DEEP REPRESENTATION LEARNING FOR CLUSTERING AND DOMAIN ADAPTATION

by

MOHSEN KHEIRANDISHFARD

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2019

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my supervisor Dr. Farhad Kamangar for his endless support, professional advice and immense knowledge during the course of my doctoral studies. I am deeply indebted to him for teaching me invaluable skills that are directly applicable to every aspects of my personal and professional life. I would also like to extend my gratitude to my co-supervisor Dr. Ramtin Madani for his ongoing support and help, to my committee members Dr. Vassilis Athitsos, Dr. Gergely Zaruba, and Dr. Manfred Huber for providing me insightful suggestions and valuable feedback, and to the graduate advisor of the department, Dr. Bahram Khalili, for his endless support and time.

I cannot be grateful enough to my family and friends, especially my mother who devoted her life to make me the person I am today. Most importantly, I would like to thank my wife, Fariba, whose love and constant encouragement helped me to complete this stage of my career.

December, 2019

ABSTRACT

DEEP REPRESENTATION LEARNING FOR CLUSTERING AND DOMAIN ADAPTATION

Mohsen Kheirandishfard, Ph.D.

The University of Texas at Arlington, 2019

Representation learning is a fundamental task in the area of machine learning which can significantly influence the performance of the algorithms used in various applications. The main goal of this task is to capture the relationships between the input data and learn feature representations that contain the most useful information of the original data. Such representations can be further leveraged in many machine learning applications such as clustering, natural language processing, recommender systems, etc. In this dissertation, we first present a theoretical framework for solving a broad class of non-convex optimization problems. The proposed method is applicable to various tasks involving representation learning such as discriminative dimensionality reduction and graph matching. We perform experiments on benchmark graph matching datasets to verify the effectiveness of our proposed approach in finding a near-optimal match between two given graphs. Besides that, we practically corroborate the capability of deep models in extracting complex underlying relationships between the data samples in two fundamental problems: subspace clustering and domain adaptation. Toward this goal, we propose two novel deep architectures for learning informative and high-quality representations that are able to considerably boost the performance of the existing algorithms. Our experiments demonstrate the potential of our proposed architectures in achieving state-of-the-art results on well-known benchmark

datasets. In the following, we give brief explanations about three different projects included in this dissertation.

Convex Relaxation of Bilinear Matrix Inequalities: Many interesting machine learning problems are inherently non-convex and computationally hard to solve. We study the applicability of convex relaxation techniques on finding solutions to these problems. We develop a novel and computationally efficient convexification technique that relies on convex quadratic constraints to transform a class of non-convex problems, known as bilinear matrix inequality (BMI), into convex surrogates. Then, the solution of the surrogates can be efficiently obtained using standard convex optimization approaches. We study the theoretical aspects of the proposed convexification algorithm and investigate the conditions under which the algorithm is guaranteed to produce feasible solutions for the BMI problem. As the BMI formulation encompasses a wide range of non-convex problems, the proposed algorithm is generally applicable to many machine learning problems such as dimensionality reduction, minimum volume ellipsoid, graph matching, and matrix completion, etc. To evaluate the effectiveness of the proposed procedure, we use the idea of sequential relaxation to find the solution of the graph matching problem. To this end, we first propose a novel convex formulation for the problem and then develop a numerical algorithm based on the alternating direction method of multipliers to solve the convexified formulation. The results of our experiments on two benchmark datasets for graph matching demonstrate the potential of the proposed algorithm in finding high-quality solutions.

Multi-Level Representation Learning for Deep Subspace Clustering: Subspace clustering is an unsupervised learning task with a variety of machine learning applications such as motion segmentation, face clustering, etc. The primary goal of this task is to partition a set of data samples, drawn from a union of low-dimensional subspaces, into disjoint clusters such that the samples within each cluster belong to the same subspace. This project proposes a novel deep subspace clustering approach which uses convolutional autoencoders

to transform input images into new representations lying on a union of linear subspaces. The first contribution of our method is to insert multiple fully-connected linear layers between the encoder layers and their corresponding decoder layers to promote learning more favorable representations for subspace clustering. These connection layers facilitate the feature learning procedure by combining low-level and high-level information for generating multiple sets of self-expressive and informative representations at different levels of the encoder. Moreover, we introduce a novel loss minimization model which leverages an initial clustering of the samples to effectively fuse the multi-level representations and recover the underlying subspaces more accurately. The loss function is then minimized through an iterative scheme which alternatively updates the network parameters and produces a new clustering of the samples until the convergence is obtained. Our experiments on four real-world datasets demonstrate that the proposed approach exhibits superior performance compared to the state-of-the-art methods on most of the subspace clustering problems.

Class Conditional Alignment for Partial Domain Adaptation: Adversarial adaptation models have demonstrated significant progress towards transferring knowledge from a labeled source dataset to an unlabeled target dataset. Partial domain adaptation (PDA) investigates the scenarios in which the source domain is large and diverse, and the target label space is a subset of the source label space. The main purpose of PDA is to identify the shared classes between the domains and promote learning transferable knowledge from these classes. In this project, we propose a multi-class adversarial architecture for PDA. The proposed approach jointly aligns the marginal and class-conditional distributions in the shared label space by minimaxing a novel multi-class adversarial loss function. Furthermore, we incorporate effective regularization terms to encourage selecting the most relevant subset of source domain classes. In the absence of target labels, the proposed approach is able to effectively learn domain-invariant feature representations, which in turn can enhance the classification performance in the target domain. Our comprehensive experiments on two

benchmark datasets Office-31 and Office-Home corroborate the effectiveness of the proposed approach in addressing different partial transfer learning tasks.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

Introduction

Many problems arising in real-world can be cast as optimization problems involving various variables and constraints. Generally, an optimization problem aims to minimize a cost function, called objective function, over a set of points that satisfy certain constraints. Such problem can be formulated as

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \qquad f(\boldsymbol{x}) \tag{1.1a}$$

$$\text{subject to} \qquad \boldsymbol{x} \in \mathcal{C}, \tag{1.1b}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of variables, $f : \mathbb{R}^n \to \mathbb{R}$ denotes the objective function, and $\mathcal{C}$ shows the set of points satisfying all constraints, named feasible set. The hardness of solving (1.1a) – (1.1b) depends on the properties of function f and set $\mathcal{C}$ (e.g. convexity or non-convexity).

This dissertation explores a wide range of optimization problems that frequently arise in different areas of machine learning. The first project provides a theoretical framework for studying a general class of non-convex optimization problems and proposes a practical algorithm to solve them in an efficient manner. We further demonstrate the applicability of the proposed algorithm in some real-world applications. The second project is related to the subspace clustering problem which is a fundamental problem in the area of machine learning. In this project, we use deep learning architectures to transform input samples into the feature representations that lie on a union of linear subspaces and then apply the spectral clustering technique to recover the clusters. The third project investigates the domain adaptation and transfer leaning problems. This project aims to transfer knowledge from a

large and diverse dataset into a small one by leveraging deep models for learning feature representations that are domain-invariant. In what follows, we briefly review each of these projects separately.

1.1 Convex Relaxation of Bilinear Matrix Inequalities

Optimization problems involving matrix inequality constraints widely arise in both theoretical and practical aspects of machine learning problems [6, 7, 8, 9, 10]. The well-known linear matrix inequality (LMI) problem is regarded as a popular and special case of these problems which can be efficiently solved by means of classical convex optimization methods. As an important generalization of LMIs, bilinear matrix inequalities (BMIs) have diverse applications, but are computationally non-convex and computationally prohibitive to solve [11, 12]. A general BMI problem can be expressed as the following optimization problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{x}) \tag{1.2a}$$

$$\text{subject to} \quad \boldsymbol{F}_0 + \sum_{k=1}^{n} x_k \boldsymbol{K}_k + \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \boldsymbol{L}_{ij} \preceq 0, \tag{1.2b}$$

where $\preceq$ denotes the negative semidefinite symbol and $\boldsymbol{F}_0$, $\{\boldsymbol{K}_k\}_{k=1}^n$, and $\{\boldsymbol{L}_{ij}\}_{i,j=1}^n$ are given symmetric matrices of size $m \times m$. Various approaches are presented in the literature to find the solution of problem (1.2a) – (1.2b). One simple and commonly-used technique is alternating method (AM) which partitions the non-convex problem into two (or more) convex sub-problems and then alternatively solves the sub-problems until the convergence is obtained. Although the AM-based approaches enjoy low per-iteration complexity and work well in practice, they mostly offer no convergence guarantees to a feasible point of the original non-convex problem.

One promising direction is to use relaxation techniques which are well-studied in many areas such as machine learning [13, 14], polynomial optimization [15, 16], etc. The

basic idea of these techniques is to first relax the non-convex problems into convex surrogates whose solution approximate the solution of the original non-convex problems. Then, the solution of the convexified problems can be efficiently obtained in polynomial time using standard convex optimization methods such as alternating direction method of multipliers (ADMM) [17], interior-point method [18, 19, 20] (see [21] for a detailed survey).

This dissertation proposes a novel and general relaxation technique which only relies on convex quadratic constraints to find the solution of problems involving BMI constraints. To ensure the relaxation provides feasible solutions for the original problem, we use an initial point to design a penalty term which is incorporated into the objective function of the relaxation. We theoretically prove that if the initial point is feasible for the BMI problem, the penalized relaxation is guaranteed to maintain the feasibility of the solutions. In the case where the initial point is not feasible, we introduce certain conditions under which the solution of the convexified problem is guaranteed to be feasible. To further improve the quality of the solution, we extend the proposed relaxation to a sequential scheme which starts from an initial point and aims to recover feasible and near-globally optimal solutions of the original non-convex problem. The proposed scheme can be widely used in the area of machine learning as many problems can be seen as special cases of BMI problems. We investigate the applicability of the proposed scheme on some fundamental problems such as dimensionality reduction, minimum volume ellipsoid, and graph matching. To see how the idea performs in real-world applications, we introduce a convexified formulation of the graph matching problem and develop a sequential numerical scheme based on alternating direction method of multipliers to find the solution of the convexified problem. Experiments on two benchmark datasets for the graph matching problem verify the effectiveness of our proposed approach in solving the graph matching problem.

1.2 Multi-Level Representation Learning for Deep Subspace Clustering

High-dimensional data are becoming increasingly common and available in many real-world machine learning applications and researchers constantly endeavor to develop efficient and fast algorithms to process this huge amount of data in a short period of time. It is widely believed that the high-dimensional data points are not uniformly distributed in the ambient space and they mostly lie on or close to low-dimensional structures. Therefore, this is of great importance to recover such low-dimensional structures as it can significantly reduce the computational complexity and the memory usage of the algorithms. Moreover, it allows learning new representations of the points that are more robust to noise than the original high-dimensional points and hence can improve the performance of the existing machine learning algorithms.

Subspace clustering is an unsupervised learning task with a variety of machine learning applications such as, motion segmentation [22, 23], face clustering [24, 25], and movie recommendation [26, 27]. The main purpose of subspace clustering is to partition a bunch of given samples, drawn from a union of low-dimensional subspaces, into disjoint clusters such that the samples within each cluster belong to the same subspace [28, 29]. In this sense, subspace clustering differs from the standard clustering problems as it assumes the samples are arbitrarily distributed on the subspaces, not only around some certain centroids.

To find the solution of subspace clustering problem, various approaches are proposed in the literature such as algebraic methods, iterative methods, and spectral clustering-based methods [30, 31, 32, 33, 34]. This dissertation mainly focuses on the spectral clustering-based approaches. The basic idea behind these methods is to first use the entire input samples to learn an affinity matrix and then apply spectral clustering technique on the matrix to infer the underlying subspaces and cluster the samples. One well-established method of this kind is sparse subspace clustering (SSC) [34] which relies on the concept of self-expressiveness property. This property states that each sample point in a union of

subspaces is efficiently expressible in terms of a linear (or affine) combination of other points in the subspaces [34]. Ideally, it is expected that the nonzero coefficients in the linear representation of each sample correspond to the points of the same subspace as the given sample. Towards this goal, SSC algorithm introduces an ℓ_1-regularized model to select only a small subset of points belonging to the same subspace for reconstructing each data point. Define matrix $\boldsymbol{X} = [\boldsymbol{x}_1|\boldsymbol{x}_2|\dots|\boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$ where $\{\boldsymbol{x}_i\}_{i=1}^n$ are n sample points drawn from a union of linear subspaces. The SSC algorithm proposes to solve the following optimization problem to find the coefficient vector $\boldsymbol{c}_i \in \mathbb{R}^n$ for reconstruction of sample point $\boldsymbol{x}_i$ based on the other points

$$\underset{\boldsymbol{c}_i\in\mathbb{R}^n}{\text{minimize}} \qquad \|\boldsymbol{c}_i\|_1 \tag{1.3a}$$

$$\text{subject to} \qquad \boldsymbol{x}_i = \boldsymbol{X}\boldsymbol{c}_i, \qquad c_{ii} = 0, \tag{1.3b}$$

where constraint $c_{ii} = 0$ ensures that $\boldsymbol{x}_i$ is not reconstructed by itself. By considering a more general case in which the samples are contaminated by noise $\boldsymbol{X} = \boldsymbol{X}\boldsymbol{C} + \boldsymbol{E}$, we can formulate the following optimization problem for the entire samples

$$\underset{\boldsymbol{C}\in\mathbb{R}^{n\times n},\boldsymbol{E}\in\mathbb{R}^{d\times n}}{\text{minimize}} \qquad \|\boldsymbol{E}\|_{\text{F}} + \|\boldsymbol{C}\|_1 \tag{1.4a}$$

$$\text{subject to} \qquad \boldsymbol{X} = \boldsymbol{X}\boldsymbol{C} + \boldsymbol{E}, \qquad \text{diag}(\boldsymbol{C}) = \boldsymbol{0}. \tag{1.4b}$$

Once optimal matrix $\boldsymbol{C}$ is obtained, a symmetric affinity matrix can be obtained via $\boldsymbol{W} = \frac{|\boldsymbol{C}|+|\boldsymbol{C}^\top|}{2}$. Applying spectral clustering technique on matrix $\boldsymbol{W}$ allows to cluster the samples into their respective clusters and recover the underlying subspaces.

Despite the key role that the self-expressiveness plays in spectral clustering-based methods, it may not be satisfied in a wide range of applications in which samples lie on non-linear subspaces, e.g. face images taken under non-uniform illumination and at different poses [35]. A common practice technique to handle these cases is to leverage well-known kernel trick to implicitly map the samples into a higher dimensional space so that they

better conform to linear subspaces [36, 37, 38, 39]. Despite the empirical success obtained by this strategy, it is not widely applicable to various applications, mainly because it is quite difficult to identify an appropriate kernel function for a given set of data points [40].

Recently, deep neural networks have exhibited exceptional ability in capturing complex underlying structures of the input data and learning discriminative features for clustering. Inspired by that, the researchers have established a new line of research to bridge deep learning and subspace clustering for developing deep subspace clustering approaches [35, 41, 42, 43]. Variational Autoencoders (VAE) [44, 45] and Generative Adversarial Network (GAN) [46] are among the most popular deep architectures adopted by these methods to produce feature representations suitable for subspace clustering [45]. Compared to the conventional subspace clustering approaches, deep methods can better exploit the non-linear relationships between the sample points and consequently they achieve higher performance, especially in those applications in which the samples do not necessarily satisfy the self-expressiveness property [35].

This dissertation proposes a novel spectral clustering-based approach which uses stacked convolutional autoencoders to tackle the problem of subspace clustering. Inspired by the idea of residual networks, our first contribution is to add multiple fully-connected linear layers between the corresponding layers of the encoder and decoder to infer multi-level representations from the output of every encoder layer. These connection layers enable to produce representations which are enforced to satisfy self-expressiveness property and hence well-suited to subspace clustering. We model each connection layer as a self-expression matrix created from the summation of a coefficient matrix shared between all layers and a layer-specific matrix that captures the unique knowledge of each individual layer. Moreover, we introduce a novel loss function that utilizes an initial clustering of the samples and efficiently aggregates the information at different levels to infer the coefficient matrix and the layer-specific matrices more accurately. This loss function is further

minimized in an iterative scheme which alternatively updates the network parameters for learning better subspace clustering representations and produces a new clustering of the samples. We perform extensive experiments on four benchmark datasets for subspace clustering, including two face image and two object image datasets, to evaluate the efficacy of the proposed method. The experiments demonstrate that our approach can efficiently handle clustering the data from non-linear subspaces and it performs better than the state-of-the-art methods on most of the subspace clustering problems.

1.3 Class Conditional Alignment for Partial Domain Adaptation

With the impressive power of learning representations, deep neural networks have shown superior performance in a wide variety of machine learning tasks such as classification [47, 48, 49], semantic segmentation [50, 51, 52], object detection [53, 49, 54], etc. These notable achievements heavily depend on the availability of large amounts of labeled training data. However, in many applications, collecting sufficient labeled data is either difficult or time-consuming. One potential solution to reduce the labeling consumption is to build an effective predictive model using readily-available labeled data from a different but related source domain. Such a learning paradigm generally suffers from the distribution shift between the source and target domains, which in turn poses significant difficulties in adapting the predictive model to the target domain tasks.

In the absence of target labels, unsupervised domain adaptation (UDA) seeks to enhance the generalization capability of the predictive model by learning feature representations that are discriminative and domain-invariant [55, 56, 57]. Various approaches have been proposed in the literature to tackle the UDA problem by embedding domain adaptation modules in a deep architectures [58, 59, 60, 61, 62, 63] (see [64] for a comprehensive survey on deep domain adaptation methods). One well-established UDA approach is de-

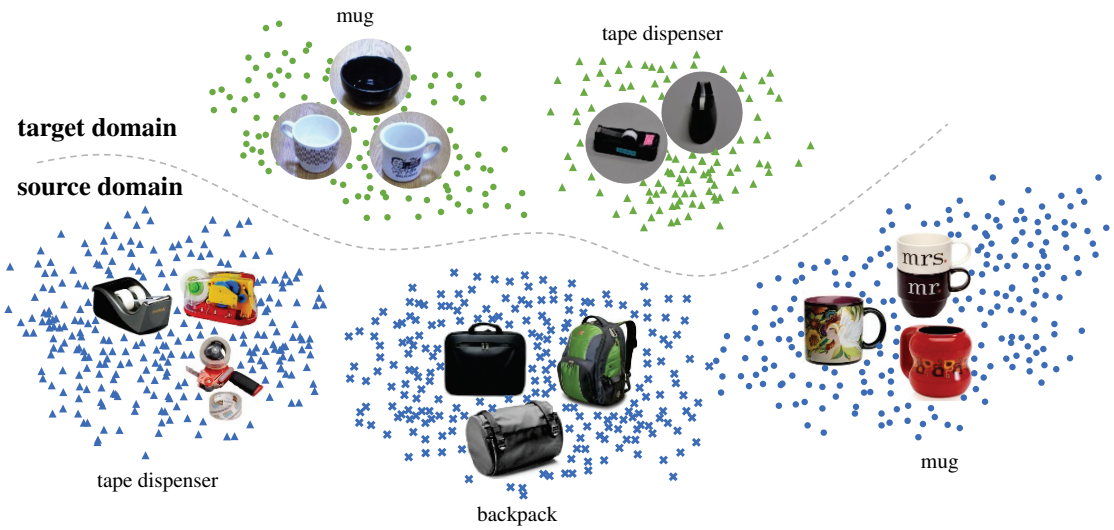

Figure 1.1: Partial domain adaptation scenario in which target label space ('tape dispenser', 'mug') is a subset of source label space ('tape dispenser', 'backpack', 'mug') [1]. The main difficulty in Partial domain adaptation scenario is to identify and reject the source domain classes that do not appear in the target domain ('backpack'), mainly because they may exert negative impacts on the overall transfer performance.

veloped in [46] which uses generative adversarial networks to adversarially learn domain-invariant feature representations that are indistinguishable for a discriminative domain classifier [65, 66, 67, 68, 69]. By adopting such a strategy, the marginal disparities between the source and target domains can be efficiently reduced, which results in significant improvement in the overall classification performance.

Despite the efficacy of the existing UDA methods, their superior performance is mostly limited to the scenarios in which the source and target domains share the same label space. With the goal of considering more realistic cases, [1] introduced partial domain adaptation (PDA) as a new adaptation scenario in which the target label space is a subset of the source label space. This scenario is clearly illustrated in Figure 1.1. The main challenge in PDA is to identify and reject the source domain classes that do not appear in the target domain, known as *outlier classes*, mainly because they may exert negative impacts on the overall transfer performance [4, 70]. Addressing this challenge enables the PDA methods

to transfer models trained on large and diverse labeled datasets (e.g. ImageNet) to small-scale datasets from different but related domains.

In this project, we propose a novel adversarial approach for partial domain adaptation which seeks to automatically reject the outlier source classes and improve the classification confidence on *irrelevant samples*, i.e. the samples that are highly dissimilar across the domains. The existing PDA methods often align the marginal distributions between the domains in the shared label space. Different from these methods, we propose a novel adversarial architecture that matches class-conditional feature distributions by minimaxing a multi-class adversarial loss function. Moreover, we propose to boost the target domain classification performance by incorporating two novel regularization functions. The first regularizer is a row-sparsity term on the output of the classifier to promote the selection of a small subset of classes that are in common between the source and target domains. The second one is a minimum entropy term which increases the classifier confidence level in predicting the labels of irrelevant samples from both domains. We empirically observe that our proposed approach considerably improves the state-of-the-art performance for various partial domain adaptation tasks on two commonly-used benchmark datasets Office-31 and Office-Home.

CHAPTER 2

Convex Relaxation of Bilinear Matrix Inequalities

In this chapter, we are concerned with the problem of minimizing a linear objective function subject to a bilinear matrix inequality (BMI) constraint. We first consider a family of convex relaxations which transform BMI optimization problems into polynomial-time solvable surrogates. As an alternative to the state-of-the-art semidefinite programming (SDP) and second-order cone programming (SOCP) relaxations, a computationally efficient *parabolic relaxation* is developed, which only relies on convex quadratic constraints. Then, we develop a family of penalty terms, that can be incorporated into the objective of SDP, SOCP, and parabolic relaxations to facilitate the recovery of feasible points for the original non-convex BMI optimization. Penalty terms can be constructed using any arbitrary initial point. We prove that the penalized relaxations are guaranteed to produce feasible points for the BMI problem if the initial point is sufficiently close to the feasible set of BMI. To further improve the quality of solutions, we generalize the penalized relaxation to a sequential scheme which starts from an arbitrary initial point (feasible or infeasible) and solves a sequence of penalized convex relaxations in order to find feasible and near-optimal solutions for the BMI optimization problems. We show that many non-convex machine learning problems can be cast as BMIs and consequently the proposed sequential algorithm is applicable to find the solution of these problems.

2.1 Introduction

A wide range of real-world problems in the different areas can be cast as optimization problems with matrix inequality constraints [6, 7, 71]. As a special case, the class of

problems with linear matrix inequalities (LMIs) can be solved efficiently up to any desired accuracy via interior-point based methods [20, 72]. However, despite various applications, optimization in the presence of bilinear matrix inequalities (BMIs) is computationally prohibitive and NP-hard in general. Significant efforts have been devoted to the development of algorithms for solving BMI problems [73, 74, 75], including software packages [76, 77, 78]. In [79, 80], alternating minimization (AM)-based algorithms are proposed which divide variables into two blocks that can be alternately optimized until convergence. Although AM-based methods enjoy simple implementation and perform satisfactorily in many cases, they offer no convergence guarantees to a feasible solution.

Another approach is to solve a sequence of convex relaxations until a satisfactory solution is obtained [81, 82, 83, 84, 85]. In [86, 82], BMI optimization problems are tackled by forming a sequence of semidefinite programming (SDP) relaxations. In [87], a sequential method is developed based on difference-of-convex programming with convergence guarantees to (sub)-optimal solutions. In [83, 84, 88] rank-constrained formulations with nuclear norm penalties are investigated along with bound-tightening methods for solving general BMI optimization problems. In [89, 90, 91, 92], branch-and-bound (BB) methods are developed with convergence guarantees to global optimality. The main shortcoming associated with BB methods is that they are often computationally prohibitive and thus their applicability is limited to moderate-sized problems. A novel global optimization method has been recently presented in [93] which tackles BMI problems using hybrid multi-objective optimization methods.

From a different viewpoint, BMIs can be categorized as a special case of polynomial matrix inequalities. Therefore, methods for solving general polynomial matrix inequalities are applicable to BMIs as well [94, 95]. Despite computational complexity for real-world applications, the most notable example is Lasserre's hierarchy of LMI relaxations [96], based on which several software packages have been developed [97, 98, 99].

The main contribution of this work is to introduce a novel and general convex relaxation, regarded as parabolic relaxation, for solving optimization problems with BMI constraints. The proposed convex relaxation relies on convex quadratic constraints as opposed to the SDP and SOCP relaxations that rely on computationally expensive conic constraints. Our second contribution is concerned with finding feasible and near-globally optimal solutions for BMI optimization problems. To this end, we incorporate a penalty term into the objective function of convex relaxations. The proposed penalty term is compatible with SDP, SOCP, and parabolic relaxations, and can be customized using any available initial point. We prove that If the initial point is feasible for the original problem, then the outcome of penalized relaxation is guaranteed to be feasible as well. Moreover, any infeasible initial point which is close to the feasible set is guaranteed to produce a feasible point. Built upon the above theoretical results, we offer a sequential penalized relaxation which is able to find feasible and near-globally optimal solutions for BMI optimization.

We show that many machine learning problems can be considered as special cases of BMI problems and hence the proposed algorithm can be utilized to find the solution of these problems.

2.1.1 Notation

Throughout the paper, the scalars, vectors, and matrices are respectively shown by italic letters, lower-case bold letters, and upper-case bold letters. Symbols $\mathbb{R}$, $\mathbb{R}^n$, and $\mathbb{R}^{n\times m}$ respectively denote the set of real scalars, real vectors of size n, and real matrices of size $n \times m$. The set of real $n \times n$ symmetric matrices and positive semidefinite matrices are shown with $\mathbb{S}_n$ and $\mathbb{S}_n^+$, respectively. For given vector $\boldsymbol{a}$ and matrix $\boldsymbol{A}$, symbols a_i and A_{ij} respectively indicate the i^{th} element of $\boldsymbol{a}$ and $(i,j)^{th}$ element of $\boldsymbol{A}$. Notations $[a]_{i\in\mathcal{I}}$ and $[A]_{ij\in\mathcal{I}}$ respectively shows the sub-vector and sub-matrix corresponding to the set of indices $\mathcal{I}$. Notation $\boldsymbol{A} \succeq 0$ means $\boldsymbol{A}$ is positive-semidefinite ($\boldsymbol{A} \succ 0$ indicates positive

definite) and $\boldsymbol{A} \preceq 0$ means $\boldsymbol{A}$ is negative-semidefinite ($\boldsymbol{A} \prec 0$ indicates negative definite). For two given matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of the same size, symbol $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \mathrm{tr}\{\boldsymbol{A}^\top \boldsymbol{B}\}$ shows the inner product between the matrices where $\mathrm{tr}\{.\}$ and $(.)^\top$ respectively denote the trace and transpose operators. Notation $\|.\|_p$ refers to either matrix norm or vector norm depending on the context and $|.|$ indicates the absolute value. Symbols $\boldsymbol{I}$, $\boldsymbol{e}_i$, and $\boldsymbol{0}$ denote the identity matrix, standard basis vector, and zero matrix of appropriate dimensions, respectively. Letters $\mathcal{N}$ and $\mathcal{M}$ are shorthand for sets $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$, respectively.

2.2 Problem Formulation

This paper is concerned with the following class of optimization problems with linear objective and a bilinear matrix inequality (BMI) constraint:

$$\underset{\boldsymbol{x} \in \mathbb{R}^n}{\text{minimize}} \qquad \boldsymbol{c}^\top \boldsymbol{x} \tag{2.1a}$$

$$\text{subject to} \qquad p(\boldsymbol{x}, \boldsymbol{x}\boldsymbol{x}^\top) \preceq 0, \tag{2.1b}$$

where $\boldsymbol{c} \in \mathbb{R}^n$ is the cost vector, and $p : \mathbb{R}^n \times \mathbb{S}_n \to \mathbb{S}_m$ is a linear matrix-valued function, which is regarded as matrix pencil. In general, p can be formulated as:

$$p(\boldsymbol{x}, \boldsymbol{X}) \triangleq \boldsymbol{F}_0 + \sum_{k \in \mathcal{N}} x_k \boldsymbol{K}_k + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} X_{ij} \boldsymbol{L}_{ij}. \tag{2.2}$$

where $\boldsymbol{F}_0$, $\{\boldsymbol{K}_k\}_{k \in \mathcal{N}}$, and $\{\boldsymbol{L}_{ij}\}_{i,j \in \mathcal{N}}$ are $m \times m$ real symmetric matrices. With no loss of generality, we can assume that $\boldsymbol{L}_{ij} = \boldsymbol{L}_{ji}$ for all $i, j \in \mathcal{N}$, since $\boldsymbol{X}$ is a symmetric matrix.

Problem (2.1a) – (2.1b) is non-convex and NP-hard in general, due to the presence of the BMI constraint (2.1b). To tackle this problem, it is common practice to solve convex surrogates that produce lower bounds on the globally-optimal cost of the original non-convex problem (2.1a) – (2.1b). To this end, an auxiliary matrix variable $\boldsymbol{X}$ is introduced

to account for $\boldsymbol{x}\boldsymbol{x}^\top$. This leads to the following *lifted reformulation* of the problem (2.1a) – (2.1b):

$$\underset{\boldsymbol{x}\in\mathbb{R}^n,\boldsymbol{X}\in\mathbb{S}_n}{\text{minimize}} \quad \boldsymbol{c}^\top\boldsymbol{x} \tag{2.3a}$$

$$\text{subject to} \quad p(\boldsymbol{x},\boldsymbol{X}) \preceq 0, \tag{2.3b}$$

$$\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top, \tag{2.3c}$$

where constraint (2.3c) is imposed to preserve the equivalency. Lifting casts the problem into a higher dimensional space in which the BMI constraint (2.1b) is transformed into a linear matrix inequality and the entire non-convexity is captured by the new constraint (2.3c). In what follows, we will substitute (2.3c) with convex alternatives and revise the objective function in order to obtain feasible and near-globally optimal points for the original problem (2.1a) – (2.1b).

2.3 Preliminaries

In order to further analyze the BMI constraint (2.1b), the next definition introduces the notion of pencil norm.

Definition 2.1 (Pencil Norm)**.** *For every $q \geq 1$, the q-norm of the matrix pencil in equation* (2.2) *is defined as*

$$\|p\|_q \triangleq \max\Big\{ \big\| \big[\boldsymbol{u}^\top \boldsymbol{L}_{ij} \boldsymbol{u}\big]_{i,j\in\mathcal{N}^2} \big\|_q \Big| \forall \boldsymbol{u} \in \mathbb{R}^m, \|\boldsymbol{u}\|_2 = 1 \Big\}. \tag{2.4}$$

The next definition provides a measure of the distance between any arbitrary point in $\mathbb{R}^n$ and the feasible set of optimization problem (2.1a) – (2.1b).

Definition 2.2 (Feasiblity Distance)**.** *For every $\boldsymbol{x} \in \mathbb{R}^n$, define the feasibility distance $d_{\mathcal{F}} : \mathbb{R}^n \to \mathbb{R}$ as*

$$d_{\mathcal{F}}(\boldsymbol{x}) \triangleq \inf\{ \|\boldsymbol{x} - \boldsymbol{a}\|_2 \;\big|\; \boldsymbol{a} \in \mathcal{F} \}, \tag{2.5}$$

where $\mathcal{F} \subseteq \mathbb{R}^n$ denotes the feasible set of the BMI problem (2.1a)–(2.1b). Observe that the feasibility distance is equal to 0 if $\boldsymbol{x} \in \mathcal{F}$.

We use the Mangasarian-Fromovitz constraint qualification (MFCQ) condition from [100] in order to characterize well-behaved feasible points of problem (2.1a)–(2.1b).

Definition 2.3 (MFCQ Condition)**.** *A feasible point $\boldsymbol{x} \in \mathcal{F}$ of problem (2.1a)–(2.1b) is said to satisfy the MFCQ condition if there exists $\boldsymbol{b} \in \mathbb{R}^n$ such that*

$$p(\boldsymbol{x}, \boldsymbol{x}\boldsymbol{x}^\top) + \sum_{k\in\mathcal{N}} b_k(\boldsymbol{K}_k + \delta_k(\boldsymbol{x})) \prec 0, \tag{2.6}$$

where for every $k \in \mathcal{N}$, the matrix function $\delta_k : \mathbb{R}^n \to \mathbb{S}_m$ is defined as

$$\delta_k(\boldsymbol{x}) \triangleq 2\sum_{i\in\mathcal{N}} x_i \boldsymbol{L}_{ki}, \tag{2.7}$$

representing the derivative of bilinear terms of pencil p with respect to x_k.

In the following definition, we introduce a generalization of the MFCQ condition to cover infeasible points as well.

Definition 2.4 (G-MFCQ Condition)**.** *An arbitrary point $\boldsymbol{x} \in \mathbb{R}^n$ is said to satisfy the Generalized Mangasarian-Fromovitz constraint qualification (G-MFCQ) condition for problem (2.1a)–(2.1b), if there exists $\boldsymbol{b} \in \mathbb{R}^n$ where*

$$\sum_{k\in\mathcal{N}} b_k(\boldsymbol{K}_k + \delta_k(\boldsymbol{x})) \prec 0. \tag{2.8}$$

Moreover, define the G-MFCQ function $s : \mathbb{R}^n \to \mathbb{R}$ as

$$s(\boldsymbol{x}) \triangleq \max\Big\{\underline{\lambda}\Big(-\sum_{k\in\mathcal{N}} b_k(\boldsymbol{K}_k + \delta_k(\boldsymbol{x}))\Big)\Big|\, \|\boldsymbol{b}\|_2 = 1\Big\}, \tag{2.9}$$

where the operator $\underline{\lambda}(.)$ returns the minimum eigenvalue of its input argument.

2.4 Convex Relaxation

This section aims at introducing a family of convex relaxations for the lifted problem (2.3a) – (2.3c). Consider the following formulation:

$$\underset{\boldsymbol{x}\in\mathbb{R}^n,\boldsymbol{X}\in\mathbb{S}_n}{\text{minimize}} \quad \boldsymbol{c}^\top\boldsymbol{x} \tag{2.10a}$$

$$\text{subject to} \quad p(\boldsymbol{x},\boldsymbol{X}) \preceq 0, \tag{2.10b}$$

$$\boldsymbol{X}-\boldsymbol{x}\boldsymbol{x}^\top \in \mathcal{C}, \tag{2.10c}$$

in which $\mathcal{C}\subseteq\mathbb{S}_n$. Observe that the problems (2.10a) – (2.10c) and (2.3a) – (2.3c) are equivalent if $\mathcal{C}=\{0\}$. We consider different choices for $\mathcal{C}$, which make the constraint (2.10c) convex. First, the standard semidefinite programming (SDP) and second-order cone programming (SOCP) relaxations are discussed and then, we introduce a novel *parabolic relaxation*, which transforms the constraint (2.3c) into a set of convex quadratic inequalities. The optimal cost for each of the above convex relaxation can serve as a lower bound for the global cost of the original problem (2.1a) – (2.1b). If the optimal solution of a relaxed problem satisfies (2.3c), the relaxation is regarded as *exact*.

2.4.1 Semidefinite Programming Relaxation

The following choice for $\mathcal{C}$ leads to the SDP relaxation of the problem (2.10a) – (2.10c):

$$\mathcal{C}_1=\{\boldsymbol{H}\in\mathbb{S}_n \,|\, \boldsymbol{H}\succeq 0\}. \tag{2.11}$$

If $\mathcal{C}=\mathcal{C}_1$ the optimization problem (2.10a) – (2.10c) boils down to a semidefinite program, which can be efficiently solved in polynomial time up to any desired accuracy using the existing methods.

2.4.2 Second-Order Cone Programming Relaxation

Semidefinite programming can be computationally demanding and its application is limited to small-scale problems. A popular alternative is the SOCP relaxation which can be deduced from the following choice for $\mathcal{C}$:

$$\mathcal{C}_2 = \{\boldsymbol{H} \in \mathbb{S}_n \mid H_{ii} \geq 0, \;\; H_{ii}H_{jj} \geq H_{ij}^2, \;\; \forall i,j \in \mathcal{N}\}. \tag{2.12}$$

It is straightforward to show that $\mathcal{C}_1$ is a subset of $\mathcal{C}_2$, which implies that the lower bounds from SDP relaxation are guaranteed to be tighter than or equal to the lower bounds obtained by SOCP relaxation.

2.4.3 Parabolic Relaxation

In this subsection, the parabolic relaxation is introduced as a computationally efficient alternative to SDP and SOCP relaxations. Parabolic relaxation transforms the non-convex constraint (2.3c) to a number of convex quadratic inequalities. To formulate the parabolic relaxation of the problem (2.10a)–(2.10c), the following choice for $\mathcal{C}$ should be employed:

$$\mathcal{C}_3 = \{\boldsymbol{H} \in \mathbb{S}_n \,|\, H_{ii} \geq 0, H_{ii} + H_{jj} \geq 2\,|H_{ij}|, \forall i,j \in \mathcal{N}\}. \tag{2.13}$$

It can be easily observed that if $\mathcal{C} = \mathcal{C}_3$, then the constraint (2.10c) is equivalent to the following quadratic inequalities:

$$X_{ii} + X_{jj} - 2X_{ij} \geq (x_i - x_j)^2 \qquad \forall i,j \in \mathcal{N}, \tag{2.14a}$$

$$X_{ii} + X_{jj} + 2X_{ij} \geq (x_i + x_j)^2 \qquad \forall i,j \in \mathcal{N}, \tag{2.14b}$$

which means that the parabolic relaxation is computationally cheaper than the SDP and SOCP relaxations.

Note that the presented relaxations are not necessarily exact. In the next section, the objective function (2.10a) is revised to facilitate the recovery of feasible points for the original non-convex problem (2.1a) – (2.1b).

2.5 Penalized Convex Relaxation

The *penalized convex relaxation* of the BMI optimization (2.1a) – (2.1b) is given as

$$\underset{\boldsymbol{x}\in\mathbb{R}^n,\boldsymbol{X}\in\mathbb{S}_n}{\text{minimize}} \quad \boldsymbol{c}^\top\boldsymbol{x} + \eta\,(\mathrm{tr}\{\boldsymbol{X}\} - 2\,\check{\boldsymbol{x}}^\top\boldsymbol{x} + \check{\boldsymbol{x}}^\top\check{\boldsymbol{x}}) \tag{2.15a}$$

$$\text{subject to} \quad p(\boldsymbol{x},\boldsymbol{X}) \;\preceq 0, \tag{2.15b}$$

$$\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top \in \mathcal{C}, \tag{2.15c}$$

where $\check{\boldsymbol{x}} \in \mathbb{R}^n$ is an initial guess for the unknown solution (either feasible or infeasible), $\eta > 0$ is a regularization parameter, which offers a trade-off between the original objective function and the penalty term, and $\mathcal{C} \in \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\}$.

The next theorem states that if the initial point $\check{\boldsymbol{x}}$ is feasible and satisfies MFCQ, then the penalized convex relaxation preserves the feasibility of $\check{\boldsymbol{x}}$ and produces a solution with improved objective value.

Theorem 2.1. *Assume that $\check{\boldsymbol{x}} \in \mathcal{F}$ is a feasible point for problem (2.1a)–(2.1b) that satisfies the MFCQ condition. If $\mathcal{C} \in \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\}$ and η is sufficiently large, then the penalized convex relaxation problem (2.15a)–(2.15c) has a unique solution $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{X}})$, which satisfies $\overset{*}{\boldsymbol{X}} = \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top$ and $\boldsymbol{c}^\top\overset{*}{\boldsymbol{x}} \leq \boldsymbol{c}^\top\check{\boldsymbol{x}}$.*

Proof. See Proof section for the proof. □

According to Theorem 2.1, the proposed penalized relaxation preserves the feasibility of the initial point. In what follows, we show that if the initial point is not feasible for

(2.1a) – (2.1b), but sufficiently close to its feasible set, then the penalized convex relaxation problem (2.15a) – (2.15c) is guaranteed to produce a feasible solution as well.

Theorem 2.2. *Assume that $k \in \{1, 2, 3\}$ and $\mathcal{C} = \mathcal{C}_k$. Consider an arbitrary point $\check{\boldsymbol{x}} \in \mathbb{R}^n$, which satisfies the G-MFCQ condition for problem (2.1a) – (2.1b), and let*

$$\frac{d_{\mathcal{F}}(\check{\boldsymbol{x}})}{s(\check{\boldsymbol{x}})} \leq \frac{\omega_k}{\|p\|_2} \tag{2.16}$$

where $\omega_1 = 4^{-1}$, $\omega_2 = (2n)^{-1}$, and $\omega_3 = (2 + 2\sqrt{n})^{-1}$. If η is sufficiently large, then the penalized convex relaxation problem (2.15a) – (2.15c) has a unique solution $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{X}})$, which satisfies $\overset{*}{\boldsymbol{X}} = \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^{\top}$.

Proof. See Proof section for the proof. □

Given the results of Theorems 2.1 and 2.2, there is a possibility to further improve the quality of the solution. Towards this end, we propose a sequential scheme which starts from an initial point and solves a sequence of penalized relaxations of form (2.15a) – (2.15c) to achieve feasible and near-globally optimal points for the BMI problem (2.1a) – (2.1b).

2.6 Sequential Penalized Relaxation

Theorems 2.1 and 2.2 give the conclusion that the proposed penalized convex relaxation is guaranteed to maintain the feasibility of Mangasarian-Fromovitz regular starting points. This property is held for infeasible points as well if they are sufficiently close to the BMI feasible set. Inspired by that, we propose a sequential scheme which can start from an arbitrary starting point and solves a sequence of penalized relaxations to recover a feasible point. If the feasibility is obtained, the sequential algorithm is guaranteed to maintain the feasibility and improve the objective value in each round of the algorithm. Algorithm 1 depicts the details of this procedure.

Algorithm 1 Sequential Penalized Relaxation

Input: $\check{\boldsymbol{x}} \in \mathbb{R}^n$, $\eta > 0$,$\mathrm{maxRound} \in \mathbb{N}$,$k = 0$

Output: $\overset{*}{\boldsymbol{x}}$

1: $\boldsymbol{x}_0 \leftarrow \check{\boldsymbol{x}}$
2: **repeat**
3: $\quad k \leftarrow k + 1$
4: $\quad \boldsymbol{x}_k \leftarrow$ Solve penalized relaxation (2.15a) – (2.15c)
5: $\quad \check{\boldsymbol{x}} \leftarrow \boldsymbol{x}_k$
6: **until** $k \leq \mathrm{maxRound}$
7: $\overset{*}{\boldsymbol{x}} \leftarrow \boldsymbol{x}_{\mathrm{maxRound}}$

The proposed algorithm proceeds until the stopping criteria is met. Notice that using the Nesterov's acceleration method can greatly enhance the convergence behavior of Algorithm 1. However, in this case, the resulting algorithm may not necessarily preserve the feasibility of the solutions.

2.7 Applications in Machine Learning

This section presents multiple machine learning problems that can be solved using the idea presented in this project.

2.7.1 Discriminative Dimensionality Reduction

Given a set of sample points from c different classes, the discriminative dimensionality reduction problem aims to infer a low-dimensional subspace on which the sample points of different classes are projected as far as possible. To find such a subspace, a max-min distance analysis (MMDA) method is presented in [14] that seeks to maximize the minimum

distance between all pairs of classes. This problem can be formulated as the following non-convex and non-smooth optimization problem

$$\underset{\boldsymbol{P}\in\mathbb{R}^{n\times m}}{\text{maximize}} \qquad \min_{1\leq i<j\leq c} \langle \boldsymbol{A}^{ij}, \boldsymbol{P}\boldsymbol{P}^{\top}\rangle \tag{2.17a}$$

$$\text{subject to} \qquad \boldsymbol{P}^{\top}\boldsymbol{P} = \boldsymbol{I}_m, \tag{2.17b}$$

where $\boldsymbol{A}^{ij} \in \mathbb{S}_n$ is a given weighted distance matrix between the i^{th} and j^{th} classes, $\boldsymbol{P}$ shows the projection matrix, and m denotes the dimension of the desired subspace. Observe that this problem is computationally hard to solve due to (possibly) non-convex objective function and orthogonality constraint. The solution of (2.17a) – (2.17b) can be obtained using the idea of penalized relaxation presented in this project. More details about the algorithm used to solve this problem are presented in [101].

2.7.2 Robust Minimum Volume Ellipsoid

Minimum volume ellipsoid (MVE) aims to find the smallest ellipsoid that covers a bunch of given sample points. This problem is regarded as a fundamental and well-studied problem in the area of machine learning. In the presence of outliers, [102] introduced robust minimum volume ellipsoid (RMVE) as a variant of MVE that allows a portion of the samples to lie outside the ellipsoid. This problem can be formulated as the following optimization problem

$$\underset{\boldsymbol{\gamma}\in\mathbb{R}^n,\, \boldsymbol{M}\in\mathbb{S}_d}{\text{minimize}} \qquad -\log(\boldsymbol{M}) + \eta\, l(\boldsymbol{\gamma}) \tag{2.18a}$$

$$\text{subject to} \qquad \boldsymbol{y}_i^{\top}\boldsymbol{M}\boldsymbol{y}_i \leq 1+\gamma_i, \qquad i=1,\ldots,n \tag{2.18b}$$

$$\boldsymbol{M} \succ 0, \;\; \boldsymbol{\gamma} \geq 0, \tag{2.18c}$$

where $\boldsymbol{M} \succ 0$ indicates the positive-definiteness of matrix $\boldsymbol{M}$, $\{\boldsymbol{y}_i \in \mathbb{R}^d\}_{i=1}^n$ denotes the sample points, function $l : \mathbb{R}^n \to \mathbb{R}$ is a regularization used to encourage a fraction of the samples to lie outside the ellipsoid, identify as outliers, $\eta > 0$ is regularization parameter

to control the relative weight between the objective function and the regularization, and $\boldsymbol{M} \in \mathbb{S}_d$ is a matrix which determines the ellipsoid. Observe that (2.18a) – (2.18c) can be cast as an LMI program, a special case of a BMI program in which the problem lacks the bilinear terms and is a convex conic problem, except if regularization function $l(.)$ introduces non-convexity into the problem. For many non-convex choices of regularization term $l(.)$ (e.g. non-convex quadratic function), the proposed relaxation technique can be utilized to convexify the problem.

2.7.3 Graph Matching

The graph matching (GM) problem has been widely employed in a variety of applications such as shape matching [103], object categorization [104, 105, 106], feature tracking [107], and kernelized sorting [108]. This problem seeks to find the best correspondence between the vertices of two graphs based on a given affinity matrix. The GM problem can be conveniently cast as a quadratic assignment problem (QAP) [109] which is generally computationally NP-hard to solve due to the (possibly) non-convex objective function and combinatorial constraints. Various frameworks have been developed to approximate the solution of this combinatorial problem using a convex program [110, 111, 112, 113].

One popular approach is to first relax the discrete constraints to obtain a continuous formulation [114, 115, 116] and then solve the resulting problem followed by a discretization step to recover feasible points for the QAP [110, 117]. This strategy may lead to poor quality points since the discretization step is performed independently from solving the continuous problem [118]. To address this issue, [111, 118] proposed to impose sparseness on the solution of the continuous problem to promote the discreteness.

Another common strategy is to convert the QAP into solving a series of problems whose solutions gradually converge to a feasible point of the QAP [119, 112, 3]. [119] proposed an iterative algorithm which starts from an initial point, either discrete or contin-

uous, and iteratively utilizes a linear approximation of the QAP to find high quality discrete points. [112] proposed to decompose the QAP as a convex-concave problem and [120] developed a path-following method which starts from the solution of the convex program and searches along a path of solutions to find the optimal solution of the concave problem. Based on this idea, [121, 113] proposed to factorize a large-scale affinity matrix into a set of smaller ones and introduced a convex-concave relaxation based on the factorized matrices. Later in [122], a strategy is introduced to improve the performance of path-following methods by first detecting the singular points and then branching at these points to improve the quality of paths.

This chapter proposes a path-following method for solving QAPs in a computationally tractable manner. To this end, we first convert the problem into a continuous one and then incorporate a regularization term to convexify the problem and promote the discreteness of the solutions. We investigate the theoretical conditions under which obtaining discrete solutions is guaranteed. Additionally, we propose a numerical algorithm based on the alternating direction method of multipliers (ADMM), which decomposes the regularized problem into two sub-problems with closed-form solutions and iteratively solves them until convergence is achieved. Since the solution of the regularized problem may not satisfy the discrete constraints, we propose to solve the problem in a sequential framework to recover feasible and near-optimal solutions. Numerical results demonstrate that the sequential algorithm not only eliminates the necessity of discretization and rounding steps but also exhibits comparable performance on two well-known datasets: CMU house dataset and Car-Motorbike dataset [2].

Consider graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$ where sets $\mathcal{V}$ and $\bar{\mathcal{V}}$ denote the vertices with the corresponding edge sets $\mathcal{E}$ and $\bar{\mathcal{E}}$, respectively. Let $|\mathcal{V}| = n$ and $|\bar{\mathcal{V}}| = m$ and assume $n \geq m$. The GM problem aims to find the best matching between the vertices

of the graphs $\mathcal{G}$ and $\bar{\mathcal{G}}$ in terms of a pre-defined similarity measure. This problem can be formulated as the following quadratic assignment problem

$$\underset{\boldsymbol{X}\in\mathbb{R}^{n\times m}}{\text{maximize}} \qquad \text{vec}(\boldsymbol{X})^{\top}\boldsymbol{W}\,\text{vec}(\boldsymbol{X}) \tag{2.19a}$$

$$\text{subject to} \qquad \boldsymbol{X}\in\Pi, \tag{2.19b}$$

where operator $\text{vec}(\cdot)$ stacks the columns of its input matrix to form a column vector, Π is the set of (sub)-permutation matrices, defined as $\Pi \triangleq \{\boldsymbol{X}\in\mathbb{R}^{n\times m}|\boldsymbol{X}\mathbf{1}\leq\mathbf{1},\boldsymbol{X}^{\top}\mathbf{1}=\mathbf{1},\boldsymbol{X}\circ\boldsymbol{X}-\boldsymbol{X}=\mathbf{0}\}$, $\circ$ shows the Hadamard product, and $\boldsymbol{W}\in\mathbb{S}_{n\times m}$ is a global pair-wise affinity matrix that encodes the vertex and edge affinity matrices of $\mathcal{G}$ and $\bar{\mathcal{G}}$ [3]. Observe that (2.19a) – (2.19b) is a non-convex optimization problem due to the non-convex objective function and the discrete constraints imposed on $\boldsymbol{X}$.

Define the set of (sub)-doubly stochastic matrices $\mathcal{D}\triangleq\{\boldsymbol{X}\in\mathbb{R}^{n\times m}|\boldsymbol{X}\geq 0,\boldsymbol{X}\mathbf{1}\leq\mathbf{1},\boldsymbol{X}^{\top}\mathbf{1}=\mathbf{1}\}$ as the relaxation of set Π. By replacing Π with $\mathcal{D}$ and incorporating a quadratic regularization term, (2.19a) – (2.19b) can be reduced into the following optimization problem

$$\underset{\boldsymbol{X}\in\mathbb{R}^{n\times m}}{\text{minimize}} \qquad \text{vec}(\boldsymbol{X})^{\top}\bar{\boldsymbol{W}}\,\text{vec}(\boldsymbol{X})+\eta\|\boldsymbol{X}-\check{\boldsymbol{X}}\|_{\text{F}}^{2} \tag{2.20a}$$

$$\text{subject to} \qquad \boldsymbol{X}\in\mathcal{D}, \tag{2.20b}$$

where $\bar{\boldsymbol{W}}\triangleq-\boldsymbol{W}$, matrix $\check{\boldsymbol{X}}\in\mathbb{R}^{n\times m}$ is an initial guess for the optimal solution of (2.19a) – (2.19b) and $\eta>0$ is a fixed parameter that controls the trade-off between the objective function and the regularization term. Notice that although (2.20b) is a convex constraint, (2.20a) – (2.20b) can be a non-convex problem due to the indefiniteness of the matrix $\bar{\boldsymbol{W}}$. It can be simply verified that for sufficiently large η ($\eta>|\underline{\lambda}(\bar{\boldsymbol{W}})|$, where $\underline{\lambda}(\bar{\boldsymbol{W}})$ indicates the smallest eigenvalue of matrix $\bar{\boldsymbol{W}}$), problem (2.20a) – (2.20b) turns into a convex problem whose solution approximates the optimal solution of (2.19a) – (2.19b). In this case, (2.20a) – (2.20b) is efficiently solvable in polynomial time, but its solution

may not necessarily lead to a meaningful correspondence between the graphs $\mathcal{G}$ and $\bar{\mathcal{G}}$. In what follows, we establish theoretical conditions for obtaining feasible points for (2.19a) – (2.19b).

Theorem 2.3. *Let $\check{\boldsymbol{X}} \in \mathbb{R}^{n\times m}$ satisfies the generalized linear independence constraint qualification condition for (2.19a) – (2.19b) [123]. Define $\boldsymbol{J}(\check{\boldsymbol{X}})$ as the Jacobian of all quasi-binding constraints in* (2.19b)*. The optimal solution of (2.20a) – (2.20b) is a feasible point for (2.19a) – (2.19b) if η is sufficiently large and the following inequality holds true*

$$d_{\scriptscriptstyle\Pi}(\check{\boldsymbol{X}}) \leq \frac{\underline{\sigma}(\boldsymbol{J}(\check{\boldsymbol{X}}))}{4}, \tag{2.21}$$

where $\underline{\sigma}(.)$ returns the smallest singular value of its input, function $d_{\scriptscriptstyle\Pi} : \mathbb{R}^{n\times m} \to \mathbb{R}$, defined as $d_{\scriptscriptstyle\Pi}(\boldsymbol{X}) \triangleq \inf\{\|\boldsymbol{C} - \boldsymbol{X}\|_{\scriptscriptstyle\mathrm{F}} \mid \boldsymbol{C} \in \Pi\}$, gives the minimum distance between every point $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ and set Π.

Proof. Proof can be derived from Theorem 2 of [101] and Theorem 3.4 of [123]. □

To find the optimal solution convex problem (2.20a) – (2.20b), we propose an efficient ADMM-based numerical algorithm [17]. To obtain an ADMM formulation for the problem, we introduce slack variable $\boldsymbol{s} \in \mathbb{R}^n$ and auxiliary variables $\boldsymbol{Z} \in \mathbb{R}^{n\times m}$, $\mathbf{u} \in \mathbb{R}^n$ and rewrite problem (2.20a) – (2.20b) as the following form

$$\underset{\substack{\boldsymbol{X},\boldsymbol{Z}\in\mathbb{R}^{n\times m}\\ \boldsymbol{s},\boldsymbol{u}\in\mathbb{R}^n}}{\text{minimize}} \quad \mathrm{vec}(\boldsymbol{X})^\top \bar{\boldsymbol{W}} \mathrm{vec}(\boldsymbol{X}) + \eta\|\boldsymbol{X} - \check{\boldsymbol{X}}\|_{\scriptscriptstyle\mathrm{F}}^2 + \frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{Z}\|_{\scriptscriptstyle\mathrm{F}}^2 + \frac{\mu}{2}\|\boldsymbol{s} - \mathbf{u}\|_2^2 \tag{2.22a}$$

$$\text{subject to} \quad \boldsymbol{Z} \geq 0, \;\; \mathbf{u} \geq 0, \tag{2.22b}$$

$$\boldsymbol{X}\mathbf{1}_m + \boldsymbol{s} = \mathbf{1}_n, \;\; \boldsymbol{X}^\top \mathbf{1}_n = \mathbf{1}_m, \tag{2.22c}$$

$$\boldsymbol{X} = \boldsymbol{Z}, \;\; \boldsymbol{s} = \mathbf{u}, \tag{2.22d}$$

where $\mu > 0$ is a given parameter. It is noteworthy to mention that (2.22a) – (2.22d) and (2.20a) – (2.20b) are equivalent as the additional terms in the objective function (2.22a) vanish for any feasible point. The solution of (2.22a) – (2.22d) can be obtained by simply

alternatingly solving some sub-problems with closed-form solutions. To serve this purpose, define function $\hat{\mathcal{L}}(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}, \boldsymbol{u}, \boldsymbol{\Lambda}, \boldsymbol{\lambda})$ as

$$\hat{\mathcal{L}}(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}, \boldsymbol{u}, \boldsymbol{\Lambda}, \boldsymbol{\lambda}) \triangleq \mathrm{vec}(\boldsymbol{X})^{\top} \bar{\boldsymbol{W}} \mathrm{vec}(\boldsymbol{X}) + \eta \|\boldsymbol{X} - \check{\boldsymbol{X}}\|_{\mathrm{F}}^2 + \frac{\mu}{2} \left\| \boldsymbol{X} - \boldsymbol{Z} + \frac{\boldsymbol{\Lambda}}{\mu} \right\|_{\mathrm{F}}^2 + \frac{\mu}{2} \left\| \boldsymbol{s} - \mathbf{u} + \frac{\boldsymbol{\lambda}}{\mu} \right\|_2^2,$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times m}$ and $\boldsymbol{\lambda} \in \mathbb{R}^n$ denote the Lagrange multipliers associated with the equality constraints (2.22d). Starting from random initialization for variables $\boldsymbol{Z}$, $\boldsymbol{u}$, $\boldsymbol{\Lambda}$, and $\boldsymbol{\lambda}$, the proposed algorithm updates them for the next iteration as

$$(\boldsymbol{X}^{k+1}, \boldsymbol{s}^{k+1}) := \underset{\boldsymbol{X}, \boldsymbol{s}}{\operatorname{argmin}} \ \hat{\mathcal{L}}(\boldsymbol{X}, \boldsymbol{Z}^k, \boldsymbol{s}, \boldsymbol{u}^k, \boldsymbol{\Lambda}^k, \boldsymbol{\lambda}^k) \quad \text{subject to} \quad (2.22c), \tag{2.23a}$$

$$(\boldsymbol{Z}^{k+1}, \boldsymbol{u}^{k+1}) := \underset{\boldsymbol{Z}, \boldsymbol{u}}{\operatorname{argmin}} \ \hat{\mathcal{L}}(\boldsymbol{X}^{k+1}, \boldsymbol{Z}, \boldsymbol{s}^{k+1}, \boldsymbol{u}, \boldsymbol{\Lambda}^k, \boldsymbol{\lambda}^k) \ \text{subject to} \quad (2.22b), \tag{2.23b}$$

$$\boldsymbol{\Lambda}^{k+1} := \boldsymbol{\Lambda}^k + \mu \, (\boldsymbol{X}^{k+1} - \boldsymbol{Z}^{k+1}), \tag{2.23c}$$

$$\boldsymbol{\lambda}^{k+1} := \boldsymbol{\lambda}^k + \mu \, (\, \boldsymbol{s}^{k+1} - \boldsymbol{u}^{k+1}). \tag{2.23d}$$

It is straightforward to verify that sub-problems (2.23a) and (2.23b) possess closed-form solutions. The optimal solution of (2.23a) is obtained by solving a system of linear equations and (2.23b) is a simple Euclidean projection onto the nonnegative orthant. Note that the solution of (2.22a) – (2.22d) may not necessarily lead to a valid matching between the graphs $\mathcal{G}$ and $\bar{\mathcal{G}}$. To circumvent this issue, we propose to start from an arbitrary point $\check{\boldsymbol{X}} = \boldsymbol{X}^0 \in \mathbb{R}^{n \times m}$ and sequentially solve (2.22a) – (2.22d) to recover feasible and near-optimal points for (2.19a) – (2.19b). Due to the discrete nature of the QAP problem, the idea of stochastic hill-climbing may work well in practice to boost the performance of the sequential method and avoid getting stuck in poor local minima of (2.19a) – (2.19b).

We evaluated the proposed sequential scheme, termed SDC, on two benchmark datasets for graph matching: Car-Motorbike [2] and the CMU house datasets. In the first experiment, we conduct an experiment on image pairs selected from the Car-Motorbike dataset to evaluate the SDC on feature matching task. Following the experimental setting used in

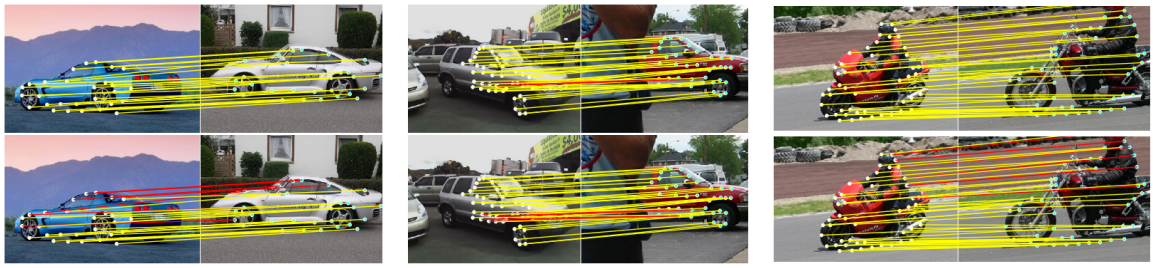

Figure 2.1: Examples of image matching on the the car and motorbike dataset [2]. **Top:** SDC, **bottom:** FGM-D [3]. Yellow lines and red lines, respectively, indicate the correct and indicate incorrect matches.

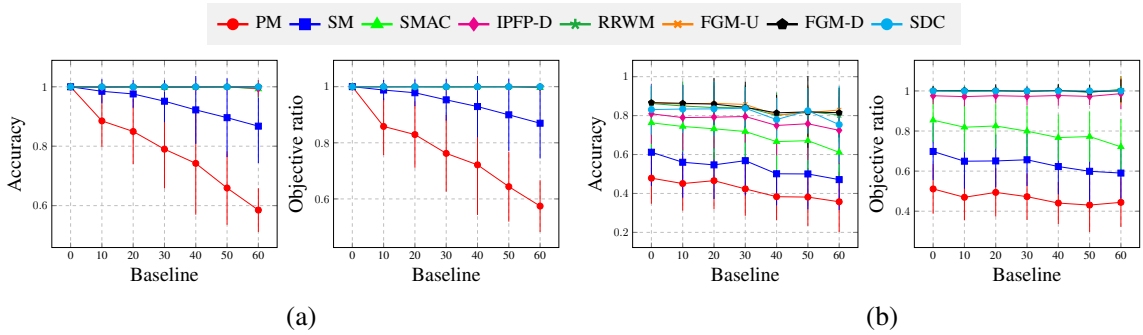

Figure 2.2: Comparison results of several graph matching algorithms on the CMU house dataset using (a) 30 nodes, (b) 25 nodes.

[121], we create the global pair-wise affinity matrix $\bar{\boldsymbol{W}}$ based on the orientation of each node's normal vector to the contour where the node was sampled. Figure 2.1 compares the performance of SDC against FGM-D [3] on three example image pairs.

In the second experiment, we compare our results with some prior works on the GM problem: FGM-D [3], FGM-U [121], RRWM [117], IPFP-D [119], PM [124], SMAC [115], and SM [110]. Figure 2.2 demonstrates the performance of all methods on the CMU house dataset which consists of 111 images where each one is manually labeled with 30 landmarks. Through this experiment, we set $\eta = \underline{\lambda}(\bar{\boldsymbol{W}}) + 10$ and use the same affinity matrix $\bar{\boldsymbol{W}}$ as [3]. The results indicate that SDC performs well on both datasets and achieves on par results compared to the earlier works.

2.8 Conclusions

In this chapter, a variety of convex relaxation methods are introduced for solving the class of optimization problems with bilinear matrix inequality (BMI) constraints. First, the well-known SDP and SOCP relaxations are discussed, and then a novel parabolic relaxation is introduced as a low-complexity alternative to conic relaxations. We propose a penalization method which is compatible with SDP, SOCP, and parabolic relaxations, and is able to produce feasible solutions for the original non-convex BMI optimization problem. Also, we generalized the proposed penalized relaxation to a sequential scheme which can start from an arbitrary initial point to recover feasible and near-optimal solutions of BMI problems. We show the applicability of the proposed sequential scheme on some fundamental machine learning problems. Based on that, we develop a convexification technique to solve the problem of graph matching and evaluate the method on two benchmark datasets. Experiments demonstrate the potential of the proposed approach in finding high quality solutions of the graph matching problem.

2.9 Proofs

In order to prove Theorems 2.1 and 2.2, we need to consider the following non-convex optimization problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \qquad \boldsymbol{c}^\top\boldsymbol{x}+\eta\|\boldsymbol{x}-\check{\boldsymbol{x}}\|_2^2 \tag{2.24a}$$

$$\text{subject to} \qquad p(\boldsymbol{x},\boldsymbol{x}\boldsymbol{x}^\top)\preceq 0, \tag{2.24b}$$

where $\check{\boldsymbol{x}}\in\mathbb{R}^n$ is the initial point. Observe that problems (2.1a)–(2.1b) and (2.24a)–(2.24b) have the same feasible set, which is denoted by $\mathcal{F}$. Assume that $\mathcal{F}$ is nonempty with an arbitrary member $\boldsymbol{x}'$. We define

$$\mathcal{A}\triangleq\Big\{\boldsymbol{x}\in\mathbb{R}^n\Big|\boldsymbol{c}^\top\boldsymbol{x}+\eta\|\boldsymbol{x}-\check{\boldsymbol{x}}\|_2^2\leq\boldsymbol{c}^\top\boldsymbol{x}'+\eta\|\boldsymbol{x}'-\check{\boldsymbol{x}}\|_2^2\Big\}. \tag{2.25}$$

Due to the compactness of the set $\mathcal{A} \cap \mathcal{F}$, it is straightforward to verify that the optimal solution of the problem (2.24a) – (2.24b) is attainable if $\eta > 0$.

Lemma 2.1. *Given an arbitrary $\varepsilon > 0$, every optimal solution $\overset{*}{\boldsymbol{x}}$ of problem (2.24a) – (2.24b) satisfies*

$$0 \leq \|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2 - d_{\mathcal{F}}(\check{\boldsymbol{x}}) \leq \varepsilon, \tag{2.26}$$

if η is sufficiently large.

Proof. Consider an optimal solution $\overset{*}{\boldsymbol{x}}$. Due to the Definition 2.2, the distance between $\check{\boldsymbol{x}}$ and every member of $\mathcal{F}$ is greater than or equal to $d_{\mathcal{F}}(\check{\boldsymbol{x}})$. Hence, the following inequality holds:

$$0 \leq \|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2 - d_{\mathcal{F}}(\check{\boldsymbol{x}}). \tag{2.27}$$

Let $\boldsymbol{x}_h$ be an arbitrary member of $\{\boldsymbol{x} \in \mathcal{F} | \|\boldsymbol{x} - \check{\boldsymbol{x}}\|_2 = d_{\mathcal{F}}(\check{\boldsymbol{x}})\}$. Due to the optimality of $\overset{*}{\boldsymbol{x}}$, we have:

$$\boldsymbol{c}^\top \overset{*}{\boldsymbol{x}} + \eta \|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2^2 \leq \boldsymbol{c}^\top \boldsymbol{x}_h + \eta \|\boldsymbol{x}_h - \check{\boldsymbol{x}}\|_2^2, \tag{2.28}$$

which implies that

$$\left\| (\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}) + \frac{\boldsymbol{c}}{2\eta} \right\|_2 \leq \left\| (\boldsymbol{x}_h - \check{\boldsymbol{x}}) + \frac{\boldsymbol{c}}{2\eta} \right\|_2. \tag{2.29}$$

Using the triangle inequality, we have

$$\|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2 - \frac{1}{2\eta}\|\boldsymbol{c}\|_2 \leq \|\boldsymbol{x}_h - \check{\boldsymbol{x}}\|_2 + \frac{1}{2\eta}\|\boldsymbol{c}\|_2, \tag{2.30}$$

which leads to the following upper-bound:

$$\|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2 - d_{\mathcal{F}}(\check{\boldsymbol{x}}) \leq \frac{1}{\eta}\|\boldsymbol{c}\|_2. \tag{2.31}$$

Hence, if $\eta \geq \frac{\|\boldsymbol{c}\|_2}{\varepsilon}$, the combination of (2.27) and (2.31) completes the proof. □

In what follows, we obtain sufficient conditions to ensure that every solution of (2.24a) – (2.24b) satisfy the MFCQ condition.

Lemma 2.2. *Assume that $\check{\boldsymbol{x}} \in \mathbb{R}^n$ is a feasible point for (2.24a) – (2.24b) that satisfies the MFCQ condition. If η is sufficiently large, every optimal solution $\overset{*}{\boldsymbol{x}}$ of (2.24a) – (2.24b), satisfies the MFCQ condition as well.*

Proof. Consider an optimal solution $\overset{*}{\boldsymbol{x}}$. Since the MFCQ condition holds for $\check{\boldsymbol{x}}$, there exists $\check{\boldsymbol{b}} \in \mathbb{R}^n$ for which the conic inequality $p(\check{\boldsymbol{x}}, \check{\boldsymbol{x}}\check{\boldsymbol{x}}^\top) + \sum_{k\in\mathcal{N}} \check{b}_k(\boldsymbol{K}_k + \delta_k(\check{\boldsymbol{x}})) \prec 0$ is satisfied. Hence, due to the continuity of the matrix pencil p, if ε is sufficiently small in Lemma 2.1, we have

$$p(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top) + \sum_{k\in\mathcal{N}} \overset{*}{b}_k(\boldsymbol{K}_k + \delta_k(\overset{*}{\boldsymbol{x}})) \prec 0 \tag{2.32}$$

which concludes the MFCQ condition holds for $\overset{*}{\boldsymbol{x}}$. □

Definition 2.5. *Given an arbitrary symmetric matrix $\boldsymbol{\Lambda} \in \mathbb{S}_m$, define the matrix function $\alpha : \mathbb{S}_m \to \mathbb{S}_n$ as,*

$$\alpha(\boldsymbol{\Lambda}) \triangleq \left[\langle \boldsymbol{L}_{ij}, \boldsymbol{\Lambda}\rangle\right]_{ij\in\mathcal{N}^2}. \tag{2.33}$$

It is straightforward to verify that

$$2\alpha(\boldsymbol{\Lambda})\boldsymbol{x} = \sum_{k\in\mathcal{N}} \langle \delta_k(\boldsymbol{x}), \boldsymbol{\Lambda}\rangle \boldsymbol{e}_k, \tag{2.34}$$

for every $\boldsymbol{x} \in \mathbb{R}^n$. This property will be used later in this section.

Lemma 2.3. *Assume that $\check{\boldsymbol{x}} \in \mathbb{R}^n$ satisfies*

$$s(\check{\boldsymbol{x}}) > 2\|p\|_2 d_{\mathcal{F}}(\check{\boldsymbol{x}}). \tag{2.35}$$

Given an arbitrary $\varepsilon > 0$, every optimal solution $\overset{*}{\boldsymbol{x}}$ of the problem (2.24a) – (2.24b) satisfies the inequality

$$s(\check{\boldsymbol{x}}) - s(\overset{*}{\boldsymbol{x}}) \leq 2\|p\|_2 d_{\mathcal{F}}(\check{\boldsymbol{x}}) + \varepsilon, \tag{2.36}$$

as well as the MFCQ condition, if η is sufficiently large.

Proof. Due to the definition of s, there exists $\check{\boldsymbol{b}} \in \mathbb{R}^n$ such that $\|\check{\boldsymbol{b}}\|_2 = 1$ and

$$s(\check{\boldsymbol{x}}) = \underline{\lambda}\Big(-\sum_{k\in\mathcal{N}} \check{b}_k(\boldsymbol{K}_k+\delta_k(\check{\boldsymbol{x}}))\Big). \tag{2.37}$$

As a result,

$$\begin{aligned} s(\overset{*}{\boldsymbol{x}}) &\geq \underline{\lambda}\Big(-\sum_{k\in\mathcal{N}} \check{b}_k(\boldsymbol{K}_k+\delta_k(\overset{*}{\boldsymbol{x}}))\Big) && \text{(2.38a)}\\ &= \underline{\lambda}\Big(-\sum_{k\in\mathcal{N}} \check{b}_k(\boldsymbol{K}_k+\delta_k(\check{\boldsymbol{x}}))-\sum_{k\in\mathcal{N}} \check{b}_k\delta_k(\overset{*}{\boldsymbol{x}}-\check{\boldsymbol{x}})\Big) && \text{(2.38b)}\\ &\geq s(\check{\boldsymbol{x}}) - \Big\| \sum_{k\in\mathcal{N}} \check{b}_k\delta_k(\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}})\Big\|_2. && \text{(2.38c)} \end{aligned}$$

Let $\boldsymbol{u}$ be the eigenvector corresponding to the largest eigenvalue of $-\sum_{k\in\mathcal{N}} \check{b}_k\delta_k(\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}})$. Then,

$$\begin{aligned} s(\check{\boldsymbol{x}}) - s(\overset{*}{\boldsymbol{x}}) &\leq \Big\| \sum_{k\in\mathcal{N}} \check{b}_k\delta_k(\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}})\Big\|_2 && \text{(2.39a)}\\ &= \Big|\boldsymbol{u}^\top\Big(\sum_{k\in\mathcal{N}} \check{b}_k\delta_k(\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}})\Big)\boldsymbol{u}\Big| && \text{(2.39b)}\\ &= \Big|\sum_{k\in\mathcal{N}} \check{b}_k\langle\delta_k(\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}), \boldsymbol{u}\boldsymbol{u}^\top\rangle\Big| && \text{(2.39c)}\\ &= \big|\check{\boldsymbol{b}}^\top[\langle\delta_k(\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}), \boldsymbol{u}\boldsymbol{u}^\top\rangle]_{k\in\mathcal{N}}\big| && \text{(2.39d)}\\ &\leq \big\|[\langle\delta_k(\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}), \boldsymbol{u}\boldsymbol{u}^\top\rangle]_{k\in\mathcal{N}}\big\|_2. && \text{(2.39e)} \end{aligned}$$

On the other hand, according to the equation (2.34), we have

$$[\langle\delta_k(\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}), \boldsymbol{u}\boldsymbol{u}^\top\rangle]_{k\in\mathcal{N}} = 2\alpha(\boldsymbol{u}\boldsymbol{u}^\top)(\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}), \tag{2.40}$$

which implies that

$$\begin{aligned} s(\check{\boldsymbol{x}}) - s(\overset{*}{\boldsymbol{x}}) &\leq 2\|\alpha(\boldsymbol{u}\boldsymbol{u}^\top)\|_2\|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2 && \text{(2.41a)}\\ &\leq 2\|p\|_2\|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2. && \text{(2.41b)} \end{aligned}$$

Therefore, according to Lemma 2.1, we have

$$s(\check{\boldsymbol{x}}) - s(\overset{*}{\boldsymbol{x}}) \leq 2\|p\|_2 \|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2 \leq 2\|p\|_2 d_{\mathcal{F}}(\check{\boldsymbol{x}}) + \varepsilon,$$

if η is sufficiently large.

Additionally, for sufficiently small choices of ε, we have $s(\overset{*}{\boldsymbol{x}}) > 0$. Hence, there exists $\overset{*}{\boldsymbol{b}} \in \mathbb{R}^n$ such that $\sum_{k\in\mathcal{N}} \overset{*}{b}_k(\boldsymbol{K}_k + \delta_k(\overset{*}{\boldsymbol{x}})) \prec 0$ and due to the feasibility of $\overset{*}{\boldsymbol{x}}$, we have:

$$p(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top) + \sum_{k\in\mathcal{N}} \overset{*}{b}_k(\boldsymbol{K}_k + \delta_k(\overset{*}{\boldsymbol{x}})) \prec 0, \tag{2.42}$$

which concludes the MFCQ condition for $\overset{*}{\boldsymbol{x}}$. □

The following lemma ensures the existence of a dual certificate matrix, if the optimal solution of (2.24a) – (2.24b) satisfies the MFCQ condition.

Lemma 2.4. *For every optimal solution $\overset{*}{\boldsymbol{x}}$ of (2.24a) – (2.24b) which meets the MFCQ condition, there exists a dual matrix $\overset{*}{\boldsymbol{\Lambda}} \succeq 0$ such that the point $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{\Lambda}})$ satisfies the following Karush-Kuhn-Tucker (KKT) equations:*

$$\boldsymbol{c} + 2\eta(\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}) + \sum_{k\in\mathcal{N}} \langle \boldsymbol{K}_k, \overset{*}{\boldsymbol{\Lambda}} \rangle \boldsymbol{e}_k + 2\alpha(\overset{*}{\boldsymbol{\Lambda}})\overset{*}{\boldsymbol{x}} = 0, \tag{2.43a}$$

$$\overset{*}{\boldsymbol{\Lambda}} p(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top) = 0. \tag{2.43b}$$

Proof. Since the optimal solution $\overset{*}{\boldsymbol{x}}$ satisfies the MFCQ condition, there exists a dual matrix $\overset{*}{\boldsymbol{\Lambda}} \succeq 0$ such that the point $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{\Lambda}})$ satisfies the following conditions:

$$\nabla_{\boldsymbol{x}} \mathcal{L}_{\mathrm{p}}(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{\Lambda}}) = 0, \tag{2.44a}$$

$$\overset{*}{\boldsymbol{\Lambda}} p(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top) = 0, \tag{2.44b}$$

where $\nabla_{\boldsymbol{x}}$ represents the gradients with respect to $\boldsymbol{x}$ and $\mathcal{L}_{\mathrm{p}}(\boldsymbol{x}, \boldsymbol{\Lambda})$ denotes the Lagrangian function of (2.24a) – (2.24b):

$$\mathcal{L}_{\mathrm{p}}(\boldsymbol{x}, \boldsymbol{\Lambda}) = \boldsymbol{c}^\top \boldsymbol{x} + \eta \|\boldsymbol{x} - \check{\boldsymbol{x}}\|_2^2 + \langle p(\boldsymbol{x}, \boldsymbol{x}\boldsymbol{x}^\top), \boldsymbol{\Lambda} \rangle. \tag{2.45}$$

Observe that (2.43a) – (2.43b) and (2.44a) – (2.44b) are equivalent. Therefore, the point $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{\Lambda}})$ satisfies the KKT conditions (2.43a) – (2.43b). □

In following two lemmas bound the value of $\frac{\mathrm{tr}\{\overset{*}{\boldsymbol{\Lambda}}\}}{\eta}$ for both cases where $\check{\boldsymbol{x}}$ is feasible and infeasible.

Lemma 2.5. *Consider an arbitrary $\varepsilon > 0$ and assume that $\check{\boldsymbol{x}} \in \mathcal{F}$ is a feasible point for (2.24a) – (2.24b) that satisfies the MFCQ condition. If η is sufficiently large, for every optimal solution $\overset{*}{\boldsymbol{x}}$ of (2.24a) – (2.24b), there exists a dual matrix $\overset{*}{\boldsymbol{\Lambda}} \succeq 0$ that satisfies the inequality*

$$\frac{\mathrm{tr}\{\overset{*}{\boldsymbol{\Lambda}}\}}{\eta} \leq \varepsilon, \tag{2.46}$$

as well as the equations (2.43a) – (2.43b).

Proof. According to Lemma 2.2, if η is large enough, $\overset{*}{\boldsymbol{x}}$ satisfies the MFCQ condition. Hence, there exists $\overset{*}{\boldsymbol{b}} \in \mathbb{R}^n$ such that

$$-p(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top) - \sum_{k\in\mathcal{N}} \overset{*}{b}_k(\boldsymbol{K}_k + \delta_k(\overset{*}{\boldsymbol{x}})) \succeq 0. \tag{2.47}$$

In addition, according to Lemma 2.4 there exists $\overset{*}{\boldsymbol{\Lambda}} \succeq 0$ such that the pair $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{\Lambda}})$ satisfies the KKT equations (2.43a) – (2.43b). Therefore, pre-multiplying $\overset{*}{\boldsymbol{b}}^\top$ to both sides of (2.43a) yields:

$$\overset{*}{\boldsymbol{b}}^\top(\boldsymbol{c}+2\eta(\overset{*}{\boldsymbol{x}}-\check{\boldsymbol{x}})) + \langle \sum_{k\in\mathcal{N}} \overset{*}{b}_k(\boldsymbol{K}_k+\delta_k(\overset{*}{\boldsymbol{x}})), \overset{*}{\boldsymbol{\Lambda}}\rangle = 0. \tag{2.48}$$

Due to the matrix inequality (2.47) and since $\overset{*}{\boldsymbol{\Lambda}} \succeq 0$, we have:

$$\langle -p(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top) - \sum_{k\in\mathcal{N}} \overset{*}{b}_k(\boldsymbol{K}_k + \delta_k(\overset{*}{\boldsymbol{x}})), \overset{*}{\boldsymbol{\Lambda}}\rangle \geq 0. \tag{2.49}$$

Hence, according to the complementary slackness (2.43b), we have:

$$\mathrm{tr}\{\overset{*}{\boldsymbol{\Lambda}}\}s(\overset{*}{\boldsymbol{x}}) \leq \langle -p(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^{\top}) - \sum_{k\in\mathcal{N}} \overset{*}{b}_k(\boldsymbol{K}_k + \delta_k(\overset{*}{\boldsymbol{x}})), \overset{*}{\boldsymbol{\Lambda}}\rangle \tag{2.50a}$$

$$= \langle -\sum_{k\in\mathcal{N}} \overset{*}{b}_k(\boldsymbol{K}_k + \delta_k(\overset{*}{\boldsymbol{x}})), \overset{*}{\boldsymbol{\Lambda}}\rangle \tag{2.50b}$$

$$= \langle \overset{*}{\boldsymbol{b}}, \boldsymbol{c} + 2\eta(\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}})\rangle \tag{2.50c}$$

$$= \langle \overset{*}{\boldsymbol{b}}, \boldsymbol{c}\rangle + 2\eta\langle \overset{*}{\boldsymbol{b}}, (\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}})\rangle \tag{2.50d}$$

$$\leq \|\overset{*}{\boldsymbol{b}}\|_2\|\boldsymbol{c}\|_2 + 2\eta\|\overset{*}{\boldsymbol{b}}\|_2\|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2 \tag{2.50e}$$

$$= \|\boldsymbol{c}\|_2 + 2\eta\|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2, \tag{2.50f}$$

and therefore:

$$\frac{\mathrm{tr}\{\overset{*}{\boldsymbol{\Lambda}}\}}{\eta} \leq \frac{\|\boldsymbol{c}\|_2}{\eta s(\overset{*}{\boldsymbol{x}})} + \frac{2\|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2}{s(\overset{*}{\boldsymbol{x}})}. \tag{2.51}$$

According to Lemma 2.1, if η is large, $\|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2$ is arbitrarily small. Due to the continuity of s, we can argue that $|s(\overset{*}{\boldsymbol{x}}) - s(\check{\boldsymbol{x}})|$ is arbitrarily small as well. Now, since $s(\check{\boldsymbol{x}}) > 0$, the right side of the inequality (2.51) is not greater than ε, if η is sufficiently large. □

Lemma 2.6. *Consider an arbitrary $\varepsilon > 0$ and assume that $\check{\boldsymbol{x}} \in \mathbb{R}^n$ satisfies the inequality* (2.35). *If η is sufficiently large, for every optimal solution $\overset{*}{\boldsymbol{x}}$ of (2.24a)–(2.24b), there exists a dual matrix $\overset{*}{\boldsymbol{\Lambda}} \succeq 0$ that satisfies the inequality*

$$\frac{\mathrm{tr}\{\overset{*}{\boldsymbol{\Lambda}}\}}{\eta} \leq \frac{2d_{\mathcal{F}}(\check{\boldsymbol{x}})}{s(\check{\boldsymbol{x}}) - 2\|p\|_2 d_{\mathcal{F}}(\check{\boldsymbol{x}})} + \varepsilon, \tag{2.52}$$

as well as the equations (2.43a)–(2.43b).

Proof. According to the Lemma 2.3, $\overset{*}{\boldsymbol{x}}$ satisfies the MFCQ condition. In addition, the Lemma 2.4 implies that there exists $\overset{*}{\boldsymbol{\Lambda}} \succeq 0$ such that point $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{\Lambda}})$ satisfies the KKT equa-

tions (2.43a) – (2.43b). Since $s(\overset{*}{\boldsymbol{x}}) > 0$, we can similarly argue that the inequality (2.51) holds true:

$$\frac{\mathrm{tr}\{\overset{*}{\boldsymbol{\Lambda}}\}}{\eta} \leq \frac{\|\boldsymbol{c}\|_2}{\eta s(\overset{*}{\boldsymbol{x}})} + \frac{2\|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2}{s(\overset{*}{\boldsymbol{x}})} \leq \frac{\|\boldsymbol{c}\|_2 + 2\eta\|\overset{*}{\boldsymbol{x}} - \check{\boldsymbol{x}}\|_2}{\eta[s(\check{\boldsymbol{x}}) - 2\|p\|_2 d_{\mathcal{F}}(\check{\boldsymbol{x}})]}. \tag{2.53}$$

Now, according to Lemma 2.1, if η is large, the above inequality concludes (2.52). □

The next lemma presents sufficient conditions under which the optimal solution of (2.24a) – (2.24b) can be obtained by solving penalized convex relaxation.

Lemma 2.7. *Consider an optimal solution $\overset{*}{\boldsymbol{x}} \in \mathcal{F}$ for the problem (2.24a) – (2.24b), and a matrix $\overset{*}{\boldsymbol{\Lambda}} \succeq 0$ such that point $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{\Lambda}})$ satisfies the conditions (2.43a) – (2.43b). Then, the pair $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top)$ is the unique primal solution to the penalized convex relaxation problem (2.15a) – (2.15c), if the following conic inequality holds true:*

$$\eta \boldsymbol{I} + \alpha(\overset{*}{\boldsymbol{\Lambda}}) \succ_{\mathcal{C}_k^*} 0, \tag{2.54}$$

where $k \in \{1, 2, 3\}$, $\mathcal{C}_k^*$ denotes the dual cone of $\mathcal{C}_k$, and the matrices $\overset{*}{\boldsymbol{\Lambda}}$ and $\eta\boldsymbol{I} + \alpha(\overset{*}{\boldsymbol{\Lambda}})$ are the dual optimal Lagrange multipliers associated with the constraints (2.15b) *and* (2.15c)*, respectively.*

Proof. The Lagrangian of the penalized relaxation problem (2.15a) – (2.15c) can be formed as follows,

$$\begin{aligned} \mathcal{L}_{\mathrm{r}}(\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{\Lambda}) = \boldsymbol{c}^\top \boldsymbol{x} + \eta\langle \boldsymbol{X} - 2\boldsymbol{x}\check{\boldsymbol{x}}^\top, \boldsymbol{I}\rangle + \langle p(\boldsymbol{x}, \boldsymbol{X}), \boldsymbol{\Lambda}\rangle \\ - \langle \eta\boldsymbol{I} + \alpha(\overset{*}{\boldsymbol{\Lambda}}), \boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^\top\rangle, \end{aligned} \tag{2.55}$$

where $\eta \boldsymbol{I} + \alpha(\overset{*}{\boldsymbol{\Lambda}}) \in \mathcal{C}_k^*$ is the dual variable associated with (2.15c). Due to the convexity of the penalized relaxation problem, if a pair $\big((\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{X}}), \overset{*}{\boldsymbol{\Lambda}}\big)$ satisfies the KKT conditions

$$\boldsymbol{c} - 2\eta\check{\boldsymbol{x}} + \sum_{k\in\mathcal{N}} \langle \boldsymbol{K}_k, \overset{*}{\boldsymbol{\Lambda}} \rangle \boldsymbol{e}_k + 2(\eta \boldsymbol{I} + \alpha(\overset{*}{\boldsymbol{\Lambda}}))\overset{*}{\boldsymbol{x}} = 0, \tag{2.56a}$$

$$\langle p(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{X}}), \overset{*}{\boldsymbol{\Lambda}} \rangle = 0, \tag{2.56b}$$

$$p(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{X}}) \preceq 0, \tag{2.56c}$$

$$\eta \boldsymbol{I} + \alpha(\overset{*}{\boldsymbol{\Lambda}}) \in \mathcal{C}_k^*, \tag{2.56d}$$

then it is an optimal primal-dual solution for (2.15a) – (2.15c).

It can be easily verified that the KKT conditions (2.56a) – (2.56d) are satisfied for $\big((\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top), \overset{*}{\boldsymbol{\Lambda}}\big)$ as a direct consequence of (2.43a) – (2.43b), (2.54) and (2.24b). Moreover, $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top)$ is the unique solution of the primal problem since $\eta \boldsymbol{I} + \alpha(\overset{*}{\boldsymbol{\Lambda}})$ belongs to the interior of $\mathcal{C}_k^*$. □

Lemma 2.8. *Consider an optimal solution $\overset{*}{\boldsymbol{x}} \in \mathcal{F}$ for problem (2.24a) – (2.24b), and a matrix $\overset{*}{\boldsymbol{\Lambda}} \succeq 0$ such that point $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{\Lambda}})$ satisfies the KKT equations (2.43a) – (2.43b). The pair $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top)$ is the unique primal solution to the penalized convex relaxation problem (2.15a) – (2.15c), if the following inequality holds true:*

$$\frac{\mathrm{tr}\{\overset{*}{\boldsymbol{\Lambda}}\}}{\eta} \leq \frac{\zeta_k}{\|p\|_2} \tag{2.57}$$

where $k \in \{1, 2, 3\}$, $\zeta_1 = 1$, $\zeta_2 = (n-1)^{-1}$, and $\zeta_3 = n^{-\frac{1}{2}}$.

Proof. According to Lemma 2.5, it suffices to verify (2.54) in order to prove that $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^{\top})$ is the unique optimal solution. Denote the eigenvalues and eigenvectors of $\overset{*}{\boldsymbol{\Lambda}}$ by $\{\overset{*}{\lambda}_l\}_{l\in\mathcal{M}}$ and $\{\overset{*}{\boldsymbol{u}}_l\}_{l\in\mathcal{M}}$, respectively. Hence:

$$\big\|\alpha(\overset{*}{\boldsymbol{\Lambda}})\big\|_q = \big\|\sum_{l\in\mathcal{M}} \overset{*}{\lambda}_l[\langle \boldsymbol{L}_{ij}, \overset{*}{\boldsymbol{u}}_l\overset{*}{\boldsymbol{u}}_l^{\top}\rangle]_{ij}\big\|_q \tag{2.58a}$$

$$\leq \sum_{l\in\mathcal{M}} \overset{*}{\lambda}_l\big\|[\langle \boldsymbol{L}_{ij}, \overset{*}{\boldsymbol{u}}_l\overset{*}{\boldsymbol{u}}_l^{\top}\rangle]_{ij}\big\|_q \tag{2.58b}$$

$$\leq \sum_{l\in\mathcal{M}} \overset{*}{\lambda}_l\big\|\alpha(\overset{*}{\boldsymbol{u}}_l\overset{*}{\boldsymbol{u}}_l^{\top})\big\|_q = \|p\|_q \mathrm{tr}\{\overset{*}{\boldsymbol{\Lambda}}\}, \tag{2.58c}$$

1. SDP relaxation: The cone of positive semidefinite matrices is self-dual i.e., $\mathcal{C}_1^* = \mathcal{C}_1$. Therefore, in order to prove (2.54), it suffices to show that

$$\eta - \big\|\alpha(\overset{*}{\boldsymbol{\Lambda}})\big\|_2 \geq 0. \tag{2.59}$$

   Hence, according to the bound provided in (2.58), $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^{\top})$ is the unique solution for the penalized SDP relaxation, if

$$\frac{\mathrm{tr}\{\overset{*}{\boldsymbol{\Lambda}}\}}{\eta} \leq \frac{1}{\|p\|_2}. \tag{2.60}$$

2. SOCP relaxation: The dual cone $\mathcal{C}_2^*$ can be expressed as:

$$\mathcal{C}_2^* \triangleq \Big\{\sum_{i,j\in\mathcal{N}} [\boldsymbol{e}_i,\boldsymbol{e}_j]\, \boldsymbol{H}_{ij} [\boldsymbol{e}_i,\boldsymbol{e}_j]^{\top} \Big| \boldsymbol{H}_{ij} \in \mathbb{S}_2^+, \forall i,j \in \mathcal{N}\Big\}. \tag{2.61}$$

   Consider the following decomposition:

$$\eta\boldsymbol{I} + \alpha(\overset{*}{\boldsymbol{\Lambda}}) = \sum_{\substack{i,j\in\mathcal{N}\\ i\neq j}} [\boldsymbol{e}_i,\boldsymbol{e}_j]\, \boldsymbol{A}_{ij}\, [\boldsymbol{e}_i,\boldsymbol{e}_j]^{\top}, \tag{2.62}$$

   where for every $(i,j) \in \mathcal{N}^2$ we have

$$\boldsymbol{A}_{ij} \triangleq \begin{bmatrix} \frac{\eta - [\alpha(\overset{*}{\boldsymbol{\Lambda}})]_{ii}}{n-1} & -[\alpha(\overset{*}{\boldsymbol{\Lambda}})]_{ij} \\ -[\alpha(\overset{*}{\boldsymbol{\Lambda}})]_{ji} & \frac{\eta - [\alpha(\overset{*}{\boldsymbol{\Lambda}})]_{jj}}{n-1} \end{bmatrix} \succeq \left(\frac{\eta}{n-1} - \|\alpha(\overset{*}{\boldsymbol{\Lambda}})\|_2\right) \boldsymbol{I}_2$$

Therefore, the inequality (2.54) is satisfied for $\mathcal{C}_2^*$ if $\frac{\eta}{n-1} \geq \|\alpha(\overset{*}{\boldsymbol{\Lambda}})\|_2$. Now, according to the bound provided in (2.58), $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top)$ is the unique optimal solution for the penalized SOCP relaxation if

$$\frac{\mathrm{tr}\{\overset{*}{\boldsymbol{\Lambda}}\}}{\eta} \leq \frac{1}{(n-1)\|p\|_2}. \tag{2.63}$$

3. Parabolic relaxation: The dual cone of $\mathcal{C}_3$ is the set of $n \times n$ symmetric diagonally dominant matrices defined as:

$$\mathcal{C}_3^* = \Big\{ \boldsymbol{H} \in \mathbb{S}_n \;\Big|\; |H_{ii}| \geq \sum_{j \in \mathcal{N}\setminus\{i\}} |H_{ij}|,\ \forall i \in \mathcal{N} \Big\}. \tag{2.64}$$

Therefore, in order to prove (2.54), it suffices to show that

$$\eta - \big\|\alpha(\overset{*}{\boldsymbol{\Lambda}})\big\|_1 \geq 0. \tag{2.65}$$

Once again, the bound presented in (2.58) implies that $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top)$ is the unique solution for the penalized parabolic relaxation if $\frac{\mathrm{tr}\{\overset{*}{\boldsymbol{\Lambda}}\}}{\eta} \leq \frac{1}{\|p\|_1}$. This is a direct consequence of

$$\frac{\mathrm{tr}\{\overset{*}{\boldsymbol{\Lambda}}\}}{\eta} \leq \frac{n^{-\frac{1}{2}}}{\|p\|_2} \tag{2.66}$$

since $\|p\|_1 \leq n^{\frac{1}{2}}\|p\|_2$.

□

Theorem 2.1. Consider an arbitrary optimal solution $\overset{*}{\boldsymbol{x}}$ for the problem (2.24a) – (2.24b). According to Lemma 2.4, if η is large enough, there exists a dual matrix $\overset{*}{\boldsymbol{\Lambda}} \succeq 0$ such that point $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{\Lambda}})$ satisfies the KKT equations (2.43a) – (2.43b), as well as the inequality (2.46) for $\varepsilon = \frac{\min\{\zeta_1,\zeta_2,\zeta_3\}}{2\|p\|_2}$. Therefore, according to the Lemma 2.8, the pair $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^\top)$ is the unique primal solution to the penalized convex relaxation problem (2.15a) – (2.15c). □

Theorem 2.2. Consider an arbitrary optimal solution $\overset{*}{\boldsymbol{x}}$ for the problem (2.24a) – (2.24b). According to the Lemma 2.4, if η is sufficiently large, there exists a dual matrix $\overset{*}{\boldsymbol{\Lambda}} \succeq 0$ such that point $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{\Lambda}})$ satisfies the KKT equations (2.43a) – (2.43b), as well as the inequality (2.52) for any arbitrarily ε. It is straightforward to verify that

$$\frac{d_{\mathcal{F}}(\check{\boldsymbol{x}})}{s(\check{\boldsymbol{x}})} < \frac{\omega_k}{\|p\|_2} \quad \Rightarrow \quad \frac{2d_{\mathcal{F}}(\check{\boldsymbol{x}})}{s(\check{\boldsymbol{x}}) - 2\|p\|_2 d_{\mathcal{F}}(\check{\boldsymbol{x}})} < \frac{\zeta_k}{\|p\|_2}, \tag{2.67}$$

for all $k \in \{1, 2, 3\}$. Therefore, according to Lemma 2.8, the pair $(\overset{*}{\boldsymbol{x}}, \overset{*}{\boldsymbol{x}}\overset{*}{\boldsymbol{x}}^{\top})$ is the unique primal solution to the penalized convex relaxation problem (2.15a) – (2.15c). □

CHAPTER 3

Multi-Level Representation Learning for Deep Subspace Clustering

This chapter proposes a novel deep subspace clustering approach which uses convolutional autoencoders to transform input images into new representations lying on a union of linear subspaces. The first contribution of our work is to insert multiple fully-connected linear layers between the encoder layers and their corresponding decoder layers to promote learning more favorable representations for subspace clustering. These connection layers facilitate the feature learning procedure by combining low-level and high-level information for generating multiple sets of self-expressive and informative representations at different levels of the encoder. Moreover, we introduce a novel loss minimization problem which leverages an initial clustering of the samples to effectively fuse the multi-level representations and recover the underlying subspaces more accurately. The loss function is then minimized through an iterative scheme which alternatively updates the network parameters and produces new clusterings of the samples until the convergence is obtained. Experiments on four real-world datasets demonstrate that our approach exhibits superior performance compared to the state-of-the-art methods on most of the subspace clustering problems.

3.1 Introduction

Subspace clustering is an unsupervised learning task with a variety of machine learning applications such as motion segmentation [22, 23], face clustering [24, 25], and movie recommendation [26, 27], etc. The primary goal of this task is to partition a set of data samples, drawn from a union of low-dimensional subspaces, into disjoint clusters such that the

samples within each cluster belong to the same subspace [28, 29]. A large body of subspace clustering literature relies on the concept of self-expressiveness which states that each sample point in a union of subspaces is efficiently expressible in terms of a linear (or affine) combination of other points in the subspaces [34]. Given that, it is expected that the nonzero coefficients in the linear representation of each sample correspond to the points of the same subspace as the given sample. In order to successfully infer such underlying relationships among the samples and to partition them into their respective subspaces, a common practice approach is to first learn an affinity matrix from the input data and then apply the spectral clustering technique [125] to recover the clusters. Recently, these spectral clustering-based approaches have shown a special interest in utilizing sparse or low-rank representations of the samples to create more accurate affinity matrices [34, 126, 127, 128, 129]. A well-established instance is sparse subspace clustering (SSC) [34] which uses an ℓ_1-regularized model to select only a small subset of points belonging to the same subspace for reconstructing each data point. More theoretical and practical aspects of the SSC algorithm are investigated and studied in detail in [130, 131, 132, 133].

Despite the key role that the self-expressiveness plays in the literature, it may not be satisfied in a wide range of applications in which samples lie on non-linear subspaces, e.g. face images taken under non-uniform illumination and at different poses [35]. A common practice technique to handle these cases is to leverage well-known kernel trick to implicitly map the samples into a higher dimensional space so that they better conform to linear subspaces [36, 37, 38, 39]. Although this strategy has demonstrated empirical success, it is not widely applicable to various applications, mainly because identifying an appropriate kernel function for a given set of data points is a quite difficult task [40].

Recently, deep neural networks have exhibited exceptional ability in capturing complex underlying structures of data and learning discriminative features for clustering [134, 135, 136, 137]. Inspired by that, a new line of research has been established to bridge

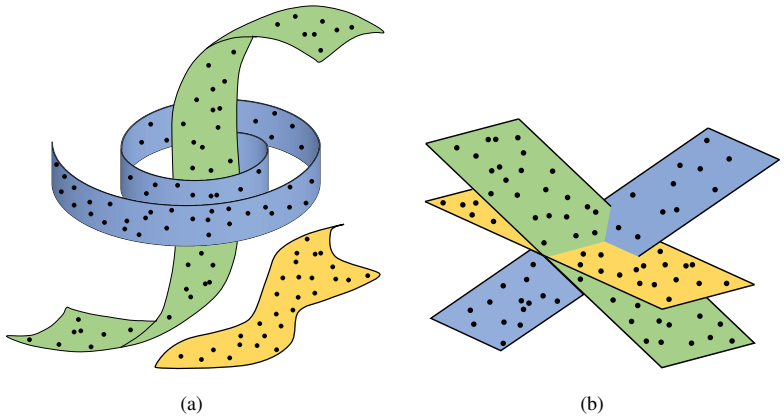

Figure 3.1: Illustration of representation learning for subspace clustering. (a) Sample points may come from a union of nonlinear subspaces; (b) Deep subspace clustering approaches aim to transform the samples into a latent space so that they lie in a union of linear subspaces.

deep learning and subspace clustering for developing deep subspace clustering approaches [35, 41, 42, 43]. Variational Autoencoders (VAE) [44, 45] and Generative Adversarial Network (GAN) [46] are among the most popular deep architectures adopted by these methods to produce feature representations suitable for subspace clustering [45]. Compared to the conventional approaches, deep subspace clustering approaches are able to better exploit the non-linear relationships between the sample points and consequently they achieve higher performance, especially in complex applications in which the samples do not necessarily satisfy the self-expressiveness property [35].

In this chapter, we propose a novel spectral clustering-based approach which utilizes stacked convolutional autoencoders to tackle the problem of subspace clustering. Inspired by the idea of residual networks, our first contribution is to add multiple fully-connected linear layers between the corresponding layers of the encoder and decoder to infer multi-level representations from the output of every encoder layer. These connection layers enable to produce representations which are enforced to satisfy self-expressiveness property and hence well-suited to subspace clustering. We model each connection layer as a self-

expression matrix created from the summation of a coefficient matrix shared between all layers and a layer-specific matrix that captures the unique knowledge of each individual layer. Moreover, we introduce a novel loss function that utilizes an initial clustering of the samples and efficiently aggregates the information at different levels to infer the coefficient matrix and the layer-specific matrices more accurately. This loss function is further minimized in an iterative scheme which alternatively updates the network parameters for learning better subspace clustering representations and produces a new clustering of the samples. We perform extensive experiments on four benchmark datasets for subspace clustering, including two face image and two object image datasets, to evaluate the efficacy of the proposed method. The experiments demonstrate that our approach can efficiently handle clustering the data from non-linear subspaces and it performs better than the state-of-the-art methods on most of the subspace clustering problems.

3.2 Related Works

Conventional subspace clustering approaches aim to learn a weighted graph whose edge weights represent the relationships between the samples of input data. Then, spectral clustering [125] (or its variants [138]) can be employed to partition the graph into a set of disjoint sub-graphs corresponding to different clusters [139, 140, 34, 126, 141, 142, 127, 143, 144]. A commonly-used formulation to obtain such a weighted graph is written as

$$\underset{\boldsymbol{C}\in\mathbb{R}^{n\times n}}{\text{minimize}} \qquad \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{C}\|_{\mathrm{F}}^{2}+\lambda\, g(\boldsymbol{C}) \tag{3.1a}$$

$$\text{subject to} \qquad \operatorname{diag}(\boldsymbol{C})=\boldsymbol{0}, \tag{3.1b}$$

where $\|.\|_{\mathrm{F}}$ indicates Frobenius norm, $\boldsymbol{X}\in\mathbb{R}^{d\times n}$ is a data matrix with its columns representing the samples $\{\boldsymbol{x}_i\in\mathbb{R}^d\}_{i=1}^n$, $\boldsymbol{C}$ is a self-expression matrix with its $(i,j)^{th}$ element denoting the contribution of sample $\boldsymbol{x}_j$ in reconstructing $\boldsymbol{x}_i$, $g:\mathbb{R}^{n\times n}\rightarrow\mathbb{R}$ is a certain regularization function, and $\lambda>0$ is a hyperparameter to balance the importance of the

terms. Equality constraint (3.1b) is imposed to eliminate the trivial solution $\boldsymbol{C} = \boldsymbol{I}_n$ that represents a point as a linear combination of itself. Once the optimal solution $\overset{*}{\boldsymbol{C}}$ of (3.1a) – (3.1b) is obtained, symmetric matrix $\frac{1}{2}(|\overset{*}{\boldsymbol{C}}| + |\overset{*}{\boldsymbol{C}}|^\top)$ can serve as the affinity matrix of the desired graph where $|.|$ shows the element-wise absolute value operator. Different variants of (3.1a) – (3.1b) have been well-studied in the literature where they utilize various choices of the regularization function $g(.)$ such as $\|\boldsymbol{C}\|_0$ [145, 132], $\|\boldsymbol{C}\|_1$ [34], $\|\boldsymbol{C}\|_*$ [128, 129], $\|\boldsymbol{C}\|_{\mathrm{F}}$ [130], etc, to impose desired structures on the graph.

Deep generative architectures, most notably GANs and VAEs, have been widely used in the recent literature to facilitate the clustering task [146], especially when the samples come from complex and irregular distributions [45, 137]. These architectures improve upon the conventional feature extractions by learning more informative and discriminative representations that are highly suitable for clustering [147, 148, 41]. To promote inferring clusters with higher quality, some deep approaches propose to jointly learn the representations and perform clustering in a unified framework [146, 149, 150, 43]. One successful deep approach to the subspace clustering problem is presented in [35], known as Deep Subspace Clustering (DSC), which employs a deep convolutional auto-encoder to learn latent representations and uses a novel self-expressive layer to enforce them to lie on a union of linear subspaces. The DSC model is further adopted by Deep Adversarial Subspace Clustering (DASC) method [43] to develop an adversarial architecture, consisting of a generator to produce subspaces and a discriminator to supervise the generator by evaluating the quality of the subspaces. More recently, [150] introduced an end-to-end trainable framework, named Self-Supervised Convolutional Subspace Clustering Network (S^2ConvSCN), which aims to jointly learn feature representations, self-expression coefficients, and the clustering results to produce more accurate clusters.

Our approach can be seen as a generalization of the DSC algorithm [35] to the case that low-level and high-level information of the input data is utilized to produce more in-

formative and discriminative subspace clustering representations. Moreover, we introduce a loss minimization problem that employs an initial clustering of the samples to effectively aggregate the knowledge gained from multi-level representations and to promote learning more accurate subspaces. Notice that although our work is close to DASC [43] and S^{2}ConvSCN [150] in the sense that it leverages a clustering of the samples to improve the feature learning procedure, we adopt a completely different strategy to incorporate the pseudo-label information into the problem.

It is noteworthy to emphasize that our approach may seem similar to the multi-view subspace clustering approaches [151, 128, 152, 153] as it aggregates information obtained from multiple modalities of the data to recover the clusters more precisely. However, it differs from them in the sense that our method leverages some connection layers to simultaneously learn multi-level deep representations and effectively fuse them to boost the clustering performance.

3.3 Problem Formulation

Let $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$ be a set of n sample points drawn from a union of K different subspaces in $\mathbb{R}^d$ that are not necessarily linear. An effective approach to cluster the samples is to transform them into a set of new representations that have linear relationships and satisfy the self-expressiveness property. Then, spectral clustering can be applied to recover the underlying clusters. To this end, the DSC algorithm [35] introduced a deep architecture consisting of a convolutional autoencoder with L layers to generate latent representations and a fully-connected linear layer inserted between the encoder and decoder to ensure the self-expressiveness property is preserved. Let $\mathcal{E}$ and $\mathcal{D}$, parameterized by Θ_e and Θ_d, denote the encoder and the decoder networks, respectively. Given that, the DSC algorithm

proposed to solve the following optimization problem to learn desired representations and infer self-expression matrix $\boldsymbol{C}$

$$\underset{\Theta}{\text{minimize}} \quad \|\boldsymbol{X}-\hat{\boldsymbol{X}}_{\Theta}\|_{\mathrm{F}}^{2}+\lambda\|\boldsymbol{Z}_{\Theta_e}-\boldsymbol{Z}_{\Theta_e}\boldsymbol{C}\|_{\mathrm{F}}^{2}+\gamma\|\boldsymbol{C}\|_{p} \tag{3.2a}$$

$$\text{subject to} \quad \operatorname{diag}(\boldsymbol{C})=\boldsymbol{0}, \tag{3.2b}$$

where $\lambda, \gamma > 0$ are fixed hyperparameters to control the importance of different terms and $\Theta = \{\Theta_e, \boldsymbol{C}, \Theta_d\}$ shows the network parameters. Matrix $\boldsymbol{Z}_{\Theta_e} \in \mathbb{R}^{\bar{d}\times n}$ indicates the latent representations where $\bar{d}$ is the dimension of the representations and $\boldsymbol{Z}_{\Theta_e} = \mathcal{E}(\boldsymbol{X};\Theta_e)$, and matrix $\hat{\boldsymbol{X}}_{\Theta} \in \mathbb{R}^{d\times n}$ denotes the reconstructed samples where $\hat{\boldsymbol{X}}_{\Theta} = \mathcal{D}(\mathcal{E}(\boldsymbol{X};\Theta_e)\boldsymbol{C};\Theta_d)$. The main goal of problem (3.2a) – (3.2b) is to compute the network parameters such that equality $\boldsymbol{Z}_{\Theta_e} = \boldsymbol{Z}_{\Theta_e}\boldsymbol{C}$ holds and the reconstructed matrix $\hat{\boldsymbol{X}}$ can well approximate the input data $\boldsymbol{X}$. [35] used the backpropagation technique followed by the spectral clustering algorithm to find the solution of the minimization problem (3.2a) – (3.2b) and determine the cluster memberships of the samples.

In what follows, we propose a new deep architecture that leverages information from different levels of the encoder to learn more informative representations and improve the subspace clustering performance.

3.4 Proposed Method

This section presents a detailed explanation of our proposed approach. As it can be seen from the problem (3.2a) – (3.2b), the DSC algorithm only relies on the latent variables $\boldsymbol{Z}_{\Theta_e}$ to perform clustering. Due to the fact that different layers of the encoder construct increasingly complex representations of the input data, it may be quite difficult to learn suitable subspace clustering representations from the output of the last layer of the encoder. This provides a strong motivation to incorporate information from the early layers of the encoder to boost the clustering performance. Towards this goal, our approach uses a new

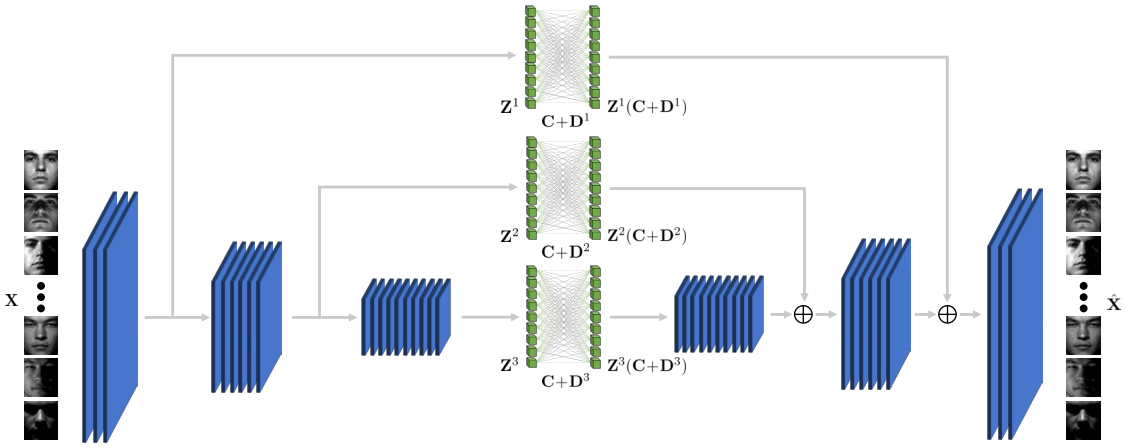

Figure 3.2: Architecture of the proposed multi-level representation learning model for $L = 3$. Observe that the representations learned at different levels of the encoder are fed into fully-connected linear layers to be used in the reconstruction procedure. Such strategy enables to combine low-level information from the early layers with high-level information from the deeper layers to produce more informative and robust subspace clustering representations. Each fully-connected layer is associated with a self-expression matrix formed from the summation of a coefficient matrix $\boldsymbol{C}$ shared between all layers and a distinctive matrix $\boldsymbol{D}^l$, $l \in \{1, \ldots, L\}$, which captures the unique information of each individual layer.

architecture which jointly benefits from low-level and high-level information to learn more informative subspace clustering representations. The approach adds fully-connected linear layers between the symmetrical layers of the encoder and the decoder to provide multiple paths of information flow through the network. These connection layers can not only considerably improve the ability of the network in extracting more complex and informative features but also promote learning self-expressive representations from the output of every encoder layer. Figure 4.2 illustrates the proposed architecture in detail. Observe that the multi-level representations learned at different layers of the encoder, denoted as $\{\boldsymbol{Z}^l_{\Theta_e}\}_{l=1}^{L}$, are input to the fully-connected linear layers and the outputs of these layers are fed into the decoder layers. This strategy allows the decoder to reuse the low-level information for producing more accurate reconstructions of the input data which in turn can improve the overall clustering performance.

We assume each fully-connected layer is associated with a self-expression matrix in the form of the summation of two matrices, where the first one is shared between the entire layers and the second one is a layer-specific matrix. Considering the encoder as a mapping function from the input space to the representation space, it aims to preserve the relations between the data samples at different levels of representations. Moreover, some samples may have stronger (or weaker) relations in different levels of the encoder. Define $\boldsymbol{C} \in \mathbb{R}^{n \times n}$ as the consistency matrix to capture the relational information shared between the encoder layers and $\{\boldsymbol{D}^l\}_{l=1}^{L} \in \mathbb{R}^{n \times n}$ as distinctive matrices to produce the unique information of the individual layers. Given that, we incorporate the following loss function to promote learning self-expressive representations

$$\mathcal{L}_{exp} = \sum_{l=1}^{L} \| \boldsymbol{Z}_{\Theta_e}^{l} - \boldsymbol{Z}_{\Theta_e}^{l} (\boldsymbol{C} + \boldsymbol{D}^l) \|_{\mathrm{F}}^2. \tag{3.3}$$

The above formulation is able to simultaneously model the shared information across different levels while considering the unique knowledge gained from each individual layer. This property allows to effectively leverage the information from the representations learned at multiple levels of the encoder and therefore is also particularly well-suited to the problem of multi-view subspace clustering [128].

The self-expression loss $\mathcal{L}_{exp}$ is employed to promote learning self-expressive feature representations at different levels of the encoder. To better accomplish this purpose, it is beneficial to adopt certain matrix norms for imposing desired structures on the elements of the distinctive matrices $\{\boldsymbol{D}^l\}_{l=1}^{L}$ and the consistency matrix $\boldsymbol{C}$. For the distinctive matrices, we use Frobenius norm to ensure the connectivity of the affinity graph associated with each fully-connected layer. For the consistency matrix $\boldsymbol{C}$, we employ ℓ_1-norm to learn sparse representations of the data. Ideally, it is desired to infer matrix $\boldsymbol{C}$ such that each sample $\boldsymbol{x}_i$ is only expressed by a linear combination of the samples belonging to the same subspace as

$\boldsymbol{x}_i$. To ensure the consistency matrix and the distinctive matrices obey the aforementioned desired structures, we propose to incorporate the following regularization terms

$$\mathcal{L}_{\boldsymbol{C}} = \|\boldsymbol{Q}^\top|\boldsymbol{C}|\|_1, \qquad \mathcal{L}_{\boldsymbol{D}} = \sum_{l=1}^{L}\|\boldsymbol{D}^l\|_{\mathrm{F}}^2, \tag{3.4}$$

where $\|.\|_1$ computes the sum of absolute values of a matrix, $\boldsymbol{Q} \in \mathbb{R}^{n\times K}$ is a membership matrix with its columns are one-hot vectors denoting the pseudo-labels assigned to the samples. The regularization term $\mathcal{L}_{\boldsymbol{C}}$ is adopted to incorporate the information gained from the pseudo-labels into the model. Unlike the commonly used regularization $\|\boldsymbol{C}\|_1$ which imposes sparsity on the entire elements of matrix $\boldsymbol{C}$, our regularization promotes sparsity on the cluster memberships of the samples. In other words, it encourages to assign each sample into a single subspace and only use the samples of the subspace to reconstruct the given sample. Moreover, the regularization term $\mathcal{L}_{\boldsymbol{D}}$ promotes the elements of the distinctive matrices to be similar in value, which in turn can enhance the connectivity of the affinity graph associated with each fully-connected layer.

Combining the loss function (3.3) and the regularization terms $\mathcal{L}_{\boldsymbol{C}}$ and $\mathcal{L}_{\boldsymbol{D}}$ together with the reconstruction loss $\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_{\mathrm{F}}^2$ leads to the following optimization problem that needs to be solved for training our proposed model

$$\begin{aligned}
\underset{\Theta\cup\{\boldsymbol{D}^l\}_{l=1}^L}{\text{minimize}} \quad & \|\boldsymbol{X} - \hat{\boldsymbol{X}}_\Theta\|_{\mathrm{F}}^2 + \lambda_1 \sum_{l=1}^{L}\|\boldsymbol{Z}_{\Theta_e}^l - \boldsymbol{Z}_{\Theta_e}^l(\boldsymbol{C}+\boldsymbol{D}^l)\|_{\mathrm{F}}^2 \\
& \lambda_2\|\boldsymbol{Q}^\top|\boldsymbol{C}|\|_1 + \lambda_3 \sum_{l=1}^{L}\|\boldsymbol{D}^l\|_{\mathrm{F}}^2 \qquad (3.5a) \\
\text{subject to} \quad & \operatorname{diag}(\boldsymbol{C}+\boldsymbol{D}^l) = \boldsymbol{0}, \qquad l \in \{1,\ldots,L\}, \qquad (3.5b)
\end{aligned}$$

where $\lambda_1, \lambda_2, \lambda_3 > 0$ are hyperparameters to balance the contribution of different losses. We adopt the standard backpropagation technique to obtain the solution of problem (3.5a) –

(3.5b). Once the solution matrices $\overset{*}{\boldsymbol{C}}$ and $\{\overset{*}{\boldsymbol{D}}^l\}_{l=1}^L$ are obtained, we can create symmetric affinity matrix $\boldsymbol{W} \in \mathbb{S}_n$ of the following form

$$\boldsymbol{W} = \frac{\left|\overset{*}{\boldsymbol{C}} + \frac{1}{L}\sum_{l=1}^{L} \overset{*}{\boldsymbol{D}}^l\right|}{2} + \frac{\left|\overset{*}{\boldsymbol{C}}^\top + \frac{1}{L}\sum_{l=1}^{L} \overset{*}{\boldsymbol{D}}^{l\top}\right|}{2}. \tag{3.6}$$

Then, the spectral clustering algorithm can be applied on matrix $\boldsymbol{W}$ to recover the underlying subspaces and cluster the samples to their respective subspaces.

Note that the pseudo-labels generated by the spectral clustering method can be leveraged to retrain the model and provide a more precise estimation of the subspaces. Motivated by that, we develop an iterative scheme which starts from a membership matrix $\boldsymbol{Q}$ (or equivalently an initial clustering of the input data) and alternatively runs the model for T epochs to train the network parameters $\Theta \cup \{\boldsymbol{D}^l\}_{l=1}^L$ and then updates the membership matrix. This training procedure is then repeated until the convergence is obtained. Different steps of our proposed scheme are described and depicted in detail in Algorithm 2.

Algorithm 2 Proposed Subspace Clustering Approach

Input: $\boldsymbol{X}$, $\boldsymbol{Q}$, T, $k = 1$

1: **repeat**
2: Update network parameters $\Theta \cup \{\boldsymbol{D}^l\}_{l=1}^L$
3: **if** $k \bmod T = 0$ **then**
4: Create affinity matrix $\boldsymbol{W}$
5: Apply spectral clustering to update $\boldsymbol{Q}$
6: **end if**
7: $k \leftarrow k + 1$
8: **until** $k \leq \mathrm{maxIter}$

Output: $\boldsymbol{Q}$

It can be seen that given the input, Algorithm 2 is able to train the network parameters $\Theta \cup \{\boldsymbol{D}^l\}_{l=1}^L$ from scratch. However, several aspects of the algorithm such as convergence

behavior and accuracy can be considerably improved by employing pre-trained models and using fine-tuning techniques to obtain initial values for the encoder and the decoder networks [35].

In the next section, we perform extensive experiments to corroborate the effectiveness of our approach. Also, we present a detailed explanation about the parameter settings, the pre-trained models, and the fine-tuning procedures used in our experiments.

3.5 Experiments

This section evaluates the clustering performance of our proposed method, termed MLRDSC, on four standard benchmark datasets for subspace clustering including two face image datasets (ORL and Extended Yale B) and two object image datasets (COIL20 and COIL100). Sample images from each of the datasets are illustrated in Figure 4.3. We perform multiple subspace clustering experiments on the datasets and compare the results against some baseline algorithms, including Low Rank Representation (LRR) [127], Low Rank Subspace Clustering (LRSC) [129], Sparse Subspace Clustering (SSC) [34], SSC with the pre-trained convolutional auto-encoder features (AE+SSC), Kernel Sparse Subspace Clustering (KSSC) [37], SSC by Orthogonal Matching Pursuit (SSC-OMP) [132], Efficient Dense Subspace Clustering (EDSC) [154], EDSC with the pre-trained convolutional auto-encoder features (AE+EDSC), Deep Subspace Clustering (DSC) [35], and Deep Adversarial Subspace Clustering (DASC) [43], Self-Supervised Convolutional Subspace Clustering Network (S^2ConvSCN) [150]. For the competitor methods, we directly collect the scores from the corresponding papers and some existing literature [35, 150].

Note that the subspace clustering problem is regarded as a specific clustering scenario which seeks to cluster a set of given unlabeled samples into a union of low-dimensional subspaces that best represent the sample data. In this sense, the subspace clustering approaches

(a) Extended Yale B (b) ORL (c) COIL20 and COIL100

Figure 3.3: Example images of Extended Yale B, ORL, COIL20, and COIL100 datasets. The main challenges in the face image datasets, Extended Yale B and ORL, are illumination changes, pose variations and facial expression variations. The main challenges in the object image datasets, COIL20 and COIL100, are the variations in the view-point and scale.

are basically different from the standard clustering methods that aim to group the samples around some cluster centers. Most of the subspace clustering literature revolves around using the spectral clustering technique to recover underlying subspaces from an affinity graph created over the entire samples. This can considerably increase the computational cost of these methods in comparison to the standard clustering approaches. As a consequence of this limitation, the benchmark datasets used for subspace clustering are generally smaller than that for the clustering task. In this work, we perform experiments on the aforementioned four datasets which are frequently used in the recent literature [34, 35, 150, 43] to evaluate the performance of the subspace clustering approaches.

In what follows, we first describe the training procedure used in our experiments. Then, we provide more details for each individual dataset and present the final clustering results of different methods.

3.5.1 Training Procedure

Following the literature [35, 43], for the convolutional layers, we use kernel filters with stride 2 in both dimensions and adopt Rectified Linear Unit (ReLU) activation function. For the fully-connected layers, we use linear weights without considering bias or

Table 3.1: Clustering error (%) of different methods on Extended Yale B dataset. The best results are in bold.

Measure	LRR	LRSC	SSC	AE+SSC	KSSC	EDSC	AE+EDSC	DSC	S^{2}ConvSCN	MLRDSC
10 subjects										
Mean	22.22	30.95	10.22	17.06	14.49	5.64	5.46	1.59	1.18	**1.10**
Median	23.49	29.38	11.09	17.75	15.78	5.47	6.09	1.25	1.09	**0.94**
15 subjects										
Mean	23.22	31.47	13.13	18.65	16.22	7.63	6.70	1.69	1.12	**0.91**
Median	23.49	31.64	13.40	17.76	17.34	6.41	5.52	1.72	1.14	**0.99**
20 subjects										
Mean	30.23	28.76	19.75	18.23	16.55	9.30	7.67	1.73	1.30	**0.99**
Median	29.30	28.91	21.17	16.80	17.34	10.31	6.56	1.80	1.25	**1.02**
25 subjects										
Mean	27.92	27.81	26.22	18.72	18.56	10.67	10.27	1.75	1.29	**1.13**
Median	28.13	26.81	26.66	17.88	18.03	10.84	10.22	1.81	1.28	**1.12**
30 subjects										
Mean	37.98	30.64	28.76	19.99	20.49	11.24	11.56	2.07	**1.67**	1.78
Median	36.82	30.31	28.59	20.00	20.94	11.09	10.36	2.19	1.72	**1.41**
35 subjects										
Mean	41.85	31.35	28.55	22.13	26.07	13.10	13.28	2.65	1.62	**1.44**
Median	41.81	31.74	29.04	21.74	25.92	13.10	13.21	2.64	1.60	**1.47**
38 subjects										
Mean	34.87	29.89	27.51	25.33	27.75	11.64	12.66	2.67	1.52	**1.36**
Median	34.87	29.89	27.51	25.33	27.75	11.64	12.66	2.67	1.52	**1.36**

non-linear activation function. In order to train the model and obtain the affinity matrix, we follow the literature [35, 150, 43] and pass the entire samples into the model as a single batch. The Adam optimizer [155] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and learning rate 0.001 is used to train the network parameters. All experiments are implemented in PyTorch.

As it is noted in [35], it is computationally very expensive to train the model from scratch since the samples are fed into the network in a single batch. Therefore, we follow [35] to produce a pre-trained model by learning a model obtained from shortcutting the connection layers and ignoring the self-expression loss term $\mathcal{L}_{exp}$. The resulting model is trained on the entire sample points and it can be further utilized to initialize the encoder and the decoder parameters of our proposed architecture. Moreover, through the entire experiments on the datasets, we initialize the membership matrix $\boldsymbol{Q}$ with a zero matrix and set each of the individual matrices $\boldsymbol{C}$ and $\{\boldsymbol{D}^l\}_{l=1}^{L}$ to a matrix with all elements 0.0001.

3.5.2 Results

The results of all experiments are reported based on the clustering error which is defined to be the percentage of the misclustered samples to the entire sample points.

Extended Yale B: This dataset is used as a popular benchmark for the subspace clustering problem. It consists of 2432 frontal face images of size 192×168 captured from 38 different human subjects. Each subject has 64 images taken under different illumination conditions and poses. For computational purposes and following the literature [34, 35, 150], we downsample the entire images from their original size to 48×42.

We perform multiple experiments for a different number of human subjects $K \in \{10, 15, 20, 25, 30, 35, 38\}$ of this dataset to evaluate the sensitivity of MLRDSC with respect to increasing the number of clusters. By numbering the subjects from 1 to 38, we perform experiments on all possible K consecutive subjects and present the mean and median clustering errors of each $39 - K$ trials. Such experiments have been frequently performed in the literature [34, 35, 43, 150]. Through these experiments, we have employed an autoencoder model consisting of three stacked convolutional encoder layers with 10, 20, and 30 filters with sizes 5×5, 3×3, and 3×3, respectively. The parameters used in the experiments on this dataset are as follows: $\lambda_1 = 1 \times 10^{\frac{K}{10}-1}$, $\lambda_2 = 40$, $\lambda_3 = 10$, and we update the membership matrix $\boldsymbol{Q}$ in every $T = 100$ consecutive epochs. For the entire choices of K, we set the maximum number of epochs to 900.

The clustering results on this dataset are reported in Table 3.1. Observe that MLRDSC achieves smaller errors than the competitor methods in all experiments, except for the mean of clustering error in case $K = 30$.

ORL: This dataset consists of 400 face images of size 112×92 from 40 different human subjects where each subject has 10 images taken under diverse variation of poses, lighting conditions, and facial expressions. Following the literature, we downsample the images from their original size to 32×32. This dataset is challenging for subspace clustering due

Table 3.2: Clustering error (%) of different methods on ORL, COIL20, and COIL100 datasets. The best results are in bold.

Dataset	LRR	LRSC	SSC	AE+SSC	KSSC	EDSC	AE+EDSC	DSC	DASC	S^{2}ConvSCN	Ours
ORL	33.50	32.50	29.50	26.75	34.25	27.25	26.25	14.00	11.75	**10.50**	11.25
COIL20	30.21	31.25	14.83	22.08	24.65	14.86	14.79	5.42	3.61	2.14	**2.08**
COIL100	53.18	50.67	44.90	43.93	47.18	38.13	38.88	30.96	–	26.67	**23.28**

to the large variation in the appearance of facial expressions (shown in Figure 4.3) and since the number of images per each subject is quite small.

Through the experiment on ORL, we have adopted a network architecture consisting of three convolutional encoder layers with 3, 3, and 5 filters, all of size 3×3. Moreover, the parameter settings used in the experiment are as follows: $\lambda_1 = 5$, $\lambda_2 = 0.5$, $\lambda_3 = 1$, $T = 10$, and the maximum number of epochs is set to 420.

The results of this experiment are presented in Table 3.2. It can be seen that MLRDSC outperforms all the competitor methods, except S^{2}ConvSCN which attains the smallest clustering error rate on ORL.

COIL20/COIL100: These two datasets are widely used for different types of clustering. COIL20 contains 1440 images captured from 20 various objects and COIL100 has 7200 images of 100 objects. Each object in either of the datasets has 72 images with black background taken at pose intervals of 5 degrees. The large viewpoint changes can pose serious challenges for the subspace clustering problem on these two datasets (Shown in in Figure 4.3).

For COIL20 and COIL100 datasets, the literature methods [35, 150, 43] mostly adopt one layer convolutional autoencoders to learn feature representations. This setting admits no connection layer and hence is not well-suited to our approach. To better demonstrate the advantages of MLRDSC, we use a two-layer convolutional autoencoder model with 5 and 10 filters for performing experiment on COIL20 and adopt the same architecture with 20 and 30 filters for COIL100. The entire filters used in both experiments are of size 3×3.

Table 3.3: Ablation study of our method in terms of clustering error (%) on Extended Yale B. The best results are in bold.

Measure	DSC-L2	DSC-L1	MLRDSC ($\|\boldsymbol{C}\|_1$)	MLRDSC
10 subjects				
Mean	1.59	2.23	**1.09**	1.10
Median	1.25	2.03	1.08	**0.94**
15 subjects				
Mean	1.69	2.17	0.98	**0.91**
Median	1.72	2.03	**0.99**	**0.99**
20 subjects				
Mean	1.73	2.17	**0.94**	0.99
Median	1.80	2.11	**0.94**	1.02
25 subjects				
Mean	1.75	2.53	**1.13**	**1.13**
Median	1.81	2.19	**1.12**	**1.12**
30 subjects				
Mean	2.07	2.63	1.84	**1.78**
Median	2.19	2.81	**1.35**	1.41
35 subjects				
Mean	2.65	3.09	1.49	**1.44**
Median	2.64	3.10	1.49	**1.47**
38 subjects				
Mean	2.67	3.33	1.40	**1.36**
Median	2.67	3.33	1.40	**1.36**

Moreover, the parameter settings for the datasets are as follows: 1) COIL20: $\lambda_1 = 20$, $\lambda_2 = 20$, $\lambda_3 = 5$, $T = 5$, and the maximum number of epochs is set to 50; 2) COIL100: $\lambda_1 = 20$, $\lambda_2 = 40$, $\lambda_3 = 10$, $T = 50$, and the maximum number of epochs is set to 350.

The results on COIL20 and COIL100 datasets are shown in Table 3.2. Observe that our approach achieves better subspace clustering results on both datasets compared to the existing state-of-the-art methods.

According to the (3.1) – (3.2), the deep subspace clustering methods, such as DSC, S^{2}ConvSCN, and MLRDSC, perform considerably well compared to the classical subspace clustering approaches on the benchmark datasets. This success can be attributed to the fact that deep models are able to efficiently capture the non-linear relationships between the samples and recover the underlying subspaces. Moreover, the results indicate that MLRDSC outperforms the DSC algorithm by a notable margin. This improvement result from

the incorporation of a modified regularization term and the insertion of connection layers between the corresponding layers of the encoder and decoder. These layers enable to combine the information of different levels of the encoder to learn more favorable subspace clustering representations. It is noteworthy to mention that although our approach achieves better clustering results than the DSC method, it has more parameters to train, which in turn increases the computational burden of the model.

Ablation Study: To highlight the benefits brought by different components of our proposed model, we carry out an ablation study by evaluating a variant of our approach, named MLRDSC($\|\boldsymbol{C}\|_1$), which replaces the regularization term $\|\boldsymbol{Q}|\boldsymbol{C}|\|_1$ with term $\|\boldsymbol{C}\|_1$. In this sense, MLRDSC($\|\boldsymbol{C}\|_1$) can be seen as a generalization of DSC-L1 (a variant of the DSC algorithm that utilizes regularization term $\|\boldsymbol{C}\|_1$ [35]) to a case that leverages multiple connection layers to learn multi-level subspace clustering representations. We perform experiments for different number of subjects K on the Extended Yale B dataset and present the clustering results in Table 3.3. As the table indicates, inserting the connection layers between the symmetrical layers of the encoder and decoder can considerably improve the clustering performance of DSC-L1 algorithm. Moreover, comparing the results of MLRDSC and MLRDSC($\|\boldsymbol{C}\|_1$) confirms the positive effect of incorporating the regularization term $\|\boldsymbol{Q}|\boldsymbol{C}|\|_1$.

3.6 Conclusions

This chapter presented a novel spectral clustering-based approach which uses a deep neural network architecture to address the subspace clustering problem. The proposed method improves upon the existing deep approaches by leveraging information exploited from different levels of the networks to transform input samples into multi-level representations lying on a union of linear subspace. Moreover, it is able to use pseudo-labels

generated by the spectral clustering technique to effectively supervise the representation learning procedure and boost the final clustering performance. Experiments on benchmark datasets demonstrate that the proposed approach is able to efficiently handle clustering from the non-linear subspaces and it achieves better results compared to the state-of-the-art methods.

CHAPTER 4

Class Conditional Alignment for Partial Domain Adaptation

Adversarial adaptation models have demonstrated significant progress towards transferring knowledge from a labeled source dataset to an unlabeled target dataset. Partial domain adaptation (PDA) investigates the scenarios in which the source domain is large and diverse, and the target label space is a subset of the source label space. The main purpose of PDA is to identify the shared classes between the domains and promote learning transferable knowledge from these classes. In this paper, we propose a multi-class adversarial architecture for PDA. The proposed approach jointly aligns the marginal and class-conditional distributions in the shared label space by minimaxing a novel multi-class adversarial loss function. Furthermore, we incorporate effective regularization terms to encourage selecting the most relevant subset of source domain classes. In the absence of target labels, the proposed approach is able to effectively learn domain-invariant feature representations, which in turn can enhance the classification performance in the target domain. Comprehensive experiments on two benchmark datasets Office-31 and Office-Home corroborate the effectiveness of the proposed approach in addressing different partial transfer learning tasks.

4.1 Introduction

With the impressive power of learning representations, deep neural networks have shown superior performance in a wide variety of machine learning tasks such as classification [47, 48, 49], semantic segmentation [50, 51, 52], object detection [53, 49, 54], etc. These notable achievements heavily depend on the availability of large amounts of

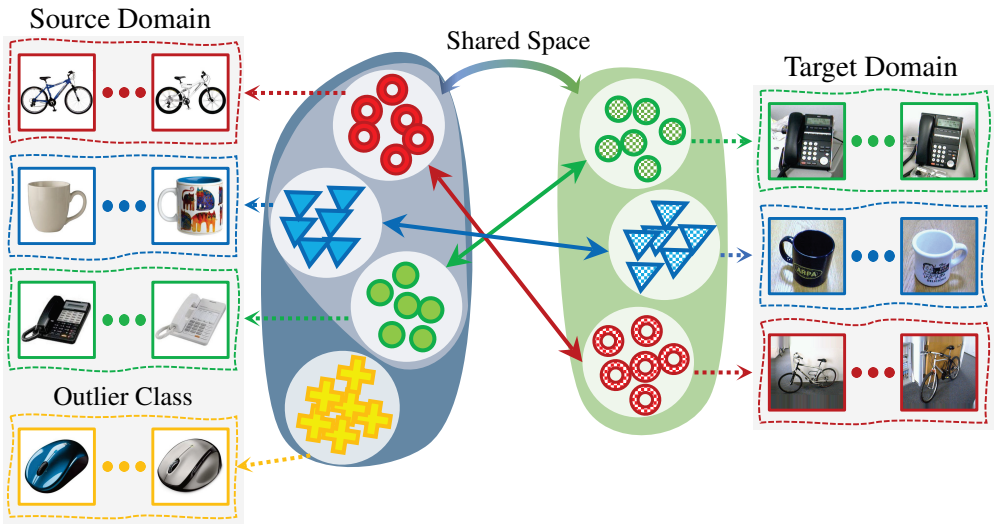

Figure 4.1: Illustration of partial domain adaptation task. The objective is to transfer knowledge between the shared classes in the source and target domains. To this end, it is desired to identify and reject the outlier source classes and align both marginal and class-conditional distributions across the shared label space. *Best viewed in color.*

labeled training data. However, in many applications, collecting sufficient labeled data is either difficult or time-consuming. One potential solution to reduce the labeling consumption is to build an effective predictive model using readily-available labeled data from a different but related source domain. Such a learning paradigm generally suffers from the distribution shift between the source and target domains, which in turn poses a significant difficulty in adapting the predictive model to the target domain tasks. In the absence of target labels, unsupervised domain adaptation (UDA) seeks to enhance the generalization capability of the predictive model by learning feature representations that are discriminative and domain-invariant [55, 56, 57]. Various approaches have been proposed in the literature to tackle UDA problems by embedding domain adaptation modules in deep architectures [58, 59, 60, 61, 62, 63] (see [64] for a comprehensive survey on deep domain adaptation methods). A line of research is developed to align the marginal distributions of the source and target domains through minimizing discrepancy measures such as maxi-

mum mean discrepancy [156, 60], central moment discrepancy [157], correlation distance [158, 159], etc. In this way, they can map both domains into the same latent space, which results in learning domain-invariant feature representations. Another strand of research is focused on designing specific distribution normalization layers which facilitates learning separate statistics for the source and target domains [160, 161]. More recently, some research studies have been carried out based on the generative adversarial networks [46] that aim to alleviate the marginal disparities across the domains by adversarially learning domain-invariant feature representations which are indistinguishable for a discriminative domain classifier [65, 66, 67].

Despite the efficacy of the existing UDA methods, their superior performance is mostly limited to the scenarios in which the source and target domains share the same set of labels. With the goal of considering more realistic and practical cases, [1] introduced partial domain adaptation (PDA) as a new adaptation scenario in which the target label space is a subset of the source label space. The main challenge in PDA is to identify and reject the source domain classes that do not appear in the target domain, known as *outlier classes*, mainly because they may exert negative impacts on the overall transfer performance [70, 4]. Addressing this challenge enables the PDA methods to transfer models trained on large and diverse labeled datasets (e.g. ImageNet) to small-scale unlabeled datasets from different but related domains.

In this paper, we propose a novel adversarial approach for partial domain adaptation which seeks to automatically reject the outlier source classes and improve the classification confidence on *irrelevant samples*, i.e. the samples that are highly dissimilar across the domains. The existing PDA methods often align the marginal distributions between the domains in the shared label space. Different from these methods, we propose a novel adversarial architecture that matches class-conditional feature distributions by minimaxing a multi-class adversarial loss function. Moreover, we propose to boost the target domain

classification performance by incorporating two novel regularization functions. The first regularizer is a row-sparsity term on the output of the classifier to promote the selection of a small subset of classes that are in common between the source and target domains. The second one is a minimum entropy term which increases the classifier confidence level in predicting the labels of irrelevant samples from both domains. We empirically observe that our proposed approach considerably improves the state-of-the-art performance for various partial domain adaptation tasks on two commonly-used benchmark datasets Office-31 and Office-Home.

4.2 Related Work

To date, various unsupervised domain adaptation (UDA) methods have been developed to learn domain-invariant feature representations in the absence of target labels. Some studies have proposed to minimize the maximum mean discrepancy between the features extracted from the source and target samples [162, 60, 163, 164, 63]. In [165], a correlation alignment (CORAL) method is developed that utilizes a linear transformation to match the second-order statistics between the domains. [159] presented an extension of the CORAL method that learns a non-linear transformation to align the correlations of layer activations in deep networks. Despite the practical success of the aforementioned methods in domain alignment, it is shown that they are unable to completely eliminate the domain shift [59, 58]. Another line of work has proposed to reduce the discrepancy by learning separate normalization statistics for the source and target domains [160, 161]. [160] adopts different batch normalization layers for each domain to align the marginal distributions. [161] embeds domain alignment layers at different levels of a deep architecture to align the domain feature distributions to a canonical one.

More recently, adversarial adaptation methods have been extensively investigated to boost the performance of UDA methods [68, 166, 167, 69, 65]. The basic idea behind these methods is to train a discriminative domain classifier for predicting domain labels and a deep network for learning feature representations that are indistinguishable by the discriminator. By doing so, the marginal disparities between the source and target domains can be efficiently reduced, which results in significant improvement in the overall classification performance [68, 69, 66]. Transferable attention for domain adaptation [67] proposed an adversarial attention-based mechanism for UDA, which effectively highlights the transferable regions or images. [168] introduced an incremental adversarial scheme which gradually reduces the gap between the domain distributions by iteratively selecting high confidence pseudo-labeled target samples to enlarge the training set. While the existing UDA models have shown tremendous progress towards reducing domain discrepancy, they mostly rely on the assumption of fully shared label space and generally align the marginal feature distributions between the source and target domains. This assumption is not necessarily valid in partial domain adaptation (PDA) which assumes the target label space is a subset of the source label space.

Great studies have been conducted towards the task of PDA to simultaneously promote positive transfer from the common classes between the domains and alleviate the negative transfer from the outlier classes [4, 1, 169]. Importance weighted adversarial nets [169] develops a two-domain classifier strategy to estimate the relative importance of the source domain samples. Selective adversarial network (SAN) [4] trains different domain discriminators for each source class separately to align the distributions of the source and target domains across the shared label space. Partial adversarial domain adaptation (PADA) [1] adopts a single adversarial network and incorporates class-level weights to both source classifier and domain discriminator for down-weighing the samples of outlier source classes. Example Transfer Network (ETN) [5] improves upon the PADA approach

by introducing an auxiliary domain discriminator to quantify the transferability of each source sample.

Despite the efficacy of the existing PDA approaches in various tasks, they often align the marginal distributions of the shared classes between the domains without considering the conditional distributions [1, 168, 5]. This may degenerate the performance of the model due to the negative transfer of irrelevant knowledge. To circumvent this issue, we utilize pseudo-labels for the target domain samples and develop a multi-class adversarial architecture to jointly align the marginal and class-conditional distributions (see Figure 4.1 for more clarification). Inspired by [170], we propose to align labeled source centroid and pseudo-labeled target centroid to mitigate the adverse effect of the noisy pseudo-labels. Similar to [1], we incorporate class-level weights into our cost function to down-weight the contributions of the source samples belonging to the outlier classes. Furthermore, we introduce two novel regularization functions to promote the selection of a small subset of classes that are in common between the source and target domains and enhance the classifier confidence in predicting the labels of irrelevant samples from both domains.

4.3 Problem Formulation

Let $\{(\boldsymbol{x}_s^i, \boldsymbol{y}_s^i)\}_{i=1}^{n_s}$ be a set of n_s sample images collected $i.i.d$ from the source domain $\mathcal{D}_s$, where $\boldsymbol{x}_s^i$ denotes the i^{th} source image with label $\boldsymbol{y}_s^i$. Similarly, let $\{\boldsymbol{x}_t^i\}_{i=1}^{n_t}$ be a set of n_t sample images drawn $i.i.d$ from the target domain $\mathcal{D}_t$, where $\boldsymbol{x}_t^i$ indicates the i^{th} target image. To clarify the notation, let $\mathcal{X} = \mathcal{X}_s \cup \mathcal{X}_t$ be the set of entire images captured from both domains, where $\mathcal{X}_s = \{\boldsymbol{x}_s^i\}_{i=1}^{n_s}$ and $\mathcal{X}_t = \{\boldsymbol{x}_t^i\}_{i=1}^{n_t}$. The UDA methods assume the source and target domains possess the same set of labels, denoted as $\mathcal{C}_s$ and $\mathcal{C}_t$, respectively. In the absence of target labels, the primary goal of the UDA methods is to learn transferable features that can reduce the shift between the marginal distributions of both domains. One

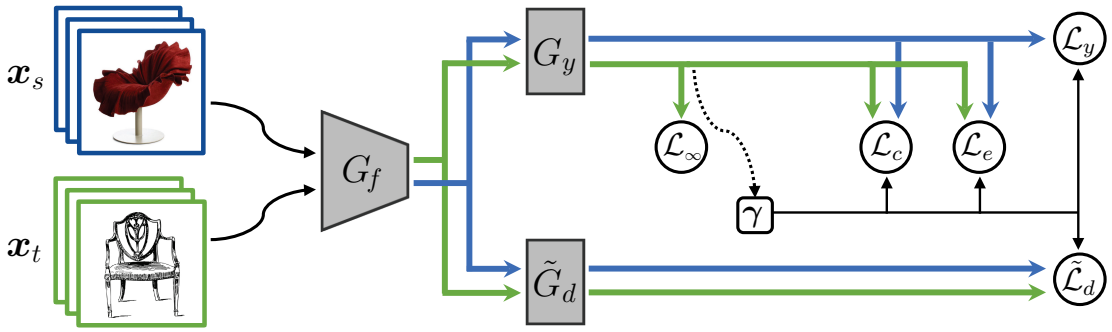

Figure 4.2: Overview of the proposed adversarial network for partial transfer learning. The network consists of a feature extractor, a classifier, and a domain discriminator, denoted by G_f, G_y, and $\tilde{G}_d$, respectively. The blue arrows show the source flow and the green ones depict the target flow. Loss functions $\mathcal{L}_y$, $\tilde{\mathcal{L}}_d$, $\mathcal{L}_c$, $\mathcal{L}_e$, and $\mathcal{L}_\infty$ denote the classification loss, the discriminative loss, the centroid alignment loss, the entropy loss, and the selection loss, respectively. *Best viewed in color.*

promising direction towards this goal is to train a domain adversarial network [61, 68] consisting of a discriminator G_d for predicting the domain labels, a feature extractor G_f to learn domain-invariant feature representations for deceiving the discriminator, and a classifier G_y that classifies the source domain samples. Training such adversarial network is equivalent to solving the following optimization problem

$$\begin{aligned}\max_{\boldsymbol{\theta}_d} \min_{\boldsymbol{\theta}_y,\boldsymbol{\theta}_f} \; & \frac{1}{n_s} \sum_{\boldsymbol{x}^i \in \mathcal{X}_s} L_y(G_y(G_f(\boldsymbol{x}^i;\boldsymbol{\theta}_f);\boldsymbol{\theta}_y), \boldsymbol{y}_s^i) \\ & -\frac{\lambda}{n} \sum_{\boldsymbol{x}^i \in \mathcal{X}} L_d(G_d(G_f(\boldsymbol{x}^i;\boldsymbol{\theta}_f);\boldsymbol{\theta}_d), d^i),\end{aligned} \tag{4.1}$$

where $n = n_s + n_t$ denotes the total number of images, $\lambda > 0$ is a regularization parameter, $\boldsymbol{y}_s^i$ is a one-hot vector denoting the class label of image $\boldsymbol{x}^i$, and $d^i \in \{0, 1\}$ indicates its domain label. L_y and L_d are cross-entropy loss functions corresponding to the classifier G_y and the domain discriminator G_d, respectively. Moreover, variables $\boldsymbol{\theta}_f$, $\boldsymbol{\theta}_y$, and $\boldsymbol{\theta}_d$ are the network parameters associated with G_f, G_y, and G_d, respectively. For the brevity of notation, we drop the reference to the network parameters in the subsequent formulations.

As noted earlier, standard domain adaptation approaches assume that the source and target domains possess the same label space, i.e. $\mathcal{C}_s = \mathcal{C}_t$. This assumption may not be fulfilled in a wide range of practical applications in which $\mathcal{C}_s$ is large and diverse (e.g., ImageNet) and $\mathcal{C}_t$ only contains a small subset of the source classes (e.g., Office-31), i.e. $\mathcal{C}_t \subset \mathcal{C}_s$. In such scenarios, it is hard to identify the shared label space between the domains since target labels and target label space $\mathcal{C}_t$ are unknown during the training procedure. Under this condition, matching the marginal distributions may not necessarily facilitate the classification task in the target domain and a classifier with adaptation may perform worse than a standard classifier trained on the source samples. This is attributed to the adverse effect of transferring information from the outlier classes $\mathcal{C}_s \backslash \mathcal{C}_t$ [4, 1]. Hence, the primary goal in partial domain adaptation is to identify and reject the outlier classes and simultaneously align the conditional distributions of the source and target domains across the shared label space. One of the well-established works toward this goal is Partial Adversarial Domain Adaptation (PADA) [1] which highlights the shared classes and reduces the importance of the outlier classes via the following weighting procedure

$$\boldsymbol{\gamma} = \frac{1}{n_t} \sum_{i=1}^{n_t} \hat{\boldsymbol{y}}_t^i, \tag{4.2}$$

where $\hat{\boldsymbol{y}}_t^i = G_y(G_f(\boldsymbol{x}_t^i))$ denotes the output of the classifier G_y to the target sample $\boldsymbol{x}_t^i$ and it can be considered as a probability distribution over the source label space $\mathcal{C}_s$. The weight vector $\boldsymbol{\gamma}$ is further normalized as $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} \backslash \max(\boldsymbol{\gamma})$ to demonstrate the relative importance of the classes. The weights associated with the outlier classes are expected to be much smaller than that of the shared classes, mainly because the target samples are significantly dissimilar to the samples belonging to the outlier classes. Ideally, $\boldsymbol{\gamma}$ is expected to be a vector whose elements are non-zero except those corresponding to the outlier classes. Given that, PADA proposed to down-weigh the contributions of the source samples belonging to the outlier classes $\mathcal{C}_s \backslash \mathcal{C}_t$ by adding the class-level weight vector $\boldsymbol{\gamma}$ to both source classifier

G_y and domain discriminator G_d. Therefore, the objective of PADA can be formulated as follows

$$\begin{aligned}\max_{\boldsymbol{\theta}_d} \min_{\boldsymbol{\theta}_y,\boldsymbol{\theta}_f} \ & \frac{1}{n_s} \sum_{\boldsymbol{x}^i \in \mathcal{X}_s} \gamma_{c_i} \, L_y(G_y(G_f(\boldsymbol{x}^i)), \boldsymbol{y}^i) \\ & -\frac{\lambda}{n_s} \sum_{\boldsymbol{x}^i \in \mathcal{X}_s} \gamma_{c_i} \, L_d(G_d(G_f(\boldsymbol{x}^i)), d^i) \\ & -\frac{\lambda}{n_t} \sum_{\boldsymbol{x}^i \in \mathcal{X}_t} L_d(G_d(G_f(\boldsymbol{x}^i)), d^i),\end{aligned} \tag{4.3}$$

where scalar γ_{c_i} denotes the class weight of sample $\boldsymbol{x}^i$ and $c_i = \operatorname{argmax}_j y^i_j$ indicates the index of the largest element in vector $\boldsymbol{y}^i$.

4.4 Proposed Method

Although the weighting scheme (4.2) is able to effectively match the marginal distributions of the source and target domains in the shared label space, there is no guarantee that the corresponding class-conditional distributions can also be drawn close. This may significantly degenerate the performance of the model due to the negative transfer of irrelevant knowledge. To circumvent this issue, we introduce a novel adversarial architecture to jointly align the marginal and class-conditional distributions in the shared label space. The proposed model adopts a multi-class discriminator $\tilde{G}_d$, parameterized by $\tilde{\boldsymbol{\theta}}_d$, to classify the feature representations $G_f(\boldsymbol{x}^i)$ into $2 \times |\mathcal{C}_s|$ categories, where the first and the last $|\mathcal{C}_s|$ categories respectively correspond to the probability distribution over the source label space $\mathcal{C}_s$ and target label space $\mathcal{C}_t$ ($\mathcal{C}_t \subset \mathcal{C}_s$). We propose to train the discriminator $\tilde{G}_d$ with the following objective function

$$\begin{aligned}\tilde{\mathcal{L}}_d(\boldsymbol{\theta}_f, \tilde{\boldsymbol{\theta}}_d) = & -\frac{1}{n_s} \sum_{\boldsymbol{x}^i \in \mathcal{X}_s} \gamma_{c_i} L_d(\tilde{G}_d(G_f(\boldsymbol{x}^i)), \tilde{\boldsymbol{d}}^i) \\ & -\frac{1}{n_t} \sum_{\boldsymbol{x}^i \in \mathcal{X}_t} L_d(\tilde{G}_d(G_f(\boldsymbol{x}^i)), \tilde{\boldsymbol{d}}^i),\end{aligned}$$

where vector $\tilde{\boldsymbol{d}}^i \in \mathbb{R}^{2\times|\mathcal{C}_s|}$ is defined as the domain-class label of sample point $\boldsymbol{x}^i$. Due to the lack of class labels for the target samples, we set $\tilde{\boldsymbol{d}}^i$ to $[\boldsymbol{y}_s^i, \mathbf{0}]$ if $\boldsymbol{x}^i \in \mathcal{X}_s$ and use $[\mathbf{0}, \tilde{\boldsymbol{y}}_t^i]$ if $\boldsymbol{x}^i \in \mathcal{X}_t$, where $\tilde{\boldsymbol{y}}_t^i$ corresponds to the pseudo-label generated by classifier G_y and is given by

$$\tilde{\boldsymbol{y}}_t^i = \operatorname{argmax}_c \, \boldsymbol{e}_c^\top G_y(G_f(\boldsymbol{x}_t^i)),$$

where $\{\boldsymbol{e}_c\}_{c=1}^{|\mathcal{C}_s|}$ denotes the standard unit basis in $\mathbb{R}^{|\mathcal{C}_s|}$. Moreover, the negative transfer can be efficiently alleviated by incorporating the weight vector $\boldsymbol{\gamma}$ into the loss $\tilde{\mathcal{L}}_d$ which results in selecting out the source samples belonging to the outlier label space $\mathcal{C}_s \setminus \mathcal{C}_t$. It is noteworthy to mention that the direct use of pseudo-labels may degrade the classification performance as the pseudo-labels are predicted by the classifier and hence they may be noisy and inaccurate. Many literature methods leverage the theory of domain adaptation [171] to present error analysis and derive certain bounds on the error introduced by incorporating the pseudo-labels [172, 170]. These analysis are not generally applicable to the problem of partial domain adaptation as they mainly rely on the assumption that the source and target domains possess the same set of labels.

With the proposed multi-class adversarial loss $\tilde{\mathcal{L}}_d$, the key challenge is how to tackle the uncertainty in pseudo-labels. One promising approach to mitigate the adverse effect of falsely-pseudo-labeled target samples is to align labeled source centroids and pseudo-labeled target centroids in the feature space [170]. However, this approach hardly fits the partial domain adaptation scenario in which the target label space is a subset of source label space. We propose to modify the aforementioned approach by incorporating weight vector $\boldsymbol{\gamma}$ to highlight the mismatch between the centroids of the shared classes. Hence, the weighted centroid alignment loss function can be formulated as

$$\mathcal{L}_c(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y) = \sum_{i=1}^{|\mathcal{C}_s|} \gamma_i \, \|\boldsymbol{M}_s^i - \boldsymbol{M}_t^i\|_2^2,$$

where $\boldsymbol{M}_s^i$ and $\boldsymbol{M}_t^i$ respectively denote the feature centroids for the i^{th} class in the source and target domains. These vectors are computed via the following formulas

$$\boldsymbol{M}_s^i = \frac{1}{|\mathcal{O}_i|} \sum_{\boldsymbol{x}^i \in \mathcal{O}_i} G_f(\boldsymbol{x}^i), \quad \boldsymbol{M}_t^i = \frac{1}{|\tilde{\mathcal{O}}_i|} \sum_{\boldsymbol{x}^i \in \tilde{\mathcal{O}}_i} G_f(\boldsymbol{x}^i),$$

where $\mathcal{O}_i$ is the set of source samples belonging to the i^{th} class and $\tilde{\mathcal{O}}_i$ denotes the set of target samples assigned to the i^{th} class.

In what follows, we propose two novel regularization functions to derive more discriminative class weights and to increase the confidence level of the classifier in predicting the labels of the irrelevant samples across both domains.

Motivated by the assumption that the target samples are dissimilar to the samples of the outlier classes, we propose a row-sparsity regularization term that promotes the selection of a small subset of source domain classes that appear in the target domain. This, in turn, encourages the weight vector $\boldsymbol{\gamma}$ to be a vector of all zeros except for the elements corresponding to the shared classes. This selection regularization can be formulated as follows

$$\mathcal{L}_\infty(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y) = \frac{1}{|\mathcal{C}_s|} \left\| G_y(G_f(\boldsymbol{x}_t^1)), \ldots, G_y(G_f(\boldsymbol{x}_t^{|\mathcal{X}_t|})) \right\|_{1,\infty},$$

where $|.|$ denotes the cardinality of its input set and $\|.\|_{1,\infty}$ computes the sum of the infinity norms of the rows of an input matrix. To illustrate, for an arbitrary matrix $\boldsymbol{A} = [\boldsymbol{a}_1|\boldsymbol{a}_2|\ldots|\boldsymbol{a}_n]^\top \in \mathbb{R}^{n\times m}$, scalar $\|\boldsymbol{a}_i\|_\infty$ denotes the maximum absolute value of i^{th} row. Therefore, regularization term $\|\boldsymbol{A}\|_{1,\infty} = \sum_{i=1}^{n} \|\boldsymbol{a}_i\|_\infty$ promotes sparsity on the maximum absolute value of each row which in turn leads to some zero rows in matrix $\boldsymbol{A}$.

The regularization term $\mathcal{L}_\infty$ takes into consideration the relation between the entire target samples and encourages the classifier to generate a sparse output vector with its non-zero entries located at certain indices correspond to the classes shared between the domains. Notice that this regularization term does not directly enforce a specific number of classes to be chosen but rather promotes the network to select a subset of source domain classes.

Besides the outlier classes, the irrelevant samples are inherently less transferable and they may significantly degrade the target classification performance in different PDA tasks. To reduce the negative effect of irrelevant samples in the training procedure, we propose to leverage the following entropy minimization term

$$\begin{aligned}\mathcal{L}_e(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y) = & \frac{1}{n_s} \sum_{\boldsymbol{x}^i \in \mathcal{X}_s} \gamma_{c_i} L_y^e(G_y(G_f(\boldsymbol{x}^i))) \\ & + \frac{1}{n_t} \sum_{\boldsymbol{x}^i \in \mathcal{X}_t} L_y^e(G_y(G_f(\boldsymbol{x}^i)),\end{aligned}$$

where L_y^e is the entropy loss function corresponding to the classifier G_y. Generally, regularization $\mathcal{L}_e$ encourages the classifier to produce vectors with one dominant element denoting the label (or pseudo-label) of samples. This, in turn, enhances the performance of the feature extractor and helps to learn more transferable features for classification. Moreover, weight vector γ is incorporated to highlight the importance of samples belonging to the shared classes.

By combining the aforementioned loss functions, training our proposed model is equivalent to solving the following minimax saddle point optimization problem

$$\begin{aligned}\max_{\tilde{\boldsymbol{\theta}}_d} \min_{\boldsymbol{\theta}_y, \boldsymbol{\theta}_f} \quad & \frac{1}{n_s} \sum_{\boldsymbol{x}^i \in \mathcal{X}_s} \gamma_{c_i} L_y(G_y(G_f(\boldsymbol{x}^i)), \boldsymbol{y}^i) \\ & + \lambda \tilde{\mathcal{L}}_d(\boldsymbol{\theta}_f, \tilde{\boldsymbol{\theta}}_d) + \mathcal{L}_c(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y) \\ & + \mu \mathcal{L}_\infty(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y) + \zeta \mathcal{L}_e(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y),\end{aligned} \tag{4.4}$$

where λ, μ, and ζ are positive hyperparameters to control the contribution of each loss component.

4.5 Experiments

This section evaluates the efficacy of our approach, named CCPDA, through conducting empirical experiments on two widely used benchmark datasets for partial domain

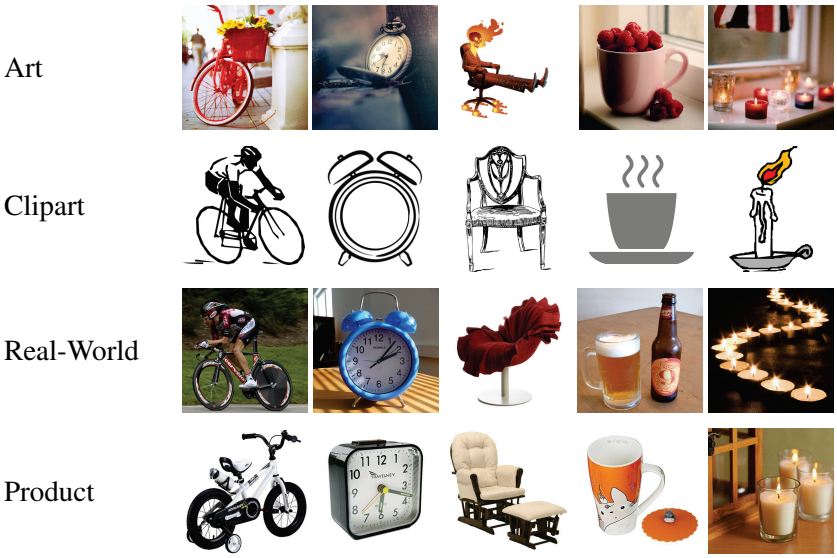

Figure 4.3: Sample images from the Office-Home dataset.

Table 4.1: Classification accuracy of partial domain adaptation tasks on Office-31.

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
ResNet	75.59	96.27	98.09	83.44	83.92	84.97	87.05
DANN	73.56	96.27	98.73	81.53	82.78	86.12	86.50
ADDA	75.67	95.38	99.85	83.41	83.62	84.25	87.03
MADA	90.00	97.40	99.60	87.80	70.30	66.40	85.20
RTN	78.98	93.22	85.35	77.07	89.25	89.46	85.56
IWAN	89.15	99.32	99.36	90.45	95.62	94.26	94.69
SAN	93.90	99.32	99.36	94.27	94.15	88.73	94.96
PADA	86.54	99.32	**100.0**	82.17	92.69	95.41	92.69
ETN	94.52	**100.0**	**100.0**	95.03	**96.21**	94.64	96.73
CCPDA	**99.66**	**100.0**	**100.0**	**97.45**	95.72	**95.71**	**98.09**

adaptation (PDA) problem. The experiments are performed on different PDA tasks in an unsupervised setting where neither the target labels nor the target label space is available. In what follows, we give more explanations about the datasets, the PDA tasks, and the network hyperparameters used in our experiments.

4.5.1 Setup

Dataset: We evaluate the performance of CCPDA on two commonly used datasets for the task of partial domain adaptation: Office-31 [173] and Office-Home [174]. Office-31

Table 4.2: Classification accuracy of partial domain adaptation tasks on Office-Home.

Method	**Ar→Cl**	**Ar→Pr**	**Ar→Rw**	**Cl→Ar**	**Cl→Pr**	**Cl→Rw**	**Pr→Ar**	**Pr→Cl**	**Pr→Rw**	**Rw→Ar**	**Rw→Cl**	**Rw→Pr**	**Avg**
ResNet	46.33	67.51	75.87	59.14	59.94	62.73	58.22	41.79	74.88	67.40	48.18	74.17	61.35
DANN	43.76	67.90	77.47	63.73	58.99	67.59	56.84	37.07	76.37	69.15	44.30	77.48	61.72
ADDA	45.23	68.79	79.21	64.56	60.01	68.29	57.56	38.89	77.45	70.28	45.23	78.32	62.82
RTN	49.31	57.70	80.07	63.54	63.47	73.38	65.11	41.73	75.32	63.18	43.57	80.50	63.07
IWAN	53.94	54.45	78.12	61.31	47.95	63.32	54.17	52.02	81.28	76.46	56.75	82.90	63.56
SAN	44.42	68.68	74.60	67.49	64.99	**77.80**	59.78	44.72	80.07	72.18	50.21	78.66	65.30
PADA	51.95	67.00	78.74	52.16	53.78	59.03	52.61	43.22	78.79	73.73	56.60	77.09	62.06
ETN	**59.24**	77.03	79.54	62.92	65.73	75.01	68.29	**55.37**	**84.37**	75.72	**57.66**	84.54	70.45
CCPDA	55.31	**80.11**	**88.07**	**73.28**	**71.21**	77.63	**71.89**	52.97	81.41	**81.81**	56.21	**85.15**	**72.92**

object dataset consists of $4,652$ images from 31 classes, where the images are collected from three different domains: *Amazon* (**A**), *Webcam* (**W**), and *DSLR* (**D**). We follow the procedure presented in the literature [1, 5] to transfer knowledge from a source domain with 31 classes to a target domain with 10 classes. The results are reported as the average classification accuracy of the target domain over five independent experiments across six different PDA tasks: **A** $\rightarrow$ **W**, **W** $\rightarrow$ **A**, **D** $\rightarrow$ **W**, **W** $\rightarrow$ **D**, **A** $\rightarrow$ **D**, and **D** $\rightarrow$ **A**.

Office-Home is a more challenging dataset that contains $15,500$ images collected from four distinct domains: *Art* (**Ar**), *Clipart* (**Cl**), *Product* (**Pr**), and *Real-World* (**Rw**), where each domain has 65 classes. Example images from this dataset are provided in Figure 4.3. Following the procedure presented in [1, 5], we aim to transfer information from a source domain containing 65 classes to a target domain with 25 classes. The results on this dataset are also reported as the average classification accuracy of the target domain over five independent experiments across twelve pairs of source-target adaptation tasks: **Ar** $\rightarrow$ **Cl**, **Ar** $\rightarrow$ **Pr**, **Ar** $\rightarrow$ **Rw**, **Cl** $\rightarrow$ **Ar**, **Cl** $\rightarrow$ **Pr**, **Cl** $\rightarrow$ **Rw**, **Pr** $\rightarrow$ **Ar**, **Pr** $\rightarrow$ **Cl**, **Pr** $\rightarrow$ **Rw**, **Rw** $\rightarrow$ **Ar**, **Rw** $\rightarrow$ **Cl**, and **Rw** $\rightarrow$ **Pr**.

We follow the standard evaluation protocols for partial domain adaptation [4, 1] and compare the performance of CCPDA against several deep transfer learning methods: Domain Adversarial Neural Network (DANN) [68], Residual Transfer Networks (RTN) [163], Adversarial Discriminative Domain Adaptation (ADDA) [69], Importance Weighted Ad-

versarial Nets (IWAN) [169], Multi-Adversarial Domain Adaptation (MADA) [66] , Selective Adversarial Network (SAN) [4], Partial Adversarial Domain Adaptation (PADA) [1], and Example Transfer Network (ETN) [5]. Moreover, in order to demonstrate the efficacy brought by different components of the proposed PDA model, we conduct an ablation study by evaluating three variants of CCPDA: CCPDA$_\infty$ is a variant of CCPDA without incorporating the selection regularization term $\mathcal{L}_\infty$, CCPDA$_e$ denotes a variant without considering $\mathcal{L}_e$, and CCPDA$_{d,c}$ [175] is a variant with a binary discriminator and without considering the weighted centroids alignment term $\mathcal{L}_c$.

Parameter: We use PyTorch [176] to implement CCPDA and adopt ResNet-50 [177] model pre-trained on ImageNet [178], as the backbone for the network G_f. We fine-tune the entire feature layers and apply back-propagation to train the domain discriminator $\tilde{G}_d$ and the classifier G_y. Since parameters $\boldsymbol{\theta}_y$ and $\tilde{\boldsymbol{\theta}}_d$ are trained from scratch, their learning rates are set to be 10 times greater than that of $\boldsymbol{\theta}_f$. To solve the minimax problem (4.3), we use mini-batch stochastic gradient descent (SGD) with a momentum of 0.95 and the learning rate is adjusted during SGD by: $\eta = \frac{\eta_0}{(1+\alpha\times\rho)^\beta}$ where $\eta_0 = 10^{-2}$, $\alpha = 10$, $\beta = 0.75$, and ρ, denoting the training progress, linearly changes from 0 to 1 [68, 1]. We use a batch size $b = 72$ with 36 samples for each domain. Parameter μ is set to 0.1 for both Office-31 and Office-Home datasets. Notice that since the classifier is not appropriately trained in the first few epochs, the value of μ can be gradually increased from 0 to 0.1. Other hyper-parameters are tuned by importance weighted cross validation [179] on labeled source samples and unlabeled target samples.

As we use mini-batch SGD for optimizing our model, categorical information in each batch is usually inadequate for obtaining an accurate estimation of the source and target centroids. This, in turn, may adversely affect the alignment performance. To mitigate this issue, we align the moving average centroids of the source and target classes in the feature

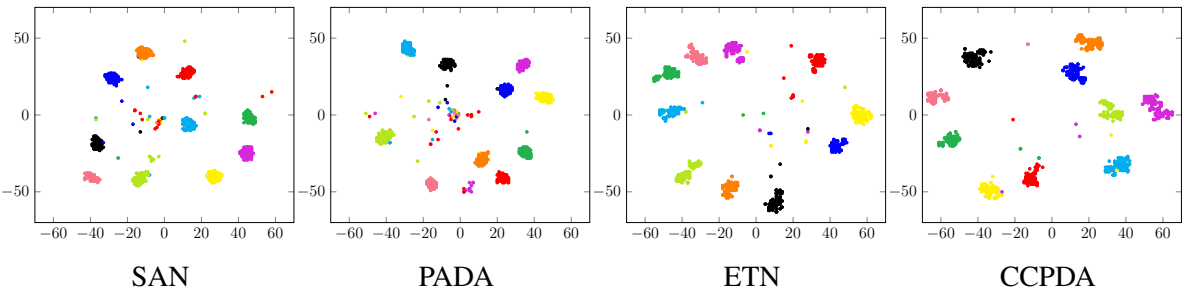

Figure 4.4: The t-SNE visualization of SAN [4], PADA [1], ETN [5], and CCPDA on partial domain adaptation task $\mathbf{A} \rightarrow \mathbf{W}$ with class information (samples are colored w.r.t. their classes). *Best viewed in color.*

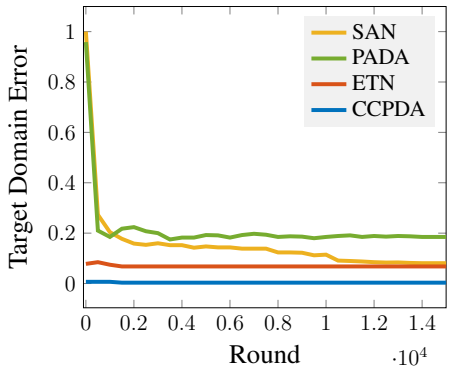

Figure 4.5: Empirical analysis of the target domain error through the training process. *Best viewed in color.*

space (with coefficient 0.7) rather than aligning the inaccurate centroids obtained in each iteration.

4.5.2 Results

The target domain classification accuracy for various methods on six PDA tasks of Office-31 dataset and twelve PDA tasks of Office-Home dataset are reported in Tables 4.1 and 4.2. The entire results are reported based on the ResNet-50 and the scores of the competitor methods are directly collected from [5].

Observe that unsupervised domain adaptation methods such as ADDA, DANN, and MADA have exhibited worse performance than the standard ResNet-50 on some PDA tasks

in both datasets. This can be attributed to the fact that these methods aim to align the marginal distributions across the domains and hence are prone to the negative transfer introduced by the outlier classes. On the other hand, the partial domain adaptation methods, such as PADA, SAN, IWAN, ETN, and CCPDA, achieve promising results on most of the PDA tasks since they leverage different mechanisms to highlight a subset of samples that are more transferable across both domains.

Among the competing partial domain adaptation approaches in Tables 4.1 and 4.2, SAN is the only approach that seeks to directly align the conditional distributions of the source and target domains. However, unlike CCPDA, SAN uses a different architecture with $|\mathcal{C}_s|$ class-wise domain discriminators to identify the domain-class label of each sample. As reported in Tables 4.1 and 4.2, CCPDA outperforms SAN with a large margin in all PDA tasks on both Office-31 and Office-Home datasets. Moreover, CCPDA requires fewer parameters compared to SAN. This in turn demonstrates the efficiency and efficacy of the proposed class-conditional model.

The results in Table 4.1 indicate that CCPDA outperforms the competing methods on most of the PDA tasks from Office-31 dataset. In particular, CCPDA achieves considerable improvement on **A** $\rightarrow$ **W** and **A** $\rightarrow$ **D** tasks. It also increases the average accuracy of all tasks by almost 1.36%. Moreover, Table 4.2 shows that CCPDA outperforms other PDA approaches with a large margin on five pairs of source-target adaptation tasks: **Ar** $\rightarrow$ **Pr**, **Ar** $\rightarrow$ **Rw**, **Cl** $\rightarrow$ **Ar**, **Cl** $\rightarrow$ **Pr**, and **Rw** $\rightarrow$ **Ar**. The numerical results provided in Tables 4.1 and 4.2 corroborate CCPDA can effectively align the class-conditional distribution, mitigate transferring knowledge from the outlier source classes, and promote positive transfer between the domains in the shared label space.

Furthermore, we perform an ablation study to evaluate the efficacy brought by different components of the proposed PDA model. We consider PADA as a baseline variant of CCPDA with binary domain discriminator G_d and without regularization terms $\mathcal{L}_c$, $\mathcal{L}_\infty$,

Table 4.3: Classification accuracy of CCPDA and its variants for Partial Domain Adaptation tasks on Office-31 dataset.

Method	**A→W**	**D→W**	**W→D**	**A→D**	**D→A**	**W→A**	**Avg**
PADA	86.54	99.32	**100.0**	82.17	92.69	95.41	92.69
CCPDA$_\infty$	95.12	99.32	**100.0**	93.21	**96.03**	95.19	96.48
CCPDA$_e$	97.45	96.64	**100.0**	96.47	94.92	93.86	96.56
CCPDA$_{d,c}$	93.42	97.62	**100.0**	90.43	93.45	95.53	95.07
CCPDA	**99.66**	**100.0**	**100.0**	**97.45**	95.72	**95.71**	**98.09**

and $\mathcal{L}_e$. The results are reported in Table 4.3 and they reveal interesting observations. CCPDA$_{d,c}$ outperforms PADA in most of the tasks, which highlights the importance of the incorporated regularization terms $\mathcal{L}_\infty$ and $\mathcal{L}_e$ in rejecting the outlier source samples. Moreover, we can see that both variants CCPDA$_\infty$ and CCPDA$_e$ improved the accuracy of the original baseline, which corroborate the efficacy of our class-conditional domain discriminator $\tilde{G}_d$. Overall, observe that different components of the proposed method bring complementary information into the model and have contributions in achieving the state-of-the-art classification results.

Visualization: To better demonstrate the ability of the proposed method in aligning the feature distributions in the shared label space, we visualize the bottleneck representations learned by SAN, PADA, ETN, and CCPDA on task **A (31 classes) → W (10 classes)** using t-SNE embedding [180] (Shown in Figure 4.4). It is desired to embed the source and target sample points of the same class close together while keeping embeddings from different classes far apart. Observe that CCPDA is able to effectively discriminate the classes shared between the domains while minimizing the distance between the same classes in both domains.

Convergence Performance: To highlight other advantages of our approach, we compare the test error rate obtained by CCPDA against various methods SAN, PADA, and ETN on partial domain adaptation task **A (31 classes) → W (10 classes)**, from Office dataset.

Figure 4.5 illustrates the convergence behavior of the test errors in $15,000$ iterations. Each curve is obtained by averaging over 5 independent runs for the entire test samples. Observe that comparing to the competitor methods, CCPDA not only converges very quickly but also achieves a lower error rate.

4.6 Conclusion

This work presented a novel adversarial architecture for the task of partial domain adaptation. The proposed model adopts a multi-class adversarial loss function to jointly align the marginal and class-conditional distributions across the shared classes between the source and target domains. Furthermore, it leverages two regularization functions to reduce the adverse effects of the outlier classes and the irrelevant samples in transferring information. Several experiments performed on the standard benchmark datasets for partial domain adaptation have demonstrated that our method can outperform the state-of-the-art methods on multiple adaptation tasks in terms of the classification performance.

CHAPTER 5

Conclusion

In this dissertation, we first introduced a general convex relaxation framework to solve a class of non-convex optimization problems, called bilinear matrix inequalities (BMI). We developed a novel computationally cheap relaxation technique that only relies on quadratic convex constraints to transform the BMIs into polynomial-time solvable programs. The proposed relaxation is then generalized to a sequential convexification scheme which can start from an arbitrary initial point to recover feasible and near-optimal solutions of BMIs. Moreover, we presented a theoretical analysis of our scheme and investigated the conditions under which the convexification method is guaranteed to produce feasible points. The proposed framework is readily applicable to a variety of machine learning problems such as discriminative dimensionality reduction and graph matching. We performed experiments on the benchmark datasets for graph matching to demonstrate the potential of the proposed method in finding solutions with high quality in polynomial-time.

Second, we presented a novel spectral clustering-based approach which uses a deep architecture to address the subspace clustering problem. The proposed method improves upon the existing deep approaches by leveraging information exploited from different levels of the networks to transform input samples into multi-level representations lying on a union of linear subspace. Moreover, it is able to use pseudo-labels generated by the spectral clustering technique to effectively supervise the representation learning procedure and boost the final clustering performance. Experiments on benchmark datasets demonstrate that the proposed approach is able to efficiently handle clustering from the non-linear subspaces and it achieves better results compared to the state-of-the-art methods.

Third, we developed a novel adversarial architecture for the task of partial domain adaptation. The proposed model adopts a multi-class adversarial loss function to jointly align the marginal and class-conditional distributions across the shared classes between the source and target domains. Furthermore, it leverages two regularization functions to reduce the adverse effects of the outlier classes and the irrelevant samples in transferring information. Several experiments performed on the standard benchmark datasets for partial domain adaptation have demonstrated that our method can outperform the state-of-the-art methods on multiple adaptation tasks in terms of the classification performance.

REFERENCES

[1] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in *ECCV*, 2018.

[2] M. Leordeanu, R. Sukthankar, and M. Hebert, "Unsupervised learning for graph matching," *Int. J. Comput. Vis*, vol. 96, no. 1, pp. 28–45, 2012.

[3] F. Zhou and F. De la Torre, "Deformable graph matching," in *CVPR*, 2013.

[4] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in *CVPR*, 2018.

[5] Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, "Learning to transfer examples for partial domain adaptation," in *CVPR*, 2019.

[6] L. El Ghaoui and S.-l. Niculescu, *Advances in linear matrix inequality methods in control*. SIAM, 2000.

[7] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[8] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, no. Jan, pp. 27–72, 2004.

[9] E. A. Yildirim, "On the minimum volume covering ellipsoid of ellipsoids," *SIAM Journal on Optimization*, vol. 17, no. 3, pp. 621–641, 2006.

[10] Y. Zheng, Y. Fu, A. Lam, I. Sato, and Y. Sato, "Separating fluorescent and reflective components by using a single hyperspectral image," in *ICCV*, 2015.

[11] O. Toker and H. Ozbay, "On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback," in *ACC*, vol. 4, 1995, pp. 2525–2526.

[12] V. Blondel and J. N. Tsitsiklis, "NP-hardness of some linear control design problems," *SIAM J. Control Optim.*, vol. 35, no. 6, pp. 2118–2127, 1997.

[13] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inform. Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.

[14] W. Bian and D. Tao, "Max-min distance analysis by using sequential SDP relaxation for dimension reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 1037–1050, 2011.

[15] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Math. Program.*, vol. 96, no. 2, pp. 293–320, 2003.

[16] A. A. Ahmadi and A. Majumdar, "Dsos and sdsos optimization: more tractable alternatives to sum of squares and semidefinite optimization," *arXiv preprint arXiv:1706.02586*, 2017.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, 2011.

[18] A. V. Fiacco and G. P. McCormick, *Nonlinear programming: sequential unconstrained minimization techniques*. SIAM, 1990, vol. 4.

[19] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright, "On projected newton barrier methods for linear programming and an equivalence to karmarkars projective method," *Math. Program.*, vol. 36, no. 2, pp. 183–209, 1986.

[20] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[21] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[22] F. Lauer and C. Schnörr, "Spectral clustering of linear subspaces for motion segmentation," in *ICCV*, 2009.

[23] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1832–1845, 2010.

[24] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *CVPR*, 2015.

[25] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *ICCV*, 2015.

[26] E. Ntoutsi, K. Stefanidis, K. Rausch, and H.-P. Kriegel, "Strength lies in differences: Diversifying friends for recommendations through subspace clustering," in *CIKM*, 2014.

[27] A. Zhang, N. Fawaz, S. Ioannidis, and A. Montanari, "Guess who rated this movie: Identifying users through subspace clustering," *arXiv preprint arXiv:1208.1544*, 2012.

[28] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, *Automatic subspace clustering of high dimensional data for data mining applications*. ACM, 1998, vol. 27, no. 2.

[29] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[30] C. W. Gear, "Multibody grouping from motion images," *Int. J. Comput. Vis*, vol. 29, no. 2, pp. 133–150, 1998.

[31] T. E. Boult and L. G. Brown, "Factorization-based segmentation of motions," in *IEEE workshop on visual motion*, 1991.

[32] L. Lu and R. Vidal, "Combined central and subspace clustering for computer vision applications," in *ICML*, 2006.

[33] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, 2005.

[34] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, 2013.

[35] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *NeurIPS*, 2017.

[36] V. M. Patel, H. Van Nguyen, and R. Vidal, "Latent space sparse subspace clustering," in *ICCV*, 2013.

[37] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *ICIP*, 2014.

[38] S. Xiao, M. Tan, D. Xu, and Z. Y. Dong, "Robust kernel low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2268–2281, 2015.

[39] M. Yin, Y. Guo, J. Gao, Z. He, and S. Xie, "Kernel sparse subspace clustering on symmetric positive definite manifolds," in *CVPR*, 2016.

[40] T. Zhang, P. Ji, M. Harandi, W. Huang, and H. Li, "Neural collaborative subspace clustering," *arXiv preprint arXiv:1904.10596*, 2019.

[41] X. Peng, S. Xiao, J. Feng, W.-Y. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior." in *IJCAI*, 2016.

[42] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu, "Deep spectral clustering using dual autoencoder network," *arXiv preprint arXiv:1904.13113*, 2019.

[43] P. Zhou, Y. Hou, and J. Feng, "Deep adversarial subspace clustering," in *CVPR*, 2018.

[44] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[45] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *ICANN*, 2011.

[46] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[48] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *NeurIPS*, 2013.

[49] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.

[50] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[51] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *ICCV*, 2015.

[52] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018.

[53] R. Girshick, "Fast r-cnn," in *ICCV*, 2015.

[54] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.

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

[56] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *CVPR*, 2012.

[57] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *ICCV*, 2013.

[58] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *NeurIPS*, 2014.

[59] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *ICML*, 2014.

[60] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *ICML*, 2015.

[61] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *ICCV*, 2015.

[62] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification," in *CVPR*, 2018.

[63] H. Kazemi, S. Soleymani, F. Taherkhani, S. Iranmanesh, and N. Nasrabadi, "Unsupervised image-to-image translation using domain-specific variational information bound," in *NeurIPS*, 2018.

[64] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[65] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *NeurIPS*, 2018.

[66] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *AAAI*, 2018.

[67] X. Wang, L. Li, W. Ye, M. Long, and J. Wang, "Transferable attention for domain adaptation," in *AAAI*, 2019.

[68] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.

[69] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *CVPR*, 2017.

[70] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data En.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[71] H. Kazemi and M. Namvar, "Adaptive compensation of actuator dynamics in manipulators without joint torque measurement," in *CDC*, 2013.

[72] L. Vandenberghe and V. Balakrishnan, "Algorithms and software for LMI problems in control," *IEEE. Control. Syst.*, vol. 17, no. 5, pp. 89–95, 1997.

[73] M. Kočvara, F. Leibfritz, M. Stingl, and D. Henrion, "A nonlinear SDP algorithm for static output feedback problems in COMPl$_e$ib," *IFAC-PapersOnLine*, vol. 38, no. 1, pp. 1055–1060, 2005.

[74] L. Ma, X. Meng, Z. Liu, and L. Du, "Multi-objective and reliable control for trajectory-tracking of rendezvous via parameter-dependent Lyapunov functions," *Acta Astronaut.*, vol. 81, no. 1, pp. 122–136, 2012.

[75] R. Correa, "A global algorithm for nonlinear semidefinite programming," *SIAM J. Optim.*, vol. 15, no. 1, pp. 303–318, 2004.

[76] R. Orsi, U. Helmke, and J. B. Moore, "A Newton-like method for solving rank constrained linear matrix inequalities," *Automatica*, vol. 42, no. 11, pp. 1875–1882, 2006.

[77] J. Fiala, M. Kočvara, and M. Stingl, "PENLAB: A MATLAB solver for nonlinear semidefinite optimization," *arXiv preprint arXiv:1311.5240*, 2013.

[78] M. Kocvara, M. Stingl, and P. GbR, "PENBMI users guide (version 2.0)," *software manual, PENOPT GbR, Hauptstrasse A*, vol. 31, p. 91338, 2005.

[79] K. Goh, L. Turan, M. Safonov, G. Papavassilopoulos, and J. Ly, "Biaffine matrix inequality properties and computational methods," in *ACC*, 1994.

[80] L. El Ghaoui and V. Balakrishnan, "Synthesis of fixed-structure controllers via numerical optimization," in *CDC*, 1994.

[81] S.-M. Liu and G. Papavassilopoulos, "Numerical experience with parallel algorithms for solving the BMI problem," *IFAC-PapersOnline*, vol. 29, no. 1, pp. 1827–1832, 1996.

[82] A. Hassibi, J. How, and S. Boyd, "A path-following method for solving BMI problems in control," in *ACC*, 1999.

[83] S. Ibaraki and M. Tomizuka, "Rank minimization approach for solving BMI problems with random search," in *ACC*, 2001.

[84] R. Doelman and M. Verhaegen, "Sequential convex relaxation for convex optimization with bilinear matrix equalities," in *Proc. IEEE P. Control Conf.*, 2016, pp. 1946–1951.

[85] A. Beck, A. Ben-Tal, and L. Tetruashvili, "A sequential parametric convex approximation method with applications to nonconvex truss topology design problems," *J. Global Optim.*, vol. 47, no. 1, pp. 29–51, 2010.

[86] D. Lee and J. Hu, "A sequential parametric convex approximation method for solving bilinear matrix inequalities," in *CDC*, 2016.

[87] Q. T. Dinh, S. Gumussoy, W. Michiels, and M. Diehl, "Combining convex–concave decompositions and linearization approaches for solving BMIs, with application to static output feedback," *IEEE Trans. Autom. Control*, vol. 57, no. 6, pp. 1377–1390, 2012.

[88] L. El Ghaoui, F. Oustry, and M. AitRami, "A cone complementarity linearization algorithm for static output-feedback and related problems," *IEEE Trans. Autom. Control*, vol. 42, no. 8, pp. 1171–1176, 1997.

[89] M. Fukuda and M. Kojima, "Branch-and-cut algorithms for the bilinear matrix inequality Eigenvalue problem," *Comput. Optim. Appl.*, vol. 19, no. 1, pp. 79–105, 2001.

[90] K.-C. Goh, M. Safonov, and G. Papavassilopoulos, "A global optimization approach for the BMI problem," in *CDC*, 1994.

[91] D. Tuan, P. Apkarian, and Y. Nakashima, "A new Lagrangian dual global optimization algorithm for solving bilinear matrix inequalities," in *ACC*, 1999.

[92] H. D. Tuan and P. Apkarian, "Low nonconvexity-rank bilinear matrix inequalities: algorithms and applications in robust controller and structure designs," *IEEE Trans. Autom. Control*, vol. 45, no. 11, pp. 2111–2117, 2000.

[93] W.-Y. Chiu, "Method of reduction of variables for bilinear matrix inequality problems in system and control designs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1241–1256, 2017.

[94] S. Kim, M. Kojima, and H. Waki, "Generalized Lagrangian duals and sums of squares relaxations of sparse polynomial optimization problems," *SIAM J. Optim.*, vol. 15, no. 3, pp. 697–719, 2005.

[95] A. A. Ahmadi and G. Hall, "Sum of squares basis pursuit with linear and second order cone programming," *Contemp. Math.*, 2017.

[96] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optim.*, vol. 11, no. 3, pp. 796–817, 2001.

[97] S. Prajna, A. Papachristodoulou, P. Seiler, and P. Parrilo, "SOSTOOLS: Sum of squares optimization toolbox for MATLAB. users guide, version 2.00," 2004.

[98] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, and P. Parrilo, *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, `http://arxiv.org/abs/1310.4716`, 2013, available from `http://www.eng.ox.ac.uk/control/sostools`,

http://www.cds.caltech.edu/sostools and http://www.mit.edu/~parrilo/sostools.

[99] D. Henrion and J.-B. Lasserre, "GloptiPoly: Global optimization over polynomials with MATLAB and SeDuMi," *ACM Trans. Math. Softw.*, vol. 29, no. 2, pp. 165–194, 2003.

[100] A. Shapiro, "First and second order analysis of nonlinear semidefinite programs," *Math. Program.*, vol. 77, no. 1, pp. 301–320, 1997.

[101] F. Zohrizadeh, M. Kheirandishfard, F. Kamangar, and R. Madani, "Non-smooth optimization over Stiefel manifolds with applications to dimensionality reduction and graph clustering," in *IJCAI*, 2019.

[102] A. A. Ahmadi, D. Malioutov, and R. Luss, "Robust minimum volume ellipsoids and higher–order polynomial level sets," in *NeurIPS workshop on optimization for machine learning*, 2014.

[103] A.-A. Liu, W.-Z. Nie, Y. Gao, and Y.-T. Su, "Multi-modal clique-graph matching for view-based 3D model retrieval," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2103–2116, 2016.

[104] L. Zhang, Y. Yang, M. Wang, R. Hong, L. Nie, and X. Li, "Detecting densely distributed graph patterns for fine-grained image categorization," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 553–565, 2016.

[105] A. Zanfir and C. Sminchisescu, "Deep learning of graph matching," in *CVPR*, 2018, pp. 2684–2693.

[106] Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli, "Graph matching networks for learning the similarity of graph structured objects," in *ICML*, 2019.

[107] H. Jiang, X. Y. Stella, and D. R. Martin, "Linear scale and rotation invariant matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011.

[108] N. Quadrianto, L. Song, and A. J. Smola, "Kernelized sorting," in *NeurIPS*, 2009.

[109] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido, "A survey for the quadratic assignment problem," *Eur. J. Oper. Res.*, vol. 176, no. 2, pp. 657–690, 2007.

[110] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *ICCV*, 2005.

[111] B. Jiang, J. Tang, C. H. Ding, and B. Luo, "Nonnegative orthogonal graph matching." in *AAAI*, 2017, pp. 4089–4095.

[112] M. Zaslavskiy, F. Bach, and J.-P. Vert, "A path following algorithm for the graph matching problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2227–2242, 2009.

[113] F. Zhou and F. De la Torre, "Factorized graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1774–1789, 2016.

[114] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 4, pp. 377–388, 1996.

[115] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," in *NeurIPS*, 2007.

[116] O. Enqvist, K. Josephson, and F. Kahl, "Optimal correspondences from pairwise constraints," in *ICCV*, 2009.

[117] M. Cho, J. Lee, and K. M. Lee, "Reweighted random walks for graph matching," in *ECCV*, 2010.

[118] B. Jiang, J. Tang, X. Cao, and B. Luo, "Lagrangian relaxation graph matching," *Pattern Recognit.*, vol. 61, pp. 255–265, 2017.

[119] M. Leordeanu, M. Hebert, and R. Sukthankar, "An integer projected fixed point method for graph matching and map inference," in *NeurIPS*, 2009.

[120] Z.-Y. Liu, H. Qiao, and L. Xu, "An extended path following algorithm for graph-matching problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1451–1456, 2012.

[121] F. Zhou and F. De la Torre, "Factorized graph matching," in *CVPR*, 2012.

[122] T. Wang, H. Ling, C. Lang, and S. Feng, "Graph matching with adaptive and branching path following," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.

[123] R. Madani, M. Kheirandishfard, J. Lavaei, and A. Atamtrk, "Penalized conic relaxations for quadratically-constrained quadratic programming," 2019.

[124] R. Zass and A. Shashua, "Probabilistic graph and hypergraph matching," in *CVPR*, 2008.

[125] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *NeurIPS*, 2002, pp. 849–856.

[126] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *CVPR*, 2011.

[127] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, 2012.

[128] S. Luo, C. Zhang, W. Zhang, and X. Cao, "Consistent and specific multi-view subspace clustering," in *AAAI*, 2018.

[129] R. Vidal and P. Favaro, "Low rank subspace clustering (lrsc)," *Pattern Recognit. Lett.*, vol. 43, pp. 47–61, 2014.

[130] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *CVPR*, 2013.

[131] M. Soltanolkotabi, E. J. Candes, *et al.*, "A geometric analysis of subspace clustering with outliers," *Ann. Stat*, vol. 40, no. 4, pp. 2195–2238, 2012.

[132] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *CVPR*, 2016.

[133] C. You and R. Vidal, "Geometric conditions for subspace-sparse recovery," in *ICML*, 2015.

[134] N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with Gaussian mixture variational autoencoders," *arXiv preprint arXiv:1611.02648*, 2016.

[135] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *ICCV*, 2017.

[136] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning deep representations for graph clustering," in *AAAI*, 2014.

[137] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *ICML*, 2016.

[138] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[139] G. Chen and G. Lerman, "Spectral curvature clustering (scc)," *Int. J. Comput. Vis.*, vol. 81, no. 3, pp. 317–330, 2009.

[140] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk, "Greedy feature selection for subspace clustering," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2487–2517, 2013.

[141] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," *IEEE Trans. Inf. Theory*, vol. 61, no. 11, pp. 6320–6342, 2015.

[142] P. Ji, M. Salzmann, and H. Li, "Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data," in *ICCV*, 2015.

[143] P. Purkait, T.-J. Chin, A. Sadri, and D. Suter, "Clustering with hypergraphs: the case for large hyperedges," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1697–1711, 2016.

[144] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *CVPR*, 2016.

[145] Y. Yang, J. Feng, N. Jojic, J. Yang, and T. S. Huang, "ℓ^0-sparse subspace clustering," in *ECCV*, 2016.

[146] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, "ClusterGAN: Latent space clustering in generative adversarial networks," *arXiv preprint arXiv:1809.03627*, 2018.

[147] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *NeurIPS*, 2016.

[148] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, 2018.

[149] X. Peng, J. Feng, J. Lu, W.-Y. Yau, and Z. Yi, "Cascade subspace clustering," in *AAAI*, 2017.

[150] J. Zhang, C.-G. Li, C. You, X. Qi, H. Zhang, J. Guo, and Z. Lin, "Self-supervised convolutional subspace clustering network," in *CVPR*, 2019.

[151] H. Gao, F. Nie, X. Li, and H. Huang, "Multi-view subspace clustering," in *ICCV*, 2015.

[152] C. Tang, X. Zhu, X. Liu, M. Li, P. Wang, C. Zhang, and L. Wang, "Learning joint affinity graph for multi-view subspace clustering," *IEEE Trans. Multimed.*, 2018.

[153] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *CVPR*, 2017.

[154] P. Ji, M. Salzmann, and H. Li, "Efficient dense subspace clustering," in *WACV*, 2014.

[155] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[156] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[157] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (cmd) for domain-invariant representation learning," in *ICLR*, 2017.

[158] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *AAAI*, 2016.

[159] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *ECCV*, 2016.

[160] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," *arXiv preprint arXiv:1603.04779*, 2016.

[161] F. M. Cariucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, "Autodial: Automatic domain alignment layers," in *ICCV*, 2017.

[162] M. Ghifary, W. B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in *PRICAI*. Springer, 2014, pp. 898–904.

[163] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *NeurIPS*, 2016.

[164] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *CVPR*, 2017.

[165] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," in *Domain Adaptation in Computer Vision Applications*. Springer, 2017, pp. 153–171.

[166] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *ECCV*, 2016.

[167] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *CVPR*, 2017.

[168] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *CVPR*, 2018.

[169] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *CVPR*, 2018.

[170] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *ICML*, 2018.

[171] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *NeurIPS*, 2007.

[172] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," *arXiv preprint arXiv:1702.08400*, 2017.

[173] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *ECCV*, 2010.

[174] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *CVPR*, 2017.

[175] F. Zohrizadeh, M. Kheirandishfard, and F. Kamangar, "Class subset selection for partial domain adaptation," in *CVPRW*, 2019.

[176] A. Paszke, S. Gross, S. Chintala, and G. Chanan, "Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration," *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*, vol. 6, 2017.

[177] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[178] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis*, vol. 115, no. 3, pp. 211–252, 2015.

[179] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *J. Mach. Learn. Res.*, vol. 8, pp. 985–1005, 2007.

[180] L. Van Der Maaten, "Barnes-hut-sne," in *ICLR*, 2013.