (RMS) error of the predictions. The results in Figure 16(a) show that most of the methods including DASGA yield similar misclassification errors. Although DASGA does not provide smaller misclassification error than the other methods, Figure 16(b) shows that it clearly outperforms the other methods in terms of the RMS prediction error. The ensemble of the results in Figure 16 suggests that the proposed DASGA algorithm is well-fit to the regression problem inherent to this setting as it relies on the analysis of the rate of variation of the user rating functions over the graphs.
5.2.4 Experiments on the Facebook data set

As the Facebook data set involves a pure graph environment, the two graphs are embedded into an Euclidean domain via the MDS algorithm as in the Amazon data set in order to provide feature representations for the other algorithms than DASGA and SSL. The parameters of DASGA are set as $\mu_1 = 1$, $\mu_2 = 1$, and $R = 8$. The misclassification rates of the compared methods are presented in Figure 17. The classification errors of all methods are relatively high in this experiment, which can be explained by observing the challenging structure of the data set in Figure 7. The proposed DASGA method is seen to generally outperform the other methods. It is interesting to compare DASGA to the reference graph-based SSL method. When the ratio of available target labels is relatively small, DASGA performs better than SSL thanks to the information of the label spectrum transmitted from the source graph. Meanwhile, when the ratio of available target labels exceeds 50%, the SSL method has sufficient information to diffuse in the target graph and it can guess the label function more accurately than DASGA. This is coherent with the principle of domain adaptation: learning the label spectrum from an exemplar source graph improves the performance in the target graph when the label information is restricted in the target graph, which is typically the case in a domain adaptation problem.

5.3 Stabilization and Sensitivity Analysis of the Proposed Algorithm

We first study the behavior of the proposed DASGA algorithm throughout the iterative optimization procedure. We examine the variations of the objective function and the misclassification rate of target samples during the iterations. The value of the objective function $\mu$ 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 and the MIT-CBCL data sets in Figure 18. The results confirm that the objective function decreases monotonically throughout the iterations and converges as discussed in Section 4.3. 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 $\mu$ underlying the proposed method captures well the actual performance of classification.

Next, we study the sensitivity of the DASGA method to the choice of the algorithm parameters.

Sensitivity to the weight parameters $\mu_1$ and $\mu_2$. We first examine how the choice of the weight parameters affects the algorithm performance. The target misclassification rate of the algorithm is reported for various $(\mu_1, \mu_2)$ pairs in a region of interest in Tables 15 for all five data sets. The results indicate that although the optimal values of the $\mu_1$ and $\mu_2$ parameters may vary among different data sets, the $(\mu_1, \mu_2)$ pair yielding the smallest misclassification error is found within the region $\mu_1 \in [0, 1]$, $\mu_2 \in [0.5, 1.5]$ for all four data sets except Amazon. While the smallest misclassification error is obtained outside this region for the Amazon data set (with $\mu_1$ taking a smaller value), the RMS
errors obtained by choosing \( \mu_1 \) and \( \mu_2 \) within this region are quite close to the optimal value. We also observe that the algorithm error does not vary dramatically within this region. In particular, the relative difference in the misclassification error varies between 1% - 4% in all data sets. Upper table: Misclassification rate (in percentage). Lower table: RMS error conditions, hence the intrinsic dimension of the data set is higher than that of COIL-20. As a result, the optimal value of the \( K \) parameter is seen to be larger for MIT-CBCL in Figure 19(b).

**Sensitivity to the number of eigenvectors \( R \).** Finally, we investigate how the choice of the number of graph basis vectors \( R \) used in the objective (15) affects the algorithm performance. The variation of the target misclassification rate is plotted with respect to the number of basis vectors \( R \) in Figure 20 for all data sets. The results suggest that the variation of the misclassification rate with \( R \) has similar characteristics among different data sets. At small \( R \) values, the classification performance improves as \( R \) increases, since the label function can be approximated more accurately when more basis vectors are used. The optimal value of \( R \) is often around 9-12, 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.

### 6 Conclusion

We have considered the problem of domain adaptation on graphs. Given a source graph with sufficiently many labeled nodes and a target graph, we have proposed a graph-based domain adaptation algorithm that 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 the target graphs. The learnt bases not only resemble in terms of their spectral content, but also “align” the two graphs such that the label functions on 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 performance of the proposed method is demonstrated mainly in data classification applications; however, it can potentially be applied to a wide range of machine learning problems concerning the inference of the unknown values of a graph function from available values. The exploration of information transfer based on more elaborate graph kernels than the graph Fourier basis, or the extension of the method to explicitly employ data embeddings in addition to graph models in order to improve its performance on data sets with available ambient space representations remain as future directions.

**Appendix A**

**Proof of Proposition 3**

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

\[
\begin{align*}
(f^s)^T L^s f^s &= (\overline{\pi}^s)^T (\overline{U}^T)^T L^s \overline{U}^T \overline{\pi}^s = (\overline{\pi}^s)^T \Lambda^s \overline{\pi}^s \\
(f^t)^T L^t f^t &= (T \overline{\pi}^t)^T (\overline{U}^T)^T L^t \overline{U}^T T \overline{\pi}^t = (T \overline{\pi}^t)^T \Lambda^t T \overline{\pi}^t
\end{align*}
\]

where $\Lambda^s$ and $\Lambda^t$ are the diagonal matrices consisting of the $R$ smallest eigenvalues of respectively $L^s$ and $L^t$, such that $\lambda_{ii}^s = \lambda_i^s$ and $\lambda_{ii}^t = \lambda_i^t$, for $i = 1, \ldots, R$.

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

\[
\begin{align*}
|(f^s)^T L^s f^s - (f^t)^T L^t f^t| &= |(\overline{\pi}^s)^T \Lambda^s \overline{\pi}^s - (T \overline{\pi}^t)^T \Lambda^t T \overline{\pi}^t| \\
&= |(\overline{\pi}^s)^T \Lambda^s \overline{\pi}^s - (\overline{\pi}^s)^T \Lambda^s \overline{\pi}^s + (\overline{\pi}^s)^T \Lambda^t \overline{\pi}^t| \\
&- |(\overline{\pi}^t)^T \Lambda^t \overline{\pi}^t + (\overline{\pi}^t)^T \Lambda^t \overline{\pi}^t - (T \overline{\pi}^t)^T T \overline{\pi}^t| \\
&\leq |(\overline{\pi}^s)^T (\Lambda^s - \Lambda^t) \overline{\pi}^s| + |(\overline{\pi}^s)^T \Lambda^t \overline{\pi}^t - (\overline{\pi}^t)^T \Lambda^t \overline{\pi}^t| \\
&+ |(\overline{\pi}^t)^T \Lambda^t \overline{\pi}^t - (T \overline{\pi}^t)^T T \overline{\pi}^t|.
\end{align*}
\]

In the following, we derive an upper bound for each one of the three terms at the right hand side of the inequality in (34). The first term is bounded as

\[
|\pi^s|^T (\Lambda^s - \Lambda^t) \pi^s | \leq \|\pi^s\|^2 \|\Lambda^s - \Lambda^t\| \leq C^2 \delta.
\]

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 is upper bounded by $\delta$ due to the assumption $|\lambda_i^s - \lambda_i^t| \leq \delta$ for all $i$.

Next, we bound the second term in (34) as

\[
\begin{align*}
|\pi^s|^T \Lambda^t \pi^s - (\pi^t)^T \Lambda^t \pi^t | &= |\pi^s|^T \Lambda^t \pi^s - (\overline{\pi}^s)^T \Lambda^t \overline{\pi}^s + (\overline{\pi}^s)^T \Lambda^t \overline{\pi}^t - (\overline{\pi}^t)^T \Lambda^t \overline{\pi}^t| \\
&\leq |\pi^s|^T \Lambda^t (\pi^s - \pi^t) + |(\overline{\pi}^s)^T \Lambda^t \overline{\pi}^t| \\
&\leq \|\pi^s\| \|\Lambda^t\| \|\pi^s - \pi^t\| + \|\overline{\pi}^s\| \|\Lambda^t\| \|\overline{\pi}^t\| \leq 2C\lambda_R \Delta_R
\end{align*}
\]

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

Lastly, the third term in (34) can be bounded as

\[
\begin{align*}
|\pi^t|^T \Lambda^t \pi^t - (T \overline{\pi}^t)^T T \Lambda^t T \overline{\pi}^t | &\leq |\pi^t|^T \Lambda^t \pi^t - (\overline{\pi}^t)^T \Lambda^t \overline{\pi}^t + (\overline{\pi}^t)^T \Lambda^t T \overline{\pi}^t| \\
&\leq |(\pi^t)^T \Lambda^t (\pi^t - T \overline{\pi}^t) + |(\overline{\pi}^t)^T \Lambda^t T \overline{\pi}^t| \\
&\leq \|\pi^t\|^2 \|\Lambda^t\| \|\pi^t - T \overline{\pi}^t\| + \|\pi^t\| \|\Lambda^t\| \|T \overline{\pi}^t\|.
\end{align*}
\]

Bounding the norm of $T$ as

\[
\|T\| = \|I + T - I\| \leq \|I\| + \|T - I\| \leq 1 + \Delta_T
\]

and using also the assumption $\|T - I\| \leq \Delta_T$ in (37), we get

\[
|\pi^t|^T \Lambda^t \pi^t - (T \overline{\pi}^t)^T T \Lambda^t T \overline{\pi}^t | \leq C^2 \lambda_R \Delta_T + C^2 \lambda_R \Delta_T (1 + \Delta_T).
\]

Finally, putting together the upper bounds for all the three terms in (34), we get the stated result

\[
|(f^s)^T L^s f^s - (f^t)^T L^t f^t| \leq C^2 \delta + 2C\lambda_R \Delta_R + C^2 \lambda_R (2\Delta_T + \Delta_T^2).
\]
APPENDIX B

CONVERGENCE ANALYSIS OF THE SQP METHOD

In order to analyze the convergence of the SQP algorithm, we first inspect the Hessian $\nabla_\eta^2 F(\tilde{\eta}, \eta)$ of the Lagrangian function. Recall that the matrices $2\mu_2 F^T F$ and $\nabla_\eta^2 g(\tilde{\eta}, \eta)$ in the expression of $\nabla_\eta^2 F(\tilde{\eta}, \eta)$ are both diagonal. From the assumption that $\mu_2 > \eta_j^*$, we get that at the local solution $(\tilde{\eta}, \eta^*)$ the entries of the diagonal matrix $2\mu_2 F^T F - \nabla_\eta^2 g(\tilde{\eta}, \eta)$ are lower bounded as

$$
2\mu_2 F^T F - \nabla_\eta^2 g(\tilde{\eta}, \eta)_{kk} = 2\mu_2 \Lambda_{kk} - 2\eta_j^* > 2\mu_2 - 2\eta_j^* > 0
$$

for $k = R(j - 1) + i$ with $i, j = 1, \ldots, R$, where the first inequality simply follows from the fact that $\Lambda_{kk} > 1$ as the definition (13) implies. This shows that the diagonal matrix $2\mu_2 F^T F - \nabla_\eta^2 g(\tilde{\eta}, \eta)$ is positive definite at the local solution $(\tilde{\eta}, \eta^*)$. Then, the Hessian $\nabla_\eta^2 F(\tilde{\eta}, \eta)$ of the Lagrangian is also positive definite at $(\tilde{\eta}, \eta^*)$, since for any $x \in \mathbb{R}^{2R}$ with $x \neq 0$ we have

$$
F^T A F - \nabla_\eta^2 g(\tilde{\eta}, \eta)_{kk} = 2\mu_2 \Lambda_{kk} - 2\eta_j^* > 2\mu_2 - 2\eta_j^* > 0
$$

which follows from (41) and the positive semi-definiteness of $A^T A$.

Next, from the form of the constraint gradients $\nabla_\eta g_j$ in (28), and the constraint that the columns of $\bar{T}$ must have unit norm at a local solution, it is easy to observe that the set of the equality constraint gradients $\{\nabla_\eta g_j, j = 1, \ldots, R\}$ is linearly independent, which are the only active constraints of the optimization problem. Since the objective and the constraint functions are twice differentiable with Lipschitz continuous second derivatives, the active constraint gradients are linearly independent, and the Hessian of the Lagrangian function is positive definite at the local solution $(\tilde{\eta}, \eta^*)$, by [60, Theorem 18.4], the SQP algorithm converges to the local solution $(\tilde{\eta}, \eta^*)$ provided that the initial solution $(\tilde{\eta}, \eta)$ is sufficiently close to $(\tilde{\eta}, \eta^*)$.

ACKNOWLEDGMENTS

This work has been supported by the TÜBİTAK 2232 research scholarship.

REFERENCES

[1] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, 2010.
[2] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, “Unsupervised visual domain adaptation using subspace alignment,” in Proc. IEEE International Conference on Computer Vision, ser. ICCV ’13, 2013, pp. 2961–2967.
[3] B. Gong, Y. Shi, F. Sha, and K. Grauman, “Geodesic flow kernel for unsupervised domain adaptation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2066–2073.
[4] D. López-Paz, J. M. Hernández-Lobato, and B. Schölkopf, “Semi-supervised domain adaptation with non-parametric copulas,” in Proc. Advances in Neural Information Processing Systems 25, 2012, pp. 674–682.
[5] B. Sun, J. Feng, and K. Saenko, “Return of frustratingly easy domain adaptation,” in Proc. Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 2058–2065.
[6] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in Proc. 32nd International Conference on Machine Learning, 2015, pp. 1180–1189.
[7] S. A. Myers and J. Leskovec, “On the convexity of latent social network inference,” in Advances in Neural Information Processing Systems 23, 2010, pp. 1741–1749.
[8] I. Jablonski, “Graph signal processing in applications to sensor networks, smart grids, and smart cities,” IEEE Sensors Journal, vol. 17, no. 23, pp. 7679–7666, 2017.
[9] D. K. Hammond, P. Vandergeyten, and R. Grisonval, “Wavelets on graphs via spectral graph theory,” Applied and Computational Harmonic Analysis, vol. 30, no. 2, pp. 129–150, 2011.
[10] 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., vol. 30, no. 3, pp. 83–98, 2013.
[11] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” Neural Computation, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.
[12] 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, 2003, pp. 321–328.
[13] X. Zhu, Z. Ghahramani, and J. D. Lafferty, “Semi-supervised learning using Gaussian fields and harmonic functions,” in Proc. 20th Int. Conf. Machine Learning, 2003, pp. 912–919.
[14] X. Zhu, “Semi-supervised learning literature survey,” Comput Sci, University of Wisconsin-Madison, vol. 2, 07 2008.
[15] M. Belkin and P. Niyogi, “Semi-supervised learning on Riemannian manifolds,” Machine Learning, vol. 56, no. 1-3, pp. 209–239, 2004.
[16] A. S. Georgiades, 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, vol. 23, no. 6, pp. 643–660, 2001.
[17] 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, 2006, pp. 601–608.
[18] W. Dai, G. Xie, Q. Yang, and Y. Yu, “Transferring naive Bayes classifiers for text classification,” in Proc. Twenty-Second AAAI Conference on Artificial Intelligence, 2007, pp. 540–545.
[19] J. Blitzer, R. T. McDonald, and F. Pereira, “Domain adaptation with structural correspondence learning,” in Proc. Conf. Empirical Methods in Natural Language Processing, 2006, pp. 120–128.
[20] 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, 2011, pp. 505–513.
[21] H. Daumé III, “Frustratingly easy domain adaptation,” in Annual Meeting-Association for Computational Linguistics, 2007.
[22] H. Daumé III, A. Kumar, and A. Saha, “Co-regularization based semi-supervised domain adaptation,” in Proc. Advances in Neural Information Processing Systems 23, 2010, pp. 478–486.
[23] 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, 2010, pp. 53–59.
[24] L. Duan, D. Xu, and I. W. Tsang, “Learning with augmented features for heterogeneous domain adaptation,” in Proc. 29th International Conference on Machine Learning, 2012.
[25] K. Crammer, M. Kearns, and J. Wortman, “Learning from multiple sources,” Journal of Machine Learning Research, vol. 9, pp. 1757–1774, 2008.
[26] Q. Wu, H. Wu, X. Zhou, M. Tan, Y. Xu, Y. Yan, and T. Hao, “Online transfer learning with multiple homogeneous or heterogeneous sources,” IEEE Trans. Knowl. Data Eng., vol. 29, no. 7, pp. 1494–1507, 2017.
[27] S. J. Pan, J. T. Kwok, and Q. Yang, “Transfer learning via dimensionality reduction,” in Proc. Twenty-Third AAAI Conference on Artificial Intelligence, 2008, pp. 677–682.
[28] Z. Fang and Z. Zhang, “Discriminative transfer learning on manifold,” in Proc. 13th SIAM Int. Conf. Data Mining, 2013, pp. 539–547.
[29] C. Wang and S. Mahadevan, “Manifold alignment using proteomes analysis,” in Proc. 25th Int. Conf. Machine Learning, 2008, pp. 1120–1127.
[30] C. Wang, “A geometric framework for transfer learning using manifold alignment,” Ph.D. dissertation, 2010.
[31] J. Zhang, W. Li, and P. Ogunbona, “Joint geometrical and statistical alignment for visual domain adaptation,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5105–5108.

[32] C. Wang and S. Mahadevan, “Heterogeneous domain adaptation using manifold alignment,” in Proc. 22nd Int. Joint Conf. on Artificial Intelligence, 2011, pp. 1541–1546.

[33] K. Muandet, D. Balduzzi, and B. Schölkopf, “Domain generalization via invariant feature representation,” in Proc. 30th International Conference on Machine Learning, 2013, pp. 10–18.

[34] X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu, “Transfer learning on heterogeneous feature spaces via spectral transformation,” in 10th IEEE International Conference on Data Mining, 2010, pp. 1049–1054.

[35] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, “Optimal transport for domain adaptation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 9, pp. 1853–1865, 2017.

[36] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, “Domain adaptation with conditional transferable components,” in Proc. 35nd International Conference on Machine Learning, 2016, pp. 2839–2848.

[37] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, “Unsupervised domain adaptation by domain invariant projection,” in IEEE International Conference on Computer Vision, 2013, pp. 769–776.

[38] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” IEEE Trans. Neural Networks, vol. 22, no. 2, pp. 199–210, 2011.

[39] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, “Adaptation regularization: A general framework for transfer learning,” IEEE Trans. Knowl. Data Eng., vol. 26, no. 5, pp. 1076–1089, 2014.

[40] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, “Scatter component analysis: A unified framework for domain adaptation and domain generalization,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 7, pp. 1414–1430, 2017.

[41] C. D. Tran, O. O. Rudovic, and V. Pavlovic, “Unsupervised domain adaptation with copula models,” in 27th IEEE Int. Workshop on Machine Learning for Signal Processing, 2017, pp. 1–6.

[42] Y. Xu, S. J. Pan, H. Xiong, Q. Wu, R. Luo, H. Min, and H. Song, “A unified framework for metric transfer learning,” IEEE Trans. Knowl. Data Eng., vol. 29, no. 6, pp. 1158–1171, 2017.

[43] S. Herath, M. T. Harandi, and F. Porikli, “Learning an invariant Hilbert space for domain adaptation,” arXiv preprint, 2016.

[44] 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, 2015, pp. 2142–2150.

[45] L. Cheng and S. J. Pan, “Semi-supervised domain adaptation on manifolds,” IEEE Trans. Neural Netw. Learning Syst., vol. 25, no. 12, pp. 2240–2249, 2014.

[46] H. Lu, C. Shen, Z. Cao, Y. Xiao, and A. van den Hengel, “An embarrassingly simple approach to visual domain adaptation,” IEEE Trans. Image Processing, vol. 27, no. 7, pp. 3403–3417, 2018.

[47] M. Long, Y. Cao, Y. Wang, and M. I. Jordan, “Learning transferrable features with deep adaptation networks,” in Proc. 32nd International Conference on Machine Learning, 2015, pp. 97–105.

[48] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2962–2971.

[49] M. Long, H. Zhu, J. Wang, and M. I. Jordan, “Deep transfer learning with joint adaptation networks,” in Proc. 34th International Conference on Machine Learning, 2017, pp. 2208–2217.

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

[51] A. Singer, “From graph to manifold laplacian: The convergence rate,” Applied and Computational Harmonic Analysis, vol. 21, no. 1, pp. 128 – 134, 2006, special Issue: Diffusion Maps and Wavelets.

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

[53] D. Eynard, A. Kovnatsky, M. M. Bronstein, K. Glashoff, and A. M. Bronstein, “Multimodal manifold analysis by simultaneous diagonalization of Laplacians,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 12, pp. 2505–2517, 2015.

[54] J. Pokras, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro, “Sparse modeling of intrinsic correspondences,” Comput. Graph. Forum, vol. 32, no. 2, pp. 459–468, 2013.

[55] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers, “Partial functional correspondence,” Comput. Graph. Forum, vol. 36, no. 1, pp. 222–226, 2017.

[56] M. Xiao and Y. Guo, “Feature space independent semi-supervised domain adaptation via kernel matching,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 1, pp. 54–66, 2015.

[57] P. S. Dhillon, P. P. Talukdar, and K. Crammer, “Metric learning for graph-based domain adaptation,” in Proc. 24th International Conference on Computational Linguistics, 2012, pp. 255–264.

[58] B. Baran, F. Bovolo, A. Bhattacharya, L. Bruzzone, S. Chaudhuri, and K. M. Mediratuar, “A novel graph-matching-based approach for domain adaptation in classification of remote sensing image pair,” IEEE Trans. Geoscience and Remote Sensing, vol. 53, no. 7, pp. 4045–4062, 2015.

[59] M. Pilanci and E. Vural, “Domain adaptation via transferring spectral properties of labels functions on graphs,” in IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop, 2016, pp. 1–5.

[60] J. Nocedal and S. I. Wright, Numerical Optimization, 2nd ed. New York, NY, USA: Springer, 2006.

[61] “MIT-CBCL face recognition database,” available: http://cbl.mit.edu/software-datasets/heisele/facerecognition-database.html.

[62] S. A. Nene, S. K. Nayar, and H. Murase, “Columbia Object Image Library (COIL-20),” Tech. Rep., Feb 1996.

[63] 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, 2015, pp. 785–794.

[64] J. J. McAuley and J. Leskovec, “Learning to discover social circles in ego networks,” in Advances in Neural Information Processing Systems, 2012, pp. 548–556.

[65] D. J. Watts and S. H. Strogatz, “Collective dynamics of "small-world" networks,” Nature, vol. 393, pp. 440–442, 1998.

[66] R. Gupta, K. R. Laghari, and T. H. Falk, “Relevance vector classifier decision fusion and EEG graph-theoretic features for automatic affective state characterization,” Neurocomputing, vol. 174, pp. 875–884, 2016.

[67] W. S. Torgerson, “Multidimensional scaling: I. theory and method,” Psychometrika, vol. 17, no. 4, pp. 401–419, Dec 1952.