On the Reliability and Robustness of Linear Generalized Regression Algorithms for Classification

Jiaqi Bao

Graduate School of Information Science and Technology

Hokkaido University

A thesis submitted for the degree of

doctor of information engineering (PhD)

2024 April

Abstract

Machine Learning (ML) continues to progress in various application areas such as healthcare, finance, and autonomous vehicles, where the accuracy of ML models is crucial due to the severe consequences that errors can cause, including harm to humans. The reliability of these models is paramount, particularly when dealing with imperfect data, which is often compromised by various factors: insufficient information due to the high costs associated with manual labeling, data bias stemming from changing environments and privacy concerns, label noise caused by human or sensor errors, and vulnerability to attacks like adversarial noise and distribution shifts. Despite substantial research efforts in addressing imperfect data challenges in ML, the existing methods have several limitations, such as unsupervised outlier removal can eliminate anomalies but may discard valuable data, reducing classification accuracy. Robust Huber loss, known to be effective in regression, performs less well in classification. Regularization techniques mitigate overfitting but struggle with noise. This thesis presents methodologies equipped with robust loss functions that are effective in both classification and regression. Noise-against-regularization techniques and strategies for handling label noise are also addressed. It is structured into three distinct parts, each focusing on addressing specific imperfections in real-world data: Part I focuses on Semi-Supervised Learning (SSL) under limited labeled data, Part II on Transfer Learning (TL) across different data domains, and Part III on Learning with Label Noise, for improving ML model precision and generalizability.

For Part I of the thesis, Semi-Supervised Learning (SSL) frequently confronts the challenge of having only a limited amount of labeled data available for training. Furthermore, the presence of data noise, arising from

various sources such as measurement errors or data collection imperfections, can undermine the accuracy of SSL models. To tackle these issues, we introduce Robust Embedding Regression (RER). RER achieves this by constructing a robust graph that adaptively adjusts weights for each data point, effectively reducing the influence of data noise on the learning process. Additionally, RER incorporates a low-rank representation to enhance the utilization of limited labeled data and mitigate the impact of redundant features. Further robustness is achieved through the introduction of appropriate norms to both reconstruction and regularization terms, facilitating feature and sample selection. Our method has been proven through extensive experiments to maintain a classification rate of over 46.67% on datasets with varying degrees of random noise or continuous noise, representing a 32.67% improvement over comparative semi-supervised methods.

In Part II, domain shift introduces disparities and variations in the data that can hinder effective knowledge transfer. To address the imperfections resulting from domain shift, we introduce the Redirected Transfer Learning (RTL) approach. By reconstructing target samples using the lowest-rank representation obtained from source samples, RTL effectively mitigates the impact of domain shift on data imperfections. Additionally, RTL incorporates the $L_{2,1}$-norm sparsity on the reconstruction and regularization terms to enhance robustness against data variations. RTL also introduces a redirected label strategy that transforms binary labels into continuous values, aiding adaptation to the diverse data distributions resulting from domain shift. The superiority of our method in classification tasks is confirmed on several cross-domain datasets, for example, RTL attained about 5%-10% improvements on average compared to the latest published methods.

Part III of the thesis tackles challenges in Partial Multi-Label Learning (PML), where instances are associated with the incomplete labeling of data, making it difficult to build accurate predictive models. Traditional PML methods, which utilize pre-defined graphs for label disambiguation, lack adaptability to changing data relationships and diminished effectiveness in scenarios with label uncertainty. Common two-step graph-based

approaches, involving static graph construction and subsequent label propagation, often result in suboptimal label confidence learning. Addressing these limitations, our research proposes a novel framework named Adaptive Dual Graph Disambiguation (ADGD) that simultaneously learns dual adaptive graphs and a sparse projection matrix. These graphs, one capturing instance interrelationships and the other focusing on label correlations, are dynamically updated to better handle label noise and enhance label confidence. The integration of $L_{2,1}$ norm in both the regression and regularization terms introduces robustness and ability of feature selection. Additionally, the sparsity of projection potentially contributes to reducing label ambiguity, further refining the label disambiguation process in PML. Extensive experiments conducted on databases with noises in both feature space and label space have confirmed the superiority of the proposed methods.

Acknowledgements

As I culminate my doctoral journey, the overwhelming sense of gratitude I feel towards those who have been instrumental in my academic and personal growth cannot be overstated. First and foremost, I extend my deepest appreciation to Prof. Mineichi Kudo, whose guidance, expertise, and unwavering support in the field of pattern recognition and machine learning have been the cornerstone of my research. His wisdom and mentorship have not only shaped my academic pursuits but also my approach to complex problems.

I am equally indebted to my co-supervisor, Dr. Keigo Kimura, whose insights and constructive critiques have significantly enriched my work. His perspectives and encouragement have been invaluable throughout this journey. Additionally, my profound thanks go to Prof. Hideyuki Imai, Prof. Akira Tanaka, and Prof. Atsuyoshi Nakamura, whose expertise and guidance have been vital in refining my research and broadening my academic horizons. Their contributions to my journey have been indispensable.

My heartfelt thanks go to my homeland, China. The rich cultural heritage and the values I have inherited from this great nation have been a constant source of inspiration and strength. I am profoundly grateful for the foundation it has provided me, fostering a spirit of perseverance and dedication in my studies and research.

Above all, I owe an immeasurable debt of gratitude to my parents 包静 and 邹艳. Their unconditional love, sacrifice, and belief in my abilities have been the driving force behind my every endeavor. Their unwavering support and encouragement have been my guiding light, illuminating the path through the toughest challenges.

To all my colleagues, friends, and everyone who has been a part of this journey, your support and companionship have made this experience not only successful but also enjoyable. Thank you for being an integral part of my story.

Contents

List of Figures **xiii**

List of Tables **xvii**

1 Introduction **1**

1.1 Background 1

1.2 Motivations 4

1.3 Thesis Organization 5

2 Literature Review **7**

2.1 Semi-supervised learning with limited labeled data 7

2.1.1 Preliminaries 7

2.1.2 Related Works 8

2.2 Transfer Learning for Cross-Domain Data 10

2.2.1 Preliminaries 10

2.2.2 Related Works 11

2.3 Complex Challenge for Label-Based Learning 12

2.3.1 Preliminaries 12

2.3.2 Related Works 14

3 Research Methodology **17**

3.1 Overview of Proposed Reliable and Robust Methods for Imperfection Data Classification 17

3.2 Research Methodology of Semi-Supervised Learning 20

3.2.1 Definitions of Semi-Supervised Learning 21

3.2.2 Robust Linear Regression 22

3.2.3 Manifold-Based Semi-Supervised Learning 22
3.2.4 Low-Rank Representation 23
3.3 Research Methodology of Transfer Learning 24
3.3.1 Definitions of Transfer Learning 24
3.3.2 Domain Adaptation with Low-Rank Constraint 25
3.3.3 Linear Regression and Relaxed-Label-Based Regressions 26
3.4 Research Methodology of Partial Multi-Label Learning 27
3.4.1 Definitions of Partial Multi-Label Learning 27
3.4.2 Disambiguation with Fixed Graph 27
3.4.3 Adaptive Graph Embedding 29

4 Robust Regression Embedding for Semi-Supervised Learning 31
4.1 Preliminaries 31
4.2 Robust embedding regression for semi-supervised learning 32
4.2.1 Formulation of RER 33
4.2.2 Support Examples 35
4.2.3 Optimization Solution 36
4.2.4 Computational Complexity and Convergence Analysis 41
4.2.5 Difference From the Existing Works 42
4.3 Experiments 43
4.3.1 Datasets 43
4.3.2 Experimental Settings 44
4.3.3 Semi-Supervised Classification 46
4.3.4 Semi-Supervised Clustering 47
4.3.5 Experimental Results and Analysis 49
4.3.6 Visualization Experiments 50
4.3.7 Parameter Sensitivity and Convergence Analysis 51
4.3.8 Ablation Analysis 51
4.4 Discussion 53
4.5 Conclusion 54

5 Redirected Transfer Learning for Robust Multi-Layer Subspace Learning 55

5.1 Motivations . . . 55
5.1.1 Model Formulation . . . 56
5.1.2 Optimization . . . 58
5.1.3 Computational Complexity . . . 61
5.1.4 Convergence Analysis . . . 62
5.1.5 Differences from Previous Work . . . 62
5.2 Experiments . . . 64
5.2.1 Datasets Introduction . . . 65
5.2.2 Experimental Setting . . . 66
5.2.3 Classification and Evaluation Metric . . . 67
5.2.4 Experimental Results . . . 71
5.2.5 Convergence and Parameter Sensitivity Analysis . . . 72
5.2.6 Time Comparison . . . 73
5.2.7 Ablation Studies . . . 76
5.3 Conclusion . . . 78

6 Partial Multi-label Learning with Adaptive Dual Graph Disambiguation 81

6.1 Preliminaries . . . 81
6.2 Formulation of ADGD . . . 82
6.3 Alternative Optimization . . . 83
6.3.1 Convergence Proof . . . 85
6.3.2 Time Complexity Analysis . . . 87
6.4 Experiments . . . 87
6.4.1 Experimental setup . . . 87
6.4.2 Competing Algorithms . . . 89
6.4.3 Experimental Results . . . 91
6.4.4 Sensitivity Analysis . . . 94
6.4.5 Convergence and Complexity Analysis . . . 98
6.4.6 Ablation Study . . . 98
6.5 Conclusion . . . 100

7 Conclusion and Future Work **101**

7.1 Summary of the Thesis 101

7.2 Future Work 102

References **105**

List of Figures

1.1 A 3D scatter plot of a torus dataset used in semi-supervised learning. Torus dataset contains 100 data points for each class, yet only 3 data points per class are labeled (● and ●). The remaining data points are unlabeled (○ and ○), demonstrating the challenge of working with extremely limited labeled data in a semi-supervised learning context. . . 2

1.2 Samples from the VisDA-2017 dataset. The top row shows synthetic renderings of various vehicles and the bottom row shows their real-world counterparts. Each row represents a class category from two domains which exhibit a diverse array of lighting, angles, and backgrounds, highlighting the significant variation and challenges involved in transfer learning across different domains. 3

1.3 Example of Label Noise in Multi-Label Classification. An image labeled with multiple concepts, including 'ocean', 'beach', 'airplane', 'mountain', 'boat', and 'person'. However, 'airplane', 'boat', and 'person' serve as examples of label noise—incorrect annotations that can mislead the learning process. 4

2.1 Semi-supervised learning frameworks 8

2.2 Transfer learning framework 11

2.3 Four kinds of machine learning frameworks 13

LIST OF FIGURES

3.1 Architecture of the proposed reliable machine learning framework of four phases: (1) imperfection dataset construction, (2) semi-supervised learning with robust embedding regression method, (3) transfer learning with redirected transfer learning method, and (4) Partial multi-label learning with feature selection method. 18

3.2 Image data classification with robust embedding regression method. . . 19

3.3 Data classification with the redirected transfer learning method. 20

3.4 Partial multi-label learning with adaptive dual graph disambiguation. . 21

4.1 The development routes of related work. Our method inherits the functions and advantages of the existing methods, including label propagation (LP), manifold learning (ML), dimension reduction (DR), feature selection (FS), robustness against noises (RB), data recovery (DRY) and sample selection (SS), for robust semi-supervised classification and clustering. The blue boxes present the properties of each method. 32

4.2 Three images from the YaleB dataset. 33

4.3 Illustration of various visualization results and matrices. 34

4.4 Sample images. From top to bottom, AR, COIL20, LFW, CMU PIE, and YaleB datasets. 43

4.5 Some examples of corrupted images under varying levels of contiguous occlusions and different percentages of salt-and-pepper noises. 44

4.6 The example of using RER to recover the corrupted YaleB face images. Left: the contaminated matrix X. Middle: the corrected data XZ. Right: the error E. 48

4.7 Visualization by tSNE of the COIL20 dataset of 20 classes. (a) The original sample distribution, (b) the sample distribution learned by FME, and (c) RER. Different colors represent different classes. 50

4.8 Sensitivity analysis of parameters λ_1 (regularization term), λ_2 (label propagation term), and λ_3 (error term) of RER on AR dataset ($p = 6$). One parameter was changed, whereas the other two were fixed empirically. 51

4.9 Convergence of RER on (a) AR ($p = 4$), (b) COIL20 ($p = 5$), (c) LFW ($p = 6$), and YaleB ($p = 6$). 52

5.1 Projections learned by TSL_LRSR, LET, and RTL for the task $C1 \rightarrow C2$. Note: all figures are shown in the HSV color space. For better comparison, we only plot the first 100 rows of these projections. From the colorbar, we can infer all element values of these projections. It is seen that the element values in some rows are equal or approximate to zeros in (c). 63

5.2 Images samples of different datasets, including (a) 4DA dataset, (b) CMU PIE dataset with clean images and corrupted images, (c) COIL20 dataset. 67

5.3 The t-SNE visualization of (a) original data, and the extracted features generated by (b) TSL_LRSR, (c) LET, (d) RTL, on the C1→C2 task respectively. In the first column, the red cross 'x' denotes source samples of $C1$ domain, the blue hollow circle '$\circ$' denotes target samples of $C2$ domain, and in the second column, the solid circle '$\bullet$' with 20 colors denotes the samples corresponding to 20 classes. 70

5.4 Convergence curves on the selected cross-domain datasets. (a) Task D→W in 4DA, (b) Task P3→P4 in CMU PIE, (c) Task C2→C1 in COIL20, (d) Task pla→peo in Reuters-21578. 74

5.5 Sensitivity of RTL to its parameters λ_1 and λ_2. (a) 4DA (SURF feature), (b) 4DA (DeCAF7 feature), (c) COIL20, and (d) CMU PIE datasets. . 75

5.6 The t-SNE visualization of RTL on the $A \rightarrow W$ task. The solid circle '$\bullet$' with 10 colors denotes the samples corresponding to the different classes. There are 10 classes in the 4DA dataset. 77

6.1 Comparison of ADGD against comparing methods with the Nemenyi test. Methods not connected with ADGD in the CD diagram are considered to have a significantly different performance from ADGD (CD = 3.29 at 0.05 significance level) 90

6.2 Experimental results on data set Yeast with 200% noisy labels. 95

6.3 Experimental results on data set Birds with 10% feature noises. 96

6.4 Results of ADGD with varying value of trade-off parameters on Mirflickr. 97

6.5 convergence curve on scene and emotions datasets with 100% label noises. 98

6.6 ADGD and its variant methods on Birds dataset with 20% feature noises and 200% label noises. 99

List of Tables

4.1 Comparison of the distance between the images using different norms. $Dist(A, B) = \|A - B\|_{a,b}$ according to chosen norm. 35

4.2 The summary of objectives of the proposed method and the most related methods 42

4.3 Characteristics of five benchmark datasets. 43

4.4 Accuracy rate and standard deviations (%) of several methods on three image datasets with the different number of labels (p represents the number of labeled data per class and bold fonts mark the best performance). 46

4.5 Accuracy rate and standard deviations (%) on the YaleB dataset with different corruption percentages ($p = 30$). 47

4.6 Accuracy rate and standard deviations (%) on the YaleB dataset with different occlusion sizes of corruption ($p = 30$). 47

4.7 Accuracy rate and standard deviations (%) on the COIL20 dataset with different base classifiers ($p = 10$). The bandwidth of RBF kernel is 3. . 47

4.8 Predicted accuracy (PreACC) rate and standard deviations (%) of several methods on three image datasets with the different number of labels (p represents the number of labeled data per class and bold fonts mark the best performance). 48

4.9 Accuracy rate and standard deviations (%) on the CMU PIE dataset with different occlusion sizes of corruption. 53

5.1 Detailed information of different datasets 66

5.2 Accuracy(%) on the 4DA dataset with SURF features. The best is typed in boldface and * denotes the pseudo-target label-based method. 68

5.3 Accuracy(%) on the 4DA dataset with DeCAF7 features. The best is typed in boldface, and * denotes deep learning methods. 68
5.4 Accuracy(%) on the CMU PIE dataset corrupted by 5 × 5 block size occlusions. The best is typed in boldface, and * denotes label relaxation strategy-based methods. 69
5.5 Accuracy(%) on the COIL20 dataset. The best is typed in boldface . . . 69
5.6 Accuracy(%) on the Reuters-21578 dataset. The best is typed in boldface 71
5.7 Run time (s) comparisons of different methods 73
5.8 Accuracy(%) on the 4DA dataset with SURF features. The best is typed in boldface 76

6.1 Characteristics of the PML experimental datasets. 87
6.2 Friedman statistics F_F in terms of each evaluation metric and the critical value at 0.05 significance level (# comparing algorithms $K = 9$, # datasets $N = 11$). 91
6.3 Comparison results of methods (mean±std.deviation) in terms of hamming loss. The best results are highlighted in bold, and the number in the bracket indicates the ranking of this algorithm. The last row shows the average ranking of each algorithm on each evaluation metric 92
6.4 Comparison results of methods (mean±std.deviation) in terms of average precision. The best results are highlighted in bold, and the number in the bracket indicates the ranking of this algorithm. The last row shows the average ranking of each algorithm on each evaluation metric 92
6.5 Comparison results of methods (mean±std.deviation) in terms of ranking loss. The best results are highlighted in bold, and the number in the bracket indicates the ranking of this algorithm. The last row shows the average ranking of each algorithm on each evaluation metric 93
6.6 Comparison results of methods (mean±std.deviation) in terms of coverage. The best results are highlighted in bold, and the number in the bracket indicates the ranking of this algorithm. The last row shows the average ranking of each algorithm on each evaluation metric 93

1

Introduction

1.1 Background

Machine Learning (ML) has seen remarkable advancements in recent years, revolutionizing industries with its ability to process and learn from vast amounts of data, particularly in areas such as healthcare, finance, and autonomous vehicles. In healthcare [1], ML algorithms are being utilized for diagnostic purposes, drug discovery, and personalized treatment plans. In finance [47], ML models are applied to fraud detection, risk management, and algorithmic trading. In the autonomous vehicles sector, ML is crucial for navigation, obstacle detection, and decision-making [5].

However, the dependence on ML in these fields introduces significant risks. For example, Figure 1.1, showcasing the torus dataset with extremely sparse labeled points, vividly illustrates the challenge of limited labeled data availability in machine learning, particularly in semi-supervised learning scenarios. In this dataset, each class comprises 100 data points, only 3 points per class are labeled. This situation mirrors real-world scenarios where acquiring labeled data is often costly and labor-intensive, particularly in expert-driven sectors like healthcare. The scarcity of labels can result in models that are poorly trained and outperform on new, unseen data. To address this, self-training techniques [6], where the model iteratively labels the unlabeled data, are used to expand the training dataset and improve model generalization. Co-training [111] proposed to utilize two different views of the data to label more examples, capitalizing on the agreement between different algorithms. Recent research focuses on graph-based methods [64], which build a graph that captures the relationships between labeled and

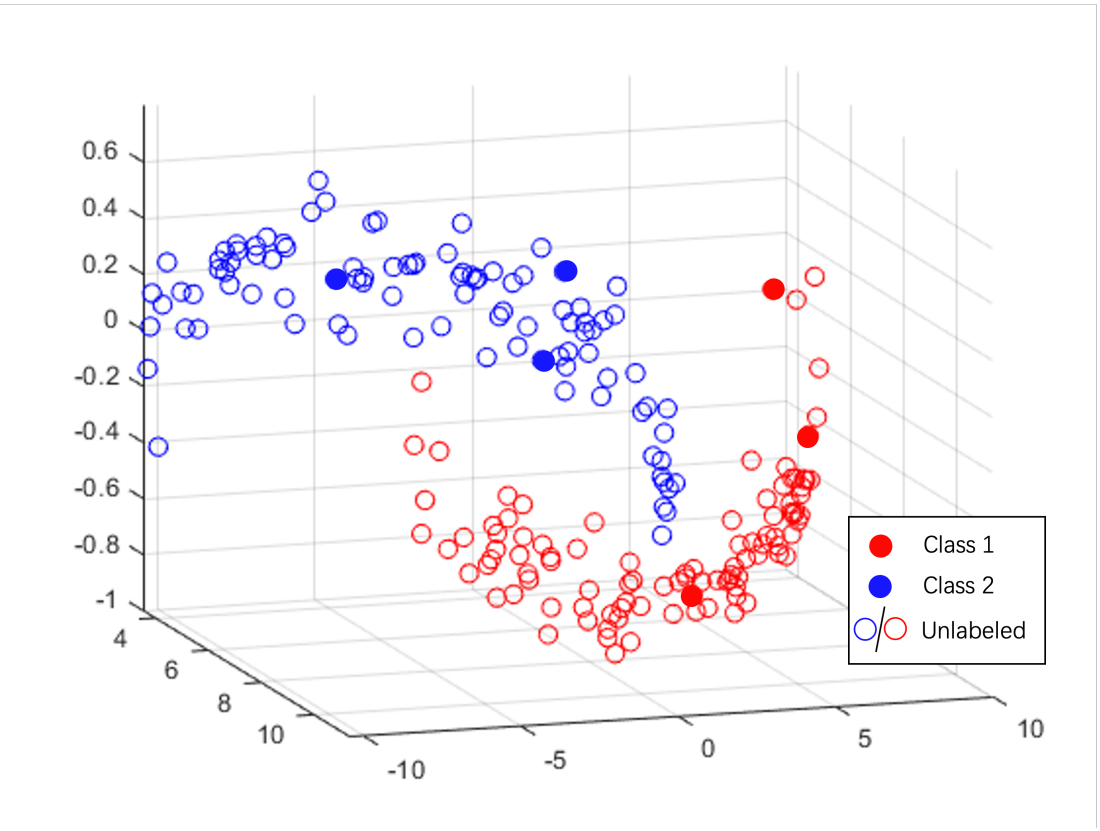


Figure 1.1: A 3D scatter plot of a torus dataset used in semi-supervised learning. Torus dataset contains 100 data points for each class, yet only 3 data points per class are labeled (● and ●). The remaining data points are unlabeled (○ and ○), demonstrating the challenge of working with extremely limited labeled data in a semi-supervised learning context.

unlabeled data points to propagate labels.

Furthermore, as illustrated in Fig. 1.2, we often utilize data from distinct domains, such as the VisDA-2017 dataset which includes synthetic renderings of 3D models and real-world images. This comparison exemplifies the difference between data domains in machine learning. The synthetic images could be from a simulated environment or a video game, which often have simplified textures and shapes. In contrast, the real images come from the physical world and contain a higher level of complexity, including varied lighting, shadows, textures, and backgrounds. Transfer learning emerges as a crucial methodology to surmount this challenge by applying knowledge gained in one domain (e.g., synthetic data) to another domain (e.g., real-world data). Models trained

Figure 1.2: Samples from the VisDA-2017 dataset. The top row shows synthetic renderings of various vehicles and the bottom row shows their real-world counterparts. Each row represents a class category from two domains which exhibit a diverse array of lighting, angles, and backgrounds, highlighting the significant variation and challenges involved in transfer learning across different domains.

on synthetic data might not perform well when directly applied to real-world data due to huge differences. Transfer learning allows these models to adapt to the new domain by fine-tuning the learned representations or features with a (usually smaller) set of labeled real-world data, thereby achieving better performance than training from scratch on the new domain [52].

In the context of learning with label noise, as illustrated by Figure 1.3, training data may include incorrect labels—such as the mislabeling of an image with an ocean, beach, and mountain as also containing an airplane, a boat, and a person. This label noise can severely degrade the performance of machine learning models. Consequently, noise-robust training algorithms are essential for mitigating the adverse effects of inaccurately labeled data. These algorithms employ robust loss functions specifically designed to be less sensitive to label noise [18]. Additionally, noise cleaning techniques [79] are employed to identify and remove or correct mislabeled data points, thus purifying the training set. Some approaches go a step further by modeling label noise directly within the training process [62], enabling the model to account for the possibility of incorrect labels and adjust its learning trajectory accordingly. By integrating these strategies, it's possible to enhance model robustness and maintain high performance even when label noise is present.

All these issues are part of what is commonly referred to as the *imperfect data problem*, and ensuring the accuracy and dependability of ML models in the face of such challenges is of the utmost importance.

Figure 1.3: Example of Label Noise in Multi-Label Classification. An image labeled with multiple concepts, including 'ocean', 'beach', 'airplane', 'mountain', 'boat', and 'person'. However, 'airplane', 'boat', and 'person' serve as examples of label noise—incorrect annotations that can mislead the learning process.

1.2 Motivations

In the preceding section, we tackled the issue of imperfect data—a prevalent challenge in the field of machine learning. Numerous methods have been developed to deal with various forms of data imperfection, achieving notable success in enhancing model performance. Nonetheless, these current methodologies exhibit a range of deficiencies that necessitate further investigation and innovation. Among the limitations of current approaches are:

- **Semi-Supervised Learning:** Although current semi-supervised learning methods have effectively utilized both labeled data and unlabeled data, most methods perform badly when confronted with datasets that contain considerable data noise and redundant information. There's a pressing need to refine these methods to reduce their sensitivity to noise and outliers in feature space and to harness label information more effectively, particularly within an affinity graph for low-dimensional embedding. These challenges have yet to be adequately addressed within the semi-supervised learning framework.

- **Transfer Learning:** Many existing transfer learning techniques employ a 0-1 matrix for labels, which significantly restricts the flexibility of the learning pro-

cess. Furthermore, a major limitation of these methods is their susceptibility to the redundant features and noise present in cross-domain data. Such challenges hinder the effective application of transfer learning in diverse and noisy environments, necessitating the development of more adaptable and robust approaches.

- **Label Noise-Resistant Learning:** Rather than only focusing on the presence of noise within labels, this paper acknowledges that noisy and redundant features within the data can also exacerbate the effects of incorrect labels. Current methods do not adequately counter this issue, leading to a need for more robust approaches that can perform feature selection and extraction in the presence of such noise.

Motivated by these challenges, our research proposes innovative machine learning solutions designed to be robust against the multifaceted nature of data imperfections. By enhancing the ability to learn from noisy data, improving the transfer of knowledge across different domains, and refining the process of dealing with label noise, we strive to create algorithms that can yield accurate and reliable results in the face of imperfect data.

1.3 Thesis Organization

According to the motivations mentioned previously, the remainder of this thesis consists of 6 chapters: Chapter 2 reviews the related works from three domains, Chapter 3 introduces the detailed research methodology, Chapter 4 presents a novel semi-supervised learning method, Chapter 5 proposes a novel method in transfer learning, Chapter 6 addresses the problem of mislabeling and feature noise by designing a novel method. learning method. Finally, Chapter 7 concludes this thesis and discusses the future work motivated by the thesis.

2

Literature Review

In the realm of machine learning, the presence of imperfect data poses a significant threat to the efficacy and reliability of algorithms. As machine learning increasingly permeates various aspects of technology and society, the robustness of these algorithms against the complexities of data becomes paramount. Imperfect data, characterized by limited labeled samples, similar yet distributionally diverse data from different domains, or data annotated with multiple labels that may contain noisy or irrelevant labels, presents a unique challenge. Numerous methods have been proposed to enhance the trustworthiness of algorithms when dealing with such data. These methods have shown promising results in addressing imperfect datasets and thus in this chapter, we will review these approaches from three perspectives: semi-supervised learning, transfer learning, multi-label learning, and partial multi-label learning.

2.1 Semi-supervised learning with limited labeled data

2.1.1 Preliminaries

In semi-supervised learning, the algorithm utilizes the labeled instances to learn the underlying patterns and applies this knowledge to predict the labels of unlabeled instances. It's a blend of supervised and unsupervised learning approaches, leveraging the strengths of both to improve learning efficiency, especially when acquiring labeled data is costly or impractical. Assume we have a set of instances (like images) and potential labels (A, B, C, D). In a semi-supervised learning scenario, some instances are labeled while others are not. This can be represented in a matrix as in Figure 2.1,

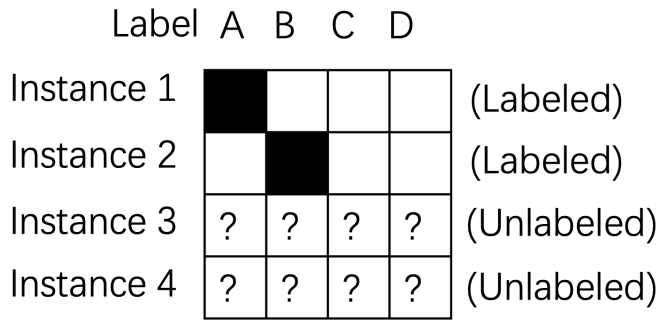


Figure 2.1: Semi-supervised learning frameworks

where Instance 1 is labeled with A and Instance 2 with B while Instances 3 and 4 do not have labels. The goal in semi-supervised learning is to predict their labels based on the labeled data and inherent data patterns.

2.1.2 Related Works

We first begin with a review of the traditional supervised learning methods, which is the foundation and motivation for proposing semi-supervised learning.

Traditional supervised learning, such as Least squares regression (LSR), which aims to learn the proper mapping of samples to a new space for regression and is one of the most popular methods in machine learning and image classification. Based on the simplicity and effectiveness of LSR, various modified methods have been proposed in the past few decades. The most representative method is the linear regression (LR) [13]. Many variants have been developed to overcome the potential disadvantages of LSR, such as weighted least squares regression (WLSR) [34], discriminative least squares regression (DLSR) [86], robust regression (RR) [26] and partial least squares regression (PLSR) [78]. To overcome the problem of sensitivity to noises of LR methods, Nie et al. [48] extended LR for feature selection and proposed a robust feature selection (RFS) framework by considering robust norm as a basic metric.

In many real-world applications, only a small number of samples are labeled since labeling requires a significant amount of human labor and time, such as face recognition [68], person re-identification [40], and image retrieval [107]. To overcome this difficulty, semi-supervised learning (SSL) has attracted significant attention because it can adequately utilize both labeled and unlabeled data. Gaussian fields and harmonic

functions (GFHF) [112] as well as local and global consistency (LGC) [108] are among the most popular SSL methods that construct a weight affinity graph based on the data distribution to propagate the information of labeled samples to unlabeled samples. A larger weight value indicates a higher probability of the paired points being in the same class. Many manifold-based SSL methods have been proposed based on GFHF and LGC, such as Laplacian regularized least squares (LapRLS/L) [4], flexible manifold embedding (FME) [49] and semi-supervised orthogonal graph embedding (SOGE) [39]. Qiu et al. [57] extended FME and proposed a fast FME (f-FME) method by building an anchor graph for accelerating FME and reducing the computation cost. Besides, Nie et al. [51] proposed a method named semi-supervised adaptive local embedding learning (SALE) which adaptively constructs two affinity graphs based on labeled data and all embedding samples separately to explore the local and global structure of data. Nevertheless, these methods ignore the fact that there are abundant irrelevant and even noisy features in real-world raw data. Thus, the selection of neighbors and features for each sample is critical for semi-supervised classification performance improvement. To address this problem, Chen et al. [10] proposed a semi-supervised learning method called rescaled linear square regression (RLSR) that introduces a rescaled projection matrix to rank the importance of each feature. Based on this idea, Nie et al. [50] designed an auto-weighting semi-supervised learning (AWSSL) method which introduces an auto-weighting matrix for jointly performing label propagation and feature selection.

In general, there are many strategies for manifold-based semi-supervised methods to construct an affinity graph for exploring the latent correlation between labeled and unlabeled samples, such as k nearest neighbor (KNN) [110] and local linear representation [73]. However, these methods suffer from two fundamental problems. First, they are sensitive to noise and outliers. The weight graph is unreliable when the distribution of raw data is corrupted. In this case, it is impossible to assign correct labels to the unlabeled set by the incorrect graph's guide. Second, these methods perform graph construction and label propagation separately. To be specific, the weight graphs exploited to guide label prediction are calculated by neighborhood relationships in the feature space without label information, that is, in an unsupervised manner. Thus, it cannot be guaranteed that the learned affinity graph accurately reflects the relationship between the labels of data samples. Recently, low-rank representation (LRR) has been widely used for manifold learning due to its robustness to the noises or outliers

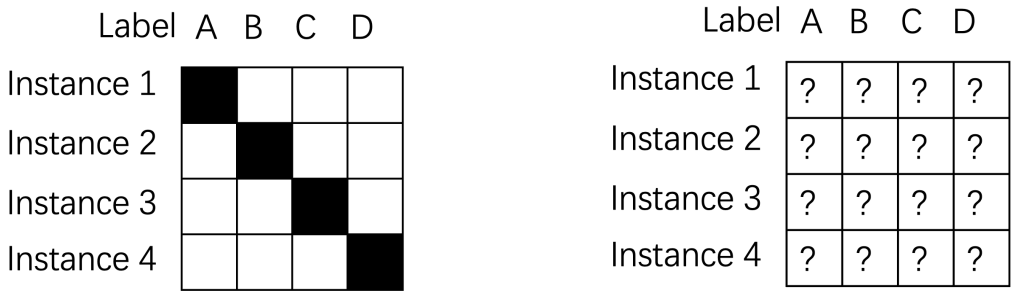
in original data [35, 43, 71]. Some methods first leveraged LRR to decompose the corrupted data into a clean matrix and a noisy matrix and then improved affinity graph construction by exploiting clean data recovered by LRR. For example, Wen et al. [80] proposed a novel graph learning method named low-rank representation with adaptive graph regularization (LRR_AGR) to adaptively learn such an ideal graph from data. Besides, for semi-supervised learning, some studies combined LRR and label propagation into a unified framework that can perform weight graph construction and semi-supervised learning simultaneously, such as non-negative low-rank representation (NNLRR) [81], graph regularized low-rank representation method for semi-supervised learning (GLR2S2) [91], robust graph learning (RGL) [31] and robust semi-supervised multi-view graph learning (RSSMvSI) [21].

2.2 Transfer Learning for Cross-Domain Data

2.2.1 Preliminaries

Traditional subspace learning methods generally assume that the training data and testing data lay on a lower but the same feature subspace with independent identically distribution (i.i.d.) [30]. However, in many real-world applications, there is a serious inconsistency between training and testing data distributions, which leads to dramatic performance degradation in classification tasks [52]. For example, it is quite a challenge to use a set of labeled childhood photos to recognize a person in his/her adult photos.

In transfer learning, particularly in the context of domain adaptation, the goal is to apply knowledge acquired from one domain (the source domain) to a different, yet related domain (the target domain). To illustrate this, consider two datasets represented in matrix form in Figure 2.2b, where instances are linked to a common set of labels (A, B, C, D), but the distribution of these labels may vary between the two domains. In the source domain, the matrix contains labeled data used for training the model. Each row represents an instance, and the columns correspond to labels. The target domain matrix consists of unlabeled data, denoted by '?' symbols, reflecting the unknown labels in this new domain. Transfer learning is particularly beneficial when there is a scarcity or complete absence of labeled data in the target domain. It significantly reduces the need for manually labeling a new dataset and enhances the model's performance in the new domain.



(a) Source domain dataset **(b)** Target domain dataset

Figure 2.2: Transfer learning framework

2.2.2 Related Works

In the past decade, transfer learning [15, 61, 95] has gained a great deal of attention which can effectively address the problem of different distributions between the source data (training set) and target data (testing set). It has been exploited in image domain adaptation [83], activity recognition [55], and reinforcement learning [105]. The existing transfer learning methods are divided into two categories: modifying the classifier for adapting to the variant distributions of domains (classifier-based methods) [33] and changing the representation of data during the process of transfer learning (representation-based methods). In this paper, we focus on the latter category.

To pursue the representation-based methods, one of the best ways is to seek a common subspace to both data of the source and target domains in such a way that the subspace minimizes the discrepancy between different domains. During this process, maximum mean discrepancy (MMD) is exploited in general to measure the distribution difference between the source and target domains. For example, Pan et al. [53] proposed a transfer component analysis (TCA) method to reduce the discrepancy between the marginal distributions of different domains by using MMD as a non-parametric discrepancy metric. Similarly, Zhang et al. proposed a joint geometrical and statistical alignment (JGSA) [92] method which jointly considered the geometrical shift and distribution shift. Besides, Si et al. [60] proposed a transfer subspace learning (TSL) method, which applies the subspace learning algorithms to transfer the knowledge gained in training samples to testing samples. Furthermore, Long et al. [41] proposed a joint distribution alignment (JDA) method that extends MMD to simulta-

neously measure the difference in marginal and conditional distributions. Han et al. [22] proposed a latent elastic-net transfer learning (LET) method by exploring a latent subspace and gaining better alignment of source and target domains with MMD.

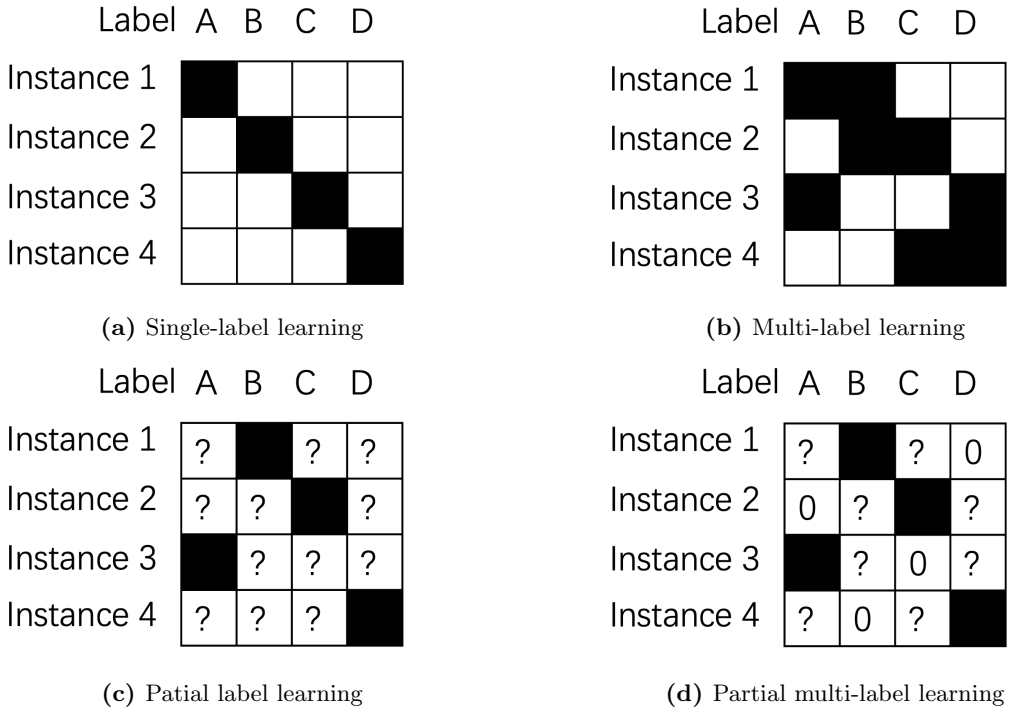
Recent research imposes a low-rank constraint (LRC) [70, 71] on the cross-domain representation matrix to reduce the domain discrepancy. For example, Shao et al. [59] proposed to handle the subtle differences between the source and target by reconstructing the target data with the lowest-rank representation of source data in the low-dimensional space. Furthermore, Xu et al. [89] proposed to use a sparse and low-rank representation for preserving the global and local structures of data during transfer. Besides, Zhang et al. [96] proposed a guide subspace learning (GSL) method that combines LRC with subspace learning so that each target sample can be represented by the source samples with a low-rank coefficient matrix in a common subspace.

Note that most of these methods exploit conventional zero-one label matrix as the regression target. However, this fixed-value label matrix is not optimal in reducing the disparity of the domain distributions. To solve this problem, researchers proposed to exploit an ε-dragging technology [86] that relaxes the binary label matrix to a more discriminative regression target during transfer [22, 89, 96]. It gives labels larger than one for the true class and less than zero for the false class. In most cases, this technology brings a better performance. However, this is not the case when data are with noises and redundant features. This is because (1) it focuses on minimizing the gap between different domains while ignoring selecting important features from the original high-dimensional data for feature extraction. (2) It is sensitive to noises which may destroy the structure of the original data and, thus, lead to overfitting.

2.3 Complex Challenge for Label-Based Learning

2.3.1 Preliminaries

In the contemporary realm of machine learning, the handling of imperfect data manifests in various forms, presenting a complex challenge for robust classification. This complexity is particularly evident when navigating through the nuances of label assignment in datasets. The complexity of this task can be categorized into four distinct paradigms, each addressing a unique aspect of label ambiguity and complexity. To il-



(a) Single-label learning

(b) Multi-label learning

(c) Patial label learning

(d) Partial multi-label learning

Figure 2.3: Four kinds of machine learning frameworks

lustrate these paradigms, consider a dataset where instances are images, and potential labels are denoted as A, B, C, and D.

Single-Label Classification represents the simplest form, as shown in Figure 2.3 (a), where each instance is associated with a unique label. This paradigm, while straightforward, often falls short in encapsulating the complexities of real-world data, as it assumes a one-dimensional label space per instance.

Multi-Label Learning (MLL) extends this concept to scenarios where instances are inherently multi-dimensional, characterized by multiple labels. As illustrated in Figure 2.3 (b), this paradigm is more aligned with complex data structures, as it allows for a richer and more nuanced representation, capturing the multiple attributes that a single instance might exhibit.

Partial Label Learning (PLL) introduces uncertainty into the classification process in Figure 2.3 (c). In PLL, each instance is associated with a set of candidate labels, but only one is correct. This paradigm addresses scenarios where label assignments are not definitive and are subject to verification, reflecting a common challenge in datasets

with noisy or incomplete labeling.

Expanding upon this complexity, **Partial Multi-Label Learning** (PML) presents a scenario where each instance is associated with multiple candidate labels, among which several may be correct. This paradigm is indicative of real-world situations where instances are often partially labeled, containing a mixture of relevant and irrelevant labels. In the PML matrix representation in Figure 2.3 (d), '?' symbols denote the uncertainty of each label's relevance, encapsulating the essence of imperfect data with multiple possible truths.

These methods together represent the range of label-based classification strategies in machine learning, each designed for varying complexities and uncertainties of data. They highlight the ongoing challenge of handling imperfect data in sophisticated machine learning tasks. This is a vital research area for developing more robust models.

2.3.2 Related Works

Multi-Label Learning (MLL) assigns multiple labels to an object, which is a fundamental and intriguing task in various real-world applications such as image retrieval, autonomous vehicles, and sentiment analysis. There is plenty of literature on MLL, such as Binary Relevance (BR) [102] and MLKNN [98], which decompose the multi-label problem into multiple independent binary classification problems. It is one of the simplest and most fundamental methods in MLL. Some methods focus on exploring the correlations between labels to improve classification performance. This includes approaches that consider pairwise label correlations and more complex inter-label relationships [58, 98]. For example, MDFS [93] proposes an embedded feature selection method via manifold regularization to select discriminative features for multi-label learning. Recent research [56] proposes a shared weight matrix with low-rank and sparse regularization for multi-label learning. It utilizes both the feature manifold and label manifold to guide the shared weight learning process.

Traditional multi-label methods often assume that all relevant labels for each training sample are precisely annotated. However, this assumption does not always hold in real-world applications, where each example is associated with multiple candidate labels, among which only one is valid. The scenario has been formalized as a learning framework called Partial Label Learning (PLL) by [12]. The approaches of PLL can be broadly categorized into several types, such as identifying the true label from the set of

candidate labels with disambiguation strategies [101], adapting traditional supervised learning algorithms to handle partial labels [85], or treating all candidate labels equally and make predictions based on the average effects of these labels [44].

Partial Multi-Label learning (PML) is a more complicated task than PLL since the desired predictor is a multi-label one in PML instead of a single-label one in PLL and the number of ground-truth labels is further unknown. PML is particularly relevant in tasks like crowd-sourced image tagging [87]. A prevalent approach of PML examples is disambiguation. It tries to recover ground-truth labeling information from candidate labels, by either introducing labeling confidences [100] or utilizing a combination of low-rank and sparse decomposition [65]. For instance, PARTICLE [97] utilizes an iterative process of label propagation to identify reliable labels with high levels of labeling confidence. Other studies adopt a heuristic approach, based on the assumption that noisy labels within the candidate set are typically sparse. This leads to the development of methods where either a noisy label matrix [66] is derived, or a noisy label identifier [88] is established, both employing sparsity regularization techniques for effective learning. For example, PMLNI [88] jointly learns a noisy label identifier, which identifies feature-induced noisy labels, as well as a multi-label classifier for prediction. Although current methods in partial multi-label classification have made significant advances, they often overlook the cause of noisy labels in the candidate set, assuming these to be randomly generated. This assumption doesn't align with many real-world scenarios. For example, if an image is tagged with labels such as 'zoo', 'lion', and 'tiger', it similarly increases the likelihood of other animal-related labels, like 'elephant' or 'giraffe', being relevant as well. Therefore, the correlation between labels is crucial. This has led to a heightened interest in graph-based partial label methods, which seek to exploit these label correlations for better disambiguation of candidate labels, as evidenced in various studies [36, 45, 74].

However, existing partial multi-label methods ignore a critical problem: the presence of excessive redundant and noisy features residing in the original data can greatly impact the performance. Some multi-label feature selection techniques are proposed to significantly reduce the complexity of multi-label data by selecting important features. For example, MDMR [37] introduces a method that assesses each feature by balancing its redundancy with other features against its relevance to the labels. LRFS [103]

distinguishes labels into two categories, factoring in label redundancy for feature evaluation. Embedded methods, which combine feature selection with the learning process, include MDFS [93], which delves into both local and global label correlations within a manifold structure. Another approach, MLMLFS [109], is tailored for multi-label data with incomplete labels, integrating feature selection within the label recovery process. SCMFS [24] utilizes coupled matrix factorization to uncover the interplay between feature and label matrices. However, these methods fail to select the most effective features in a partial multi-label context since they are based on the assumption that all collected labels are accurate. Current partial multi-label feature selection methods, such as PMLFS [76], directly uses sparse regularization as the feature selection technique which may have limited performance since it lacks disambiguation strategies for identifying true labels. PML-FSSO [23] proposed a subspace optimization method that utilizes the theory of linear weighted to jointly consider the label subspace and feature subspace.

3

Research Methodology

In this chapter, the methodology for solving imperfection data and classification problems is systematically presented and the related methods are introduced. Additionally, the imperfect datasets in different learning situations, which are collected from the real world, are presented. The datasets are used for experimental evaluations of the proposed algorithms and the compared methods in this thesis.

3.1 Overview of Proposed Reliable and Robust Methods for Imperfection Data Classification

The goal of this study is to develop a series of high-reliability machine learning methods for the applications of classification on various complexity datasets. To achieve this goal, algorithms that correspond to each specific task are developed. For semi-supervised learning, we proposed a novel method named robust embedding regression (RER) for semi-supervised learning [3], a transfer learning method named redirected transfer learning (RTL) for robust multi-layer subspace learning [2], and a partial multi-label learning method named partial multi-label learning with adaptive dual graph disambiguation (ADGD). The overall framework of these methods is shown in Figure 3.1 and the content of the framework can be described in four phases.

Phase 1. Imperfection Dataset Construction: To simulate imperfect data in real-world scenarios and demonstrate the effectiveness and reliability of our experimental methods, we selected real-world data for model training involving different learning

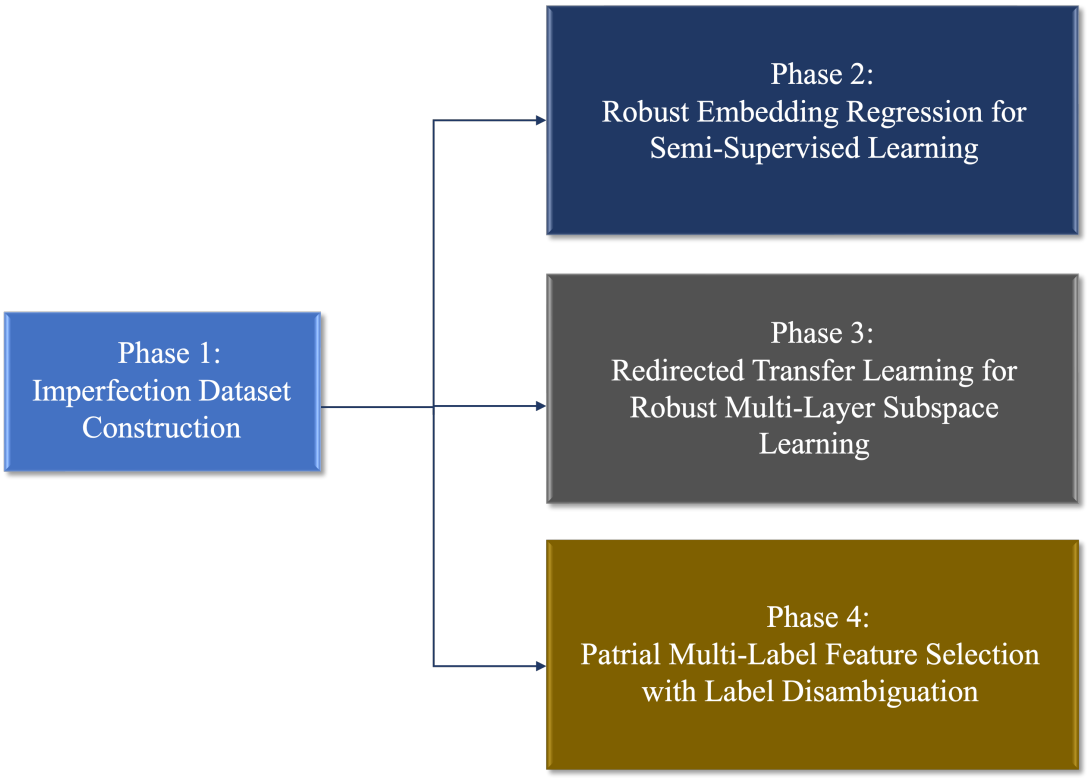


Figure 3.1: Architecture of the proposed reliable machine learning framework of four phases: (1) imperfection dataset construction, (2) semi-supervised learning with robust embedding regression method, (3) transfer learning with redirected transfer learning method, and (4) Partial multi-label learning with feature selection method.

paradigms such as semi-supervised learning, transfer learning, and multi-label learning. Key steps in the dataset construction included:

- For semi-supervised learning, we chose a subset of images that naturally encompass both labeled and unlabeled data. This selection mirrors typical real-world scenarios where obtaining fully labeled datasets is often impractical.
- To train the transfer learning models, we gathered images from a diverse array of domains. This variety is critical in assessing the models' ability to effectively transfer learned knowledge from one domain (source) to another (target), which is a hallmark of successful transfer learning.
- For partial multi-label learning, our dataset included images with complex label sets, reflecting the multifaceted nature of real-world objects and scenes. These

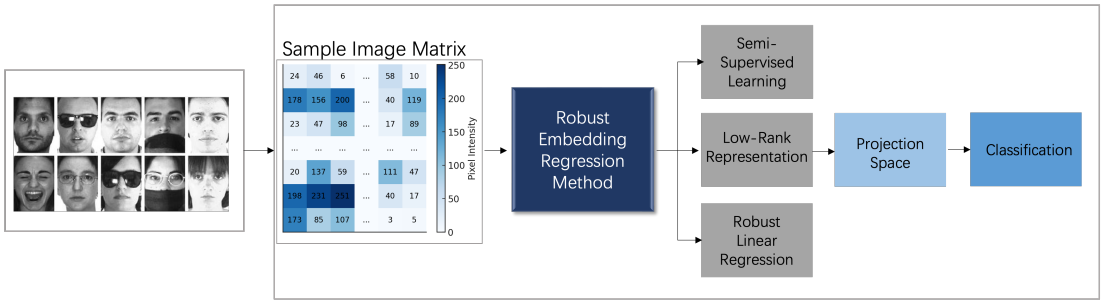


Figure 3.2: Image data classification with robust embedding regression method.

images, associated with multiple labels, provide a realistic challenge for multi-label classification models.

In addition to these specific selections, a key aspect of our dataset construction was the introduction of noise into the original, real-world images. This step was crucial in testing the robustness of our models. By deliberately adding random noise and other forms of data imperfections, we aimed to simulate the often noisy and imperfect nature of real-world data. This approach not only challenges the models under more strenuous conditions but also provides a more authentic evaluation of their performance, particularly in terms of their resilience and adaptability to real-world data imperfections.

Phase 2. As shown in Figure 3.2, this phase aims to solve the data classification problem at the image-level for analyzing and reducing the influence of noisy and redundant data in the learning process. The images are read and converted into data matrices for the robust embedding regression method. The data are usually divided into three sets: training, validation, and testing. The training set is used to tune parameters in various algorithms and the validation process is conducted to show the fitting of the models (this process can be omitted in certain cases). Finally, experiments on the testing set are conducted to evaluate the effectiveness of the trained method. The robust embedding regression method synthesizes the strengths of existing semi-supervised learning frameworks, robust linear regression, and low-rank representation techniques. The integration of these methods is aimed at enhancing the model's robustness and efficacy. The method finally learns projection spaces for image classification.

Phase 3. As shown in Figure 3.3, this phase improves the effectiveness and efficiency of the transfer learning method, which leverages labeled source domain data to infer classifications on an unlabeled target domain with a different distribution. Every

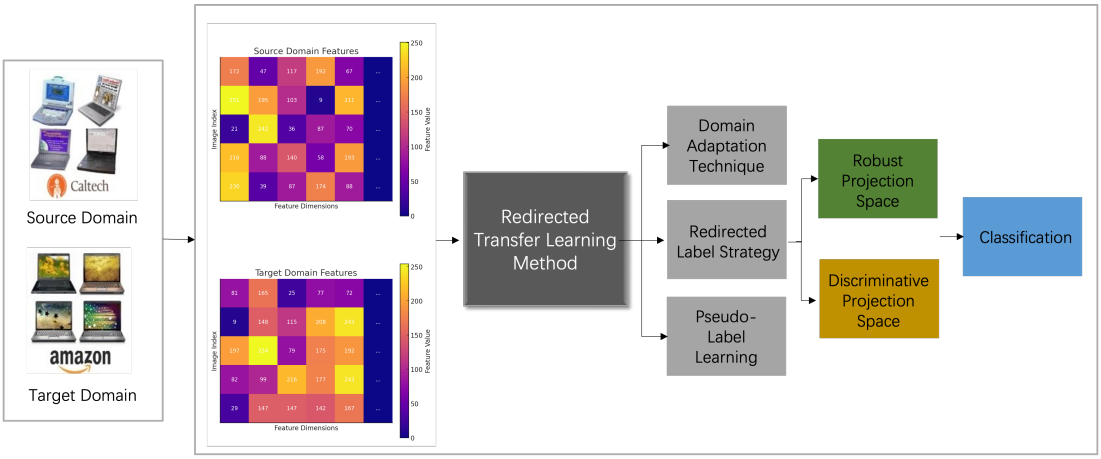


Figure 3.3: Data classification with the redirected transfer learning method.

data from different domains is first represented as a data matrix and then analyzed one by one. This phase trains the model with the source data and tests on target data. The redirected transfer learning method applies the domain adaptation technique, redirected label strategy, and pseudo-label learning. The method finally learns two different spaces for knowledge transformation and data classification separately.

Phase 4. As shown in Figure 3.4, we proposed a novel Partial Multi-Label Learning (PML) method that focuses on the problem of label ambiguity and redundant features. Both images with noisy labels and the corresponding label information are read and converted into data matrices. Our novel approach integrates dual adaptive graphs and a sparse projection matrix. The dual graphs dynamically capture complex relationships in the data, one focusing on features and the other on labels, allowing for a more precise and adaptable representation of the data, leading to enhanced label disambiguation. Concurrently, the sparse projection matrix, regulated by the $L_{2,1}$ norm, optimizes feature-to-label mapping and ensures the model focuses on the most relevant features. This method aims to not only learn an optimal projection for classification but also a reliable label confidence matrix for label disambiguation propagation.

3.2 Research Methodology of Semi-Supervised Learning

In this section, we review the related techniques to solve the semi-supervised learning problem in this thesis, such as robust linear regression, manifold-based semi-supervised

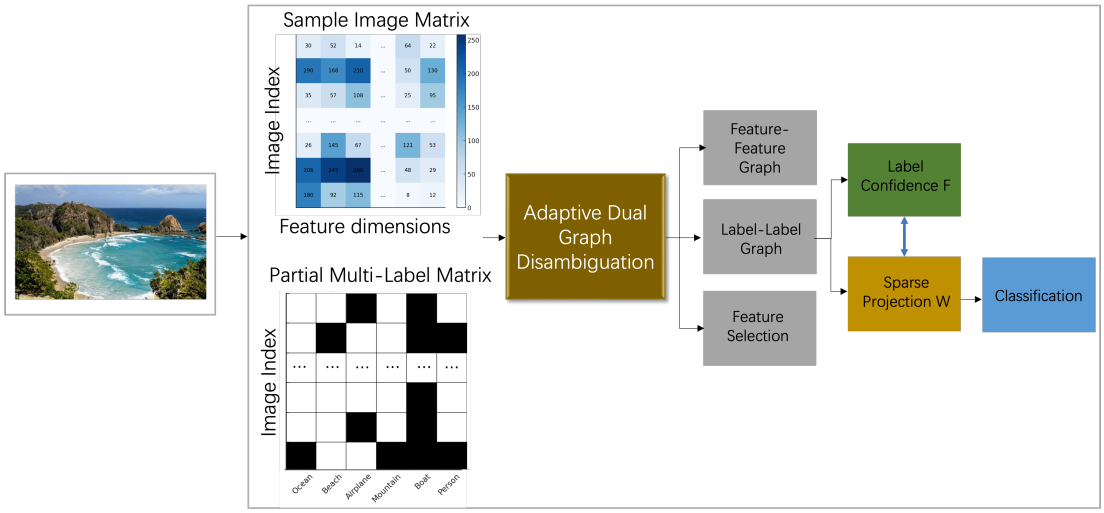


Figure 3.4: Partial multi-label learning with adaptive dual graph disambiguation.

learning, and low-rank representation.

3.2.1 Definitions of Semi-Supervised Learning

For setting semi-supervised learning, matrix $X_l = [\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_l}] \in \mathbb{R}^{d\times l}$ denotes the l labeled data samples, and matrix $X_u = [\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_u}] \in \mathbb{R}^{d\times u}$ denotes the u unlabeled data samples, where d is the dimensionality of the features. Matrix $Y_l = [\boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_l}] \in \mathbb{R}^{c\times l}$ represents the label matrix of labeled data, and matrix $Y_u \in \mathbb{R}^{c\times u}$ represents the missing label matrix of unlabeled data, where c is the number of classes. When $\boldsymbol{x_i}$ belongs to the jth $(1 \leqslant j \leqslant c)$ class, $y_{ij} = 1$, otherwise, $y_{ij} = 0$. We define $X = [X_l, X_u] \in \mathbb{R}^{d\times n}$ and $Y = [Y_l, Y_u] \in \mathbb{R}^{c\times n}$, where $n = l + u$. For a matrix $W = (w_{ij})$, $\boldsymbol{w^i}$ denotes the ith row and $\boldsymbol{w_j}$ is the jth column of matrix W. Moreover, $L_{a,b}$-norm of matrix $W \in \mathbb{R}^{d\times c}$ is computed as follows:

$$\begin{aligned}\|W\|_{a,b} &= \left(\sum_{i=1}^{d}\left(\sum_{j=1}^{c}|w_{i,j}|^a\right)^{b/a}\right)^{1/b} \\ &= \left(\sum_{i=1}^{d}\left\|\boldsymbol{w^i}\right\|_a^b\right)^{1/b}, a > 0, b > 0.\end{aligned}$$

Minimizing $\|W\|_{2,1} = \sum_{i=1}^{d}\left\|\boldsymbol{w^i}\right\|_2$ implies to make W sparse in rows, thus feature selection in a linear model $W^T X$.

3.2.2 Robust Linear Regression

We first review some robust linear regression methods. Ridge regression is almost the first robust linear regression method. It minimizes the penalized least square:

$$\min_{W} \left\| W^T X - Y \right\|_F^2 + \lambda \left\| W \right\|_F^2, \tag{3.1}$$

where $\|\cdot\|_F^2$ $(a = 2, b = 2)$ is Frobenius norm, and $\lambda > 0$ is the regularization coefficient. It has the solution

$$W = \left(X X^T + \lambda I \right)^{-1} X Y^T. \tag{3.2}$$

Essentially, ridge regression is applicable only when data are labeled ($X = X_l$, $Y = Y_l$). By regularization, the regression has a higher generalization ability than the linear regression without regularization.

Motivated by ridge regression, robust feature selection (RFS) [48] is proposed to select the effective features and enhance robustness by imposing the $L_{2,1}$-norm in both regression and regularization terms as follows:

$$\min_{W} \left\| W^T X - Y \right\|_{2,1} + \lambda \left\| W \right\|_{2,1}, \tag{3.3}$$

where $\|\cdot\|_{2,1}$ $(a = 2, b = 1)$ is $L_{2,1}$-norm. The first term brings robustness in regression and the second term works as feature selection.

3.2.3 Manifold-Based Semi-Supervised Learning

According to the rapid increase of the data size, semi-supervised learning (SSL) has obtained more attention for saving the cost of labeling which requires a great amount of labor in real applications. SSL utilizes unlabeled samples in addition to labeled samples for designing classifiers. In general, we can expect that if $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are close to each other in the feature space, they would belong to the same class with a higher probability. According to the label propagation theory [112], the latent relationship between labeled samples and unlabeled samples is characterized by a weight undirected graph S. In S, $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are connected by the similarity s_{ij} defined by the following:

$$s_{ij} = \begin{cases} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{2\sigma^2} \right) & \text{if } \boldsymbol{x}_j \in N_k\left(\boldsymbol{x}_i\right) \vee \text{or } \mathbf{x}_i \in N_k\left(\mathbf{x}_j\right), \\ 0 & \text{otherwise,} \end{cases} \tag{3.4}$$

where $N_k\left(\boldsymbol{x_i}\right)$ means the set of k-nearest neighbors of $\boldsymbol{x_i}$ and σ is a heat kernel parameter. We denote $F = [F_l, F_u] \in \mathbb{R}^{c \times n}$ is soft prediction label matrix and $F_l = Y_l \in \mathbb{R}^{c \times l}$

denotes the value on the labeled data points. In general, the label propagation method solves the problem:

$$\min_{F,F_l=Y_l} \sum_{i,j=1}^{n} \|\boldsymbol{f_i} - \boldsymbol{f_j}\|^2 s_{ij}. \tag{3.5}$$

The problem (3.5) can be rewritten as

$$\min_{F:F_l=Y_l} \mathrm{tr}\left(FL_SF^T\right), \tag{3.6}$$

where L_S is the graph Laplacian induced from S and calculated as $L_S = D - S$, $D = \mathrm{diag}(d_{11}, ..., d_{nn})$ for $d_{ii} = \sum_i s_{ij}$. However, this method fails to construct a classifier explicitly (out-of-sample problem), because it is an embedding, not a transformation. To cope with this problem, FME [49] was proposed to perform semi-supervised learning and linear regression simultaneously as follows:

$$\min_{W,F:F_l=Y_l} \left\|W^TX - F\right\|_F^2 + \lambda \|W\|_F^2 + \alpha \mathrm{tr}\left(FL_SF^T\right), \tag{3.7}$$

Here, the explicit classifier is a linear mapping with W. However, FME is sensitive to noises and outliers.

3.2.4 Low-Rank Representation

Low-rank representation (LRR) [71] is another critical development route in our approach. LRR has strong matrix recovery ability and robustness to outliers which assumes the raw data X is not perfect but corrupted by the noises E, i.e., $X = XZ + E$, where $Z = [\boldsymbol{z_1}, \boldsymbol{z_2}, ..., \boldsymbol{z_n}] \in \mathbb{R}^{n\times n}$ is a coefficient matrix, in which $\boldsymbol{z_i}$ is the representation of data point $\boldsymbol{x_i}$ by the other points. Thus, in order to recover the clean data and handle the noisy part, LRR solves the following minimization problem:

$$\begin{aligned} &\min_{Z,E} \|Z\|_* + \beta \|E\|_{2,1}, \\ &\text{subject to } X = XZ + E. \end{aligned} \tag{3.8}$$

where β is a bias parameter, $\|\cdot\|_*$ is the nuclear norm, i.e., the sum of singular values. The matrix $Z \in \mathbb{R}^{n\times n}$ is expected to select a small number of samples (columns) of X to recover X with the low-rank requirement when $d < n$. In other words, a low-rank Z performs sample selection.

After obtaining the coefficient matrix Z, some clustering methods construct an affinity graph by computing $S = 0.5(|Z| + |Z|^T)$ or directly utilize the representation

graph Z to obtain the final clustering result [90]. However, these methods cannot find the optimal solution since the graph construction and subsequent optimization are not in a unified framework. More importantly, the graph is generally learned from raw data, which usually contains noises or redundant features in reality. Then, the obtained graph may be inexact or sub-optimal. LRR_AGR is proposed to construct an adaptive graph and extend LRR to the following graph learning model [80]:

$$\begin{aligned} &\min_{Z,E} \sum_{i,j}^{n} \|\boldsymbol{x_i} - \boldsymbol{x_j}\|_2^2 z_{ij} + \lambda_1 \|Z\|_* + \lambda_2 \|E\|_{2,1}, \\ &\text{subject to } X = XZ + E, Z \geq 0, \text{diag}(Z) = \mathbf{0}, Z^T\mathbf{1} = 1. \end{aligned} \tag{3.9}$$

The non-negative and diagonal element constraints ($Z \geq 0$ and $\text{diag}(Z) = \mathbf{0}$) are to ensure that the learned matrix can be directly used as the affinity matrix and avoid self-representation. The condition $Z^T\mathbf{1} = 1$ is to avoid the case that those elements of any row of graph Z are all zeros. By introducing a graph regularization term $\sum_{i,j}^{n} \|\boldsymbol{x_i} - \boldsymbol{x_j}\|_2^2 z_{ij}$, the similarity relationships between sample points $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ can be truly reflected by the adaptive matrix Z based on reconstruction error minimization. In this proposed RER algorithm, we borrow the idea of low-rank representation for sample selection and for extraction of neighborhood relationships among clean feature vectors.

3.3 Research Methodology of Transfer Learning

In this section, we review the related techniques to solve the transfer learning problem in this thesis, such as the domain adaptation technique, and redirected label strategy.

3.3.1 Definitions of Transfer Learning

For setting transfer learning [54], assuming there is only one source domain and one target domain: Given two different domains and corresponding learning tasks, i.e., a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, the aim of transfer learning is to help improve the learning of target task in D_T by using the knowledge in D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$. In order to address the problem of different distributions between the source data and target data, recent research paid attention to representation-based methods with low-rank constraints, which model the distribution difference between D_S and D_T and minimize it.

3.3.2 Domain Adaptation with Low-Rank Constraint

For the unsupervised transfer learning method for classification tasks, we assume two different domains and two different but related probability distributions over them. Each distribution is defined jointly over $\mathcal{X}$ and $\mathcal{Y}$. For more detail, the source domain D_S and a distribution $p_s(x, y)$ is assumed. The target domain D_T and has a distribution $p_t(x, y)$. We denote the samples separately in the feature space and in the label space, as $X = [X_S, X_T] \in \mathbb{R}^{d \times n}$ and $Y = [Y_S, Y_T] \in \mathbb{R}^{c \times n}$, where c is the number of classes, d is the number of dimensionality, $n = n_S + n_T$, and Y_T is unknown. The goal of unsupervised transfer learning is to predict the target labels $Y_T \in \mathbb{R}^{c \times n_T}$ as precisely as possible, given (X_S, Y_S) from a source domain and X_T from the target domain.

Transfer learning typically aims to seek a common subspace where the source data X_S and target data X_T are projected into and have similar distributions. In the common subspace, we assume each test data from the target domain can be approximately reconstructed by the data from source domains. This requirement can be formulated as

$$\min_{W,Z} \left\| W^t X_T - W^t X_S Z \right\|_F^2, \tag{3.10}$$

where $W \in \mathbb{R}^{d \times c}$ is the transformation matrix and $Z \in \mathbb{R}^{n_S \times n_T}$ is reconstruction matrix. The superscript t denotes the transpose. In this formulation, we seek W such that the target data are linearly approximated by source data ($W^t X_T \approx W^t X_S Z$), that is, the domain distribution gap is minimized.

For better capturing and exploring the latent structure of cross-domain data, a low-rank constraint is imposed on the reconstructive matrix Z. Specifically, we solve a rank minimization problem [59], with $\alpha > 0$:

$$\begin{aligned} \min_{W,Z,E} \frac{1}{2}\Phi(W, X_S, Y_S) + \alpha \cdot \operatorname{rank}(Z) + \|E\|_l, \\ \text{subject to } W^t X_T = W^t X_S Z + E, \end{aligned} \tag{3.11}$$

where $\operatorname{rank}(\cdot)$ denotes the rank and $\|\cdot\|_l$ denotes a certain norm, such as L_1 or $L_{2,1}$-norm. The parameter α controls the intrinsic correlation of the reconstruction matrix. Note that E represents the reconstructive error, and $\Phi(W, X_S, Y_S)$ is a certain regression error of X_S to Y_S in the source domain. In practice, due to its NP-hardness, instead

of the rank of Z, the nuclear norm is adopted for Z penalizing [25]:

$$\begin{aligned} \min_{W,Z,E} \frac{1}{2}\Phi\left(W, X_S, Y_S\right) + \alpha\|Z\|_* + \|E\|_l, \\ \text{subject to } W^t X_T = W^t X_S Z + E. \end{aligned} \tag{3.12}$$

However, this method cannot directly apply $\Phi\left(W, X_T, Y_T\right)$ to unknown data from the target domain because of the unavailability of Y_T.

3.3.3 Linear Regression and Relaxed-Label-Based Regressions

The existing transfer learning methods design the regression error $\Phi\left(W, X_S, Y_S\right)$ in (3.12) as a linear regression model as follows [106]:

$$\Phi\left(W, X_S, Y_S\right) = \left\|Y_S - W^t X_S\right\|_F^2 + \lambda \left\|W\right\|_F^2. \tag{3.13}$$

The parameter λ weights the importance of the regularization term. Once W is obtained by solving (3.13), classification of a test sample $\boldsymbol{x} \in \mathbb{R}^d$ in the target domain is made by assigning class k such that $k = \arg\max_i \left(W^t \boldsymbol{x}\right)_i$, the largest row.

Usually, Y consists of one-hot vectors taking one only in the class index. However, in reality, since the distributions are different between source and target domains, strict zero-one indicators are not appropriate and sometimes even harmful to classification. Therefore, some methods introduce a label relaxation strategy [22, 89, 96] that the binary labels are dragged to directions such that the distances of inter-class are enlarged as much as possible or a better alignment is achieved between two distributions. As an example, let x_1, x_2, x_3 be three training samples taken from the second, third, and first class, respectively, that is, $Y = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$. Then, a relaxed matrix $\hat{Y}$ can be $\hat{Y} = Y + B \circ M = \begin{bmatrix} -m_{11} & -m_{12} & 1+m_{13} \\ 1+m_{21} & -m_{22} & -m_{23} \\ -m_{31} & 1+m_{32} & -m_{33} \end{bmatrix}, m_{ij} \geq 0$, where B is defined as $(B)_{ij}$ that is 1 if $y_{ij} = 1$, otherwise -1, and $\circ$ is the Hadamard product operator. In this example, the labels (values) are enhanced so as to increase class separability. Then, (3.3) becomes

$$\min_{W,M} \left\|Y_S + B \circ M - W^t X_S\right\|_F^2 + \lambda \left\|W\right\|_F^2, M \geq 0. \tag{3.14}$$

This idea might work for enhancing class separability, but it works equally for every sample. This equality is not always sufficient for aligning source and target

distributions. Therefore, we adopt a "margin" restriction instead, such as [104], as follows:

$$\begin{aligned} &\min_{W,T} \left\|T - W^t X_S\right\|_F^2 + \lambda \left\|W\right\|_F^2, \\ &\text{subject to } t_{i,l_i} - \max_{j \neq l_i} t_{i,j} \geqslant 1, \end{aligned} \tag{3.15}$$

where l_i is the true class index of the ith sample. It requires that the margin is larger or equal to one, in other words, the target value t_{i,l_i} should be larger at least one to any other value $t_{i,j}$ $(j \neq l_i)$. This way is more flexible than (3.14) in re-labeling.

3.4 Research Methodology of Partial Multi-Label Learning

In this section, we review the related works to solve the partial multi-label learning problem in this thesis, including the graph-based label disambiguation propagation technique and adaptive graph embedding.

3.4.1 Definitions of Partial Multi-Label Learning

The scenario is referred as to partial multi-label (PML) learning which is formalized by [87]. Formally, denote $X = [\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n}] \in \mathbb{R}^{d \times n}$ as the instance-feature matrix for n instances, $Y = [\boldsymbol{y_1}; \boldsymbol{y_2}; ...; \boldsymbol{y_n}] \in \{0, 1\}^{n \times q}$ as the candidate label matrix where $\boldsymbol{y_i}$ corresponds to i-th instance's label vector, $y_{ij} = 1$ means the j-th label is included in the candidate label set of instance x_i, $y_{ij} = 0$, otherwise. PML aims to learn a multi-label model from the feature matrix together with the candidate label matrix and assign the predictive labels for unseen instances.

3.4.2 Disambiguation with Fixed Graph

Graph-based disambiguation strategy [17] is to make use of the local manifold structure in feature space to help disambiguate the candidate label set. The current partial multi-label learning methods[11, 74, 97] usually employ such a strategy in a two-stage manner which first generates latent labeling confidence over the candidate label set by utilizing the graph structure of feature space and then learns a multi-class model by fitting a multi-output regressor with the generated labeling confidence.

In the first stage, graph-based PML methods aim to generate a normalized real-valued labeling confidence matrix $F \in \mathbb{R}^{n \times q}$ by adapting the label propagation procedure based on a weighted graph over training instances. Given a weighted directed graph $G = (V, E, S)$, $V = \{\boldsymbol{x_i} | 1 \leq i \leq n\}$ corresponds to the set of training instances, and the corresponding set of directed edges is $E = \{(\boldsymbol{x_i}, \boldsymbol{x_j}) | i \in N_k(\boldsymbol{x_i}), 1 \leq j \leq n\}$. Specifically, weight matrix $S \in \mathbb{R}^{n \times n}$ is optimized by solving the following minimum error reconstruction problem

$$\begin{aligned} &\min_{S} \sum_{j=1}^{n} \left\| x_j - \sum_{i=1}^{n} s_{ij} \cdot x_i \right\|_2^2 \\ &\text{subject to } S^T \mathbf{1}_n = \mathbf{1}_n, s_{ij} \geq 0 \left(i \in N_k(\boldsymbol{x_j})\right), s_{ij} = 0 \left(i \notin N_k(\boldsymbol{x_j})\right). \end{aligned} \tag{3.16}$$

After obtaining the graph weight matrix, the labeling confidence matrix F can be acquired by solving the following problem:

$$\begin{aligned} &\min_{F} \sum_{j=1}^{n} \left\| f_j - \sum_{i=1}^{n} s_{ij} \cdot f_i \right\|_2^2 \\ &\text{subject to } F \mathbf{1}_q = \mathbf{1}_n, f_{il} \geq 0 \left(\forall y_{il} = 1\right), f_{il} = 0 \left(\forall y_{il} = 0\right), \end{aligned} \tag{3.17}$$

where $\mathbf{1_n}$ is an all $\mathbf{1}$ vector with size n. For each training example $(\boldsymbol{x_i}, \boldsymbol{Y_i})$, f_{il} represents its corresponding labeling confidence of the lth label being the ground-truth label for $\boldsymbol{x_i}$. The labeling confidence vector $\boldsymbol{f_i} \in \mathbb{R}^q$ satisfies the following constraints: (I) $\sum_{y_{il}=1} f_{il} = 1$ (normalization), (ii) $f_{il} \geq 0 \left(\forall y_{il} = 1\right)$ (non-negativity), and (iii) $f_{il} = 0 \left(\forall y_{il} = 0\right)$. The second constraint implies that the ground-truth label of each example resides in the candidate label set, and the third constraint guarantees that the labeling confidence of each non-candidate label must be 0. Note that F is a learnable matrix and we can learn this matrix by minimizing the following (3.17) once the similarity graph weights S is determined.

In the second stage, PML methods make use of the confidence labels elicited in the first stage to induce the multi-label predictive model. Two potential drawbacks lie in that real-world data are usually contaminated by significant noises and outliers which make the fixed graph structure recovered from the original feature space less reliable [77]. Another is the two-stage learning process cannot obtain the optimal label confidence since it cannot take full advantage of the correlation between feature instance and label.

3.4.3 Adaptive Graph Embedding

The quality of S in (3.16) affects learning performance significantly. If S is obtained directly in feature space, it is challenging for S to reveal the intrinsic structure within the data since the noise and outliers are high. To this end, Some research [72] proposes an adaptive graph instead of a fixed graph to obtain the similarity matrix and simultaneously update the label confidence to achieve the best results as follows:

$$\begin{aligned} &\min_{F,S} \sum_{j=1}^{n} \left\| x_j - \sum_{i=1}^{n} s_{ij} \cdot x_i \right\|_2^2 + \alpha \sum_{j=1}^{n} \left\| f_j - \sum_{i=1}^{n} s_{ij} \cdot f_i \right\|_2^2 \\ &\text{subject to } S^T \mathbf{1}_n = \mathbf{1}_n, 0_{n\times n} \leq S \leq N, \\ &\quad F\mathbf{1}_q = \mathbf{1}_n, 0_{n\times q} \leq F \leq Y, \end{aligned} \tag{3.18}$$

where $N \in \{0,1\}^{n\times n}$ is defined as: $n_{ij} = 1$ if $i \in N_k(\boldsymbol{x_j})$ and $n_{ij} = 0$ otherwise. Furthermore, $0_{n\times n}$ is the $n \times n$ all 0 matrix, and α is the trade-off parameter between label space and feature space respectively. However, due to high-dimensional feature space, complex label correlations and noises in multi-label data, (3.18) has limited performance since it lacks the feature selection ability. Furthermore, S is constructed based on data and thus cannot explore both the feature correlation and label correlation as well as the correlation between the feature space and the label space.

4

Robust Regression Embedding for Semi-Supervised Learning

4.1 Preliminaries

To utilize both labeled data and unlabeled data in real-world applications, semi-supervised learning is widely used as an effective technique. However, most semi-supervised methods do not perform well when there are many noises and redundant information in the original data. Therefore, In this chapter, we introduce a novel semi-supervised method named *Robust Embedding Regression* (RER) [3], which focuses on coping with the sensitivity to noise and outliers in feature space and to exploit label information more effectively in an affinity graph for low-dimensional embedding. These issues have not been well addressed in the framework of semi-supervised learning so far.

For this goal, we learn several ideas from three research streams of *semi-supervised learning, robust linear regression*, and *low-rank representation*. Specifically, we discuss the best choice of norms in the proposed objective function so as to have robustness and employ the low-rank techniques to realize effective label propagation. Thus, RER has been strengthened functionally by inheriting the functions equipped in *semi-supervised learning*, *robust linear regression*, and *low-rank representation* (Figure 4.1). Especially, the robustness comes from *robust linear regression* and the efficiency of label propagation comes from *low-rank representation*.

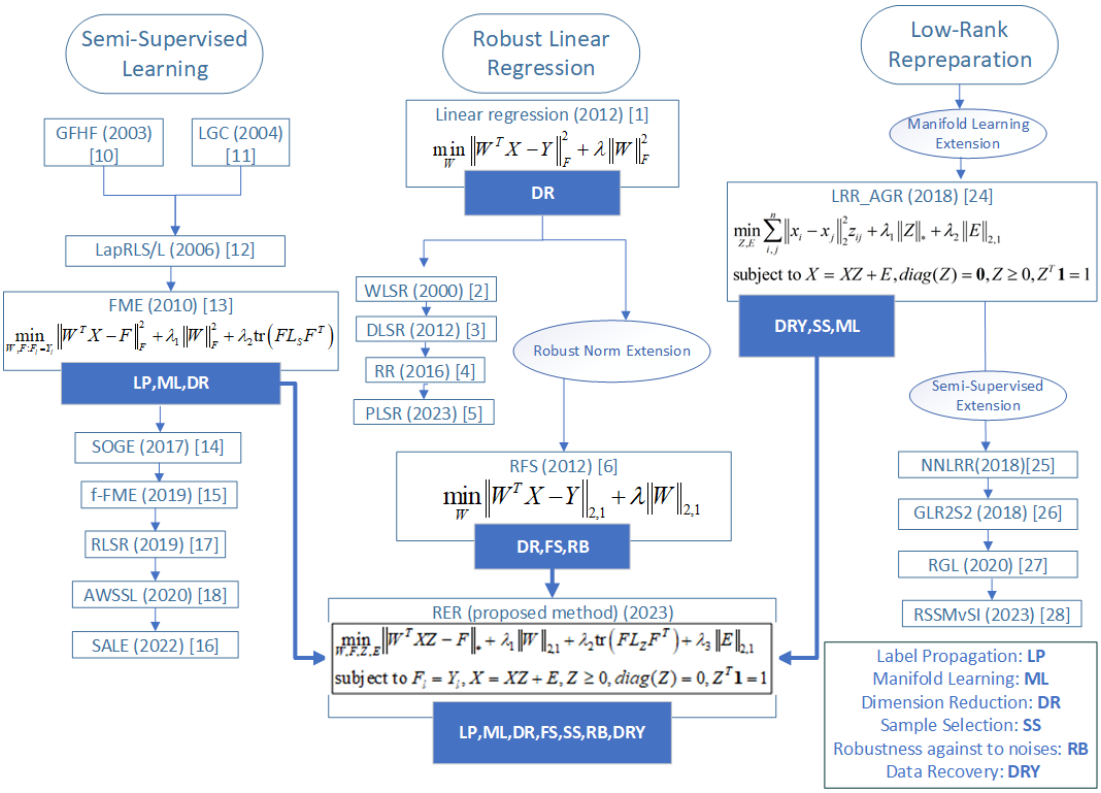


Figure 4.1: The development routes of related work. Our method inherits the functions and advantages of the existing methods, including label propagation (LP), manifold learning (ML), dimension reduction (DR), feature selection (FS), robustness against noises (RB), data recovery (DRY) and sample selection (SS), for robust semi-supervised classification and clustering. The blue boxes present the properties of each method.

4.2 Robust embedding regression for semi-supervised learning

For SSL methods, how to effectively utilize potential relationships between labeled data and unlabeled data to predict label information is crucial to a promising performance. However, the existing methods are sensitive to the noises and outliers in the original data which may greatly degrade the final performance. Thus, in this section, we will present a robust semi-supervised regression method that inherits the advantages of the existing techniques and elaborately integrates them into a unified objective function for semi-supervised classification.

(a) A_1 **(b)** A_2 **(c)** B_1

Figure 4.2: Three images from the YaleB dataset.

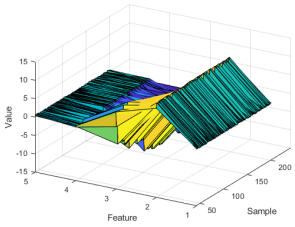
4.2.1 Formulation of RER

We design a novel semi-supervised regression method that aims to improve the robustness of the algorithm by elaborately integrating robust linear regression, low-rank representation, and semi-supervised learning into a unified optimization objective function. Specifically, the relationship between the labeled and unlabeled samples is captured by an adaptive graph rather than the pre-defined graph, more specifically, Laplacian L_Z of noiseless data instead of Laplacian L_S of raw data with noise, in which LRR technique is used to remove the noises and outliers in L_Z. More importantly, the proper norm selection for both reconstruction and regularization terms further brings the robustness of the algorithm and obtains promising classification performance. Therefore, we have the following objective function of RER:

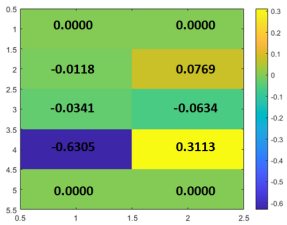
$$\begin{aligned} &\min_{W,F,Z,E} \left\|W^T XZ - F\right\|_* + \lambda_1 \left\|W\right\|_{2,1} + \lambda_2 \mathrm{tr}\left(FL_Z F^T\right) + \lambda_3 \|E\|_{2,1} \\ &\text{subject to } F_l = Y_l, X = XZ + E, \mathrm{diag}(Z) = \mathbf{0}, Z \geq 0, Z^T\mathbf{1} = 1, \end{aligned} \tag{4.1}$$

where λ_1, λ_2 and λ_3 are non-negative trade-off parameters. We directly use Z as an affinity matrix whose (i, j)-edge is weighed by z_{ij}. L_Z is the Laplacian matrix which is computed as $D - Z$, where $(D)_{ii} = \sum_{i \neq j} z_{ij}$.

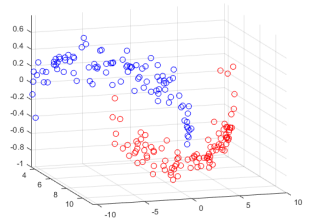
In (5.7), we use a more robust norm, the nuclear norm, rather than the Frobenius norm as a basic metric to encode the reconstruction term (i.e., $\left\|W^T XZ - F\right\|_*$), which has been proven its effectiveness in recent research [35, 43]. The second term imposes a row-sparsity norm constraint on projection W to accomplish feature selection. The third term with constraint $F_l = Y_l$ functions as label propagation in SSL. It constructs an adaptive graph and induces the label information for the unlabeled data points in a unified objective function. The constraint $X = XZ + E$ is utilized to ensure the graph construction based on the clean data and reduce the negative influence of noises and outliers. The last term ensures the sparsity of the corruption matrix and further improves the robustness by introducing $L_{2,1}$-norm.



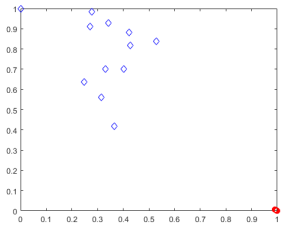
(a) The visualization of 5 features of torus data (Features 1 and 5 are noise features)



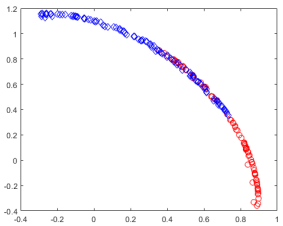
(b) The projection matrix W obtained by RER



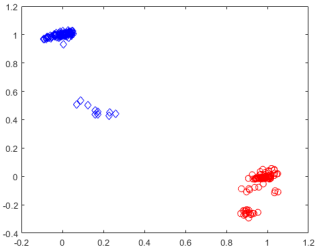
(c) Visualization results of original data without noises



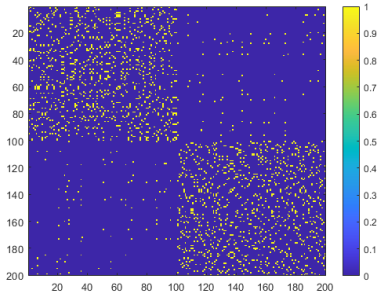
(d) Visualization results of matrix F obtained by RER



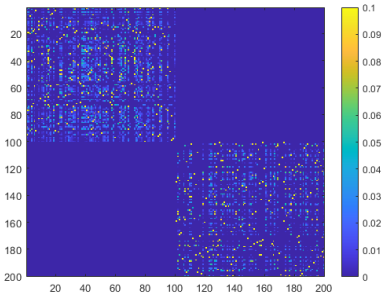
(e) Visualization results of matrix $W^T X$ obtained by RER



(f) Visualization results of matrix $W^T XZ$ obtained by RER



(g) The similarity matrix S of the pre-constructed graph obtained by KNN



(h) The similarity matrix Z of the adaptive graph obtained by RER

Figure 4.3: Illustration of various visualization results and matrices.

Table 4.1: Comparison of the distance between the images using different norms. $Dist(A, B) = \|A - B\|_{a,b}$ according to chosen norm.

	Dist(A_1,A_2)		Dist(A_1,B_1)
$\|\cdot\|_*$	2278	$<$	4232
$\|\cdot\|_1$	1414	$>$	1369
$\|\cdot\|_F^2$	1329	$>$	1282

4.2.2 Support Examples

On the toy problem, we demonstrate the validity of chosen norms and functional advantages.

The justification for the nuclear norm in the regression residual can be demonstrated by a simple example. We reproduce the experiment in [43]. We show 3 images of 2 people in Figure 4.2, in which (a) A_1 and (b) A_2 are from one person A and (c) B_1 is from person B. The distances between the images of the same person and a different person are computed by the nuclear norm, L_1-norm, and Frobenius norm, respectively, as shown in Table 4.1. From the results, we can observe that both L_1-norm and Frobenius norm cannot properly classify the image of person A since the distance between A_1 and A_2 is larger than A_1 and B_1. The nuclear norm correctly classifies the images since the distance between A_1 and A_2 is smaller than A_1 and B_1. This example proves that it is a better choice to use the nuclear norm in our method.

Next, to gain an intuitive understanding and analyze the properties of RER, we implemented experiments on a synthetic dataset, i.e., torus data, which contains two classes (remarked as red and blue in Figure 4.3, respectively). Every sample includes 5 features: the first and last features are noise features while the rest are distributed in a two-moon shape. For each class, we randomly assign 30 labeled samples and the rest as the unlabeled set.

It can be seen in Figure 4.3 (a) and (b) that the projector matrix W does not include noise features, this means RER successfully selected the right features due to $L_{2,1}$-norm in the second term in (5.7). Comparison between (e) and (f) shows that Z in $W^T XZ$ succeeds in extracting the manifold structure. This shows that the structure is extracted from noiseless data through the third term with condition $X = XZ + E$.

The visualized separability is also confirmed in Figure 4.3 (d) and (f). This is also the effectiveness of the third term as label propagation.

Figure 4.3 (g) and (h) show that the separability of two classes is enhanced by adopting L_Z instead of L_S. Specifically, the similarity graph L_Z obtained by our method has a clearer and cleaner block structure than the pre-defined graph L_S since the noisy points and outliers in (h) are removed by the constraint $X = XZ + E$ and each sample point in (g) can be continuously refined in the process of label propagation by the adaptive graph L_Z.

4.2.3 Optimization Solution

Since it is impossible to simultaneously obtain the optimal solutions for variables F, W, Z and E in the model (5.7), thus, we use alternative iterations, that is, updating each variable by fixing the other variables. In this subsection, we present an algorithm that uses the alternating direction method of multipliers (ADMM) [71]. We first introduce two auxiliary variables K and S to separate the objective function as follows:

$$\begin{gathered}\min_{F,W,Z,E,K,S} \|K\|_* + \lambda_1\|W\|_{2,1} + \lambda_2 \mathrm{tr}\left(FL_SF^T\right) + \lambda_3\|E\|_{2,1} \\ \text{subject to } F_l = Y_l, X = XZ + E, K = F - W^TXZ, \\ Z = S, \mathrm{diag}(S) = \mathbf{0}, S \geq 0, S^T\mathbf{1} = 1.\end{gathered} \tag{4.2}$$

Then, we minimize the augmented Lagrangian function of the problem (5.8) as follows:

$$\begin{aligned}&\Gamma\left(F, W, Z, E, K, S\right) = \|K\|_* + \lambda_1\|W\|_{2,1} \\ &+\lambda_2 \mathrm{tr}\left(FL_SF^T\right) + \lambda_3\|E\|_{2,1} \\ &+\frac{\mu}{2}\left\|X - XZ - E + \frac{M_1}{\mu}\right\|_F^2 + \frac{\mu}{2}\left\|F - W^TXZ - K + \frac{M_2}{\mu}\right\|_F^2 \\ &+\frac{\mu}{2}\left\|Z - S + \frac{M_3}{\mu}\right\|_F^2 - \frac{1}{2\mu}\left(\|M_1\|_F^2 + \|M_2\|_F^2 + \|M_3\|_F^2\right),\end{aligned} \tag{4.3}$$

where M_1, M_2, and M_3 are Lagrangian multipliers and μ>0 is the penalty parameter. We update the variables alternately with others fixed by solving the sub-problems:

The F sub-problem: We first split the Laplacian matrix L_S into four blocks after the lth row and column as $L_S = \begin{bmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{bmatrix}$. When the other variables are fixed

except F, problem (5.9) can be converted to:

$$\min_{F,F_l=Y_l} \lambda_2 \mathrm{tr}\left(FL_SF^T\right) + \frac{\mu}{2}\left\|F - W^TXZ - K + \frac{M_2}{\mu}\right\|_F^2. \tag{4.4}$$

Since $F_l = Y_l$, the optimal F_u can be obtained by taking the partial derivative of (4.4) with respect to F_u and set it to zero as follows:

$$\begin{aligned}\frac{\partial L\left(F_u\right)}{\partial F_u} &= 2\lambda_2 F_u L_{uu} + 2\lambda_2 Y_l L_{ul}^T + \mu F_u - \mu\left(K_u + W^TXZ_u - \frac{M_{2,u}}{\mu}\right) \\ &= \mathbf{0}\end{aligned} \tag{4.5}$$

Then we have:

$$F_u = \left(2\lambda_2 L_{uu} + \mu I_n\right)^{-1}\left(\mu\left(K_u + W^TXZ_u - \frac{M_{2,u}}{\mu}\right) - 2\lambda_2 Y_l L_{lu}\right). \tag{4.6}$$

The W sub-problem: When other variables are fixed except W, problem (5.9) can be converted to:

$$\min_W \lambda_1 \|W\|_{2,1} + \frac{\mu}{2}\left\|F - W^TXZ - K + \frac{M_2}{\mu}\right\|_F^2. \tag{4.7}$$

From the definition of $L_{2,1}$-norm, we first define diagonal matrix D_W as follows:

$$\left(D_W\right)_{jj} = \frac{1}{2\left\|\boldsymbol{w^j}\right\|_2}. \tag{4.8}$$

Thus we can rewrite (4.7) as

$$\min_W \lambda_1 \mathrm{tr}\left(W^TD_WW\right) + \frac{\mu}{2}\mathrm{tr}\left(Q - W^TXZ\right)^T\left(Q - W^TXZ\right), \tag{4.9}$$

where $Q = F - K + \frac{M_2}{\mu}$. The optimal W can be obtained by taking the partial derivative of (4.9) with respect to W and set it to zero as follows:

$$\frac{\partial L\left(W\right)}{\partial W} = 2\lambda_1 D_W W - \mu XZQ^T + \mu ZXZ^TX^TW. \tag{4.10}$$

Then we have:

$$W = \left(2\lambda_1 D_W + \mu XZZ^TX^T\right)^{-1}\left(\mu XZQ^T\right), Q = F - K + \frac{M_2}{\mu}. \tag{4.11}$$

The Z sub-problem: When other variables are fixed except Z, problem (5.9) can be converted to:

$$\begin{aligned}\min_{Z} \frac{\mu}{2}\left\|X - XZ - E + \frac{M_1}{\mu}\right\|_F^2 \\ +\frac{\mu}{2}\left\|F - W^T XZ - K + \frac{M_2}{\mu}\right\|_F^2 + \frac{\mu}{2}\left\|Z - S + \frac{M_3}{\mu}\right\|_F^2.\end{aligned} \tag{4.12}$$

We take the derivative of (4.12) with respect to Z and set it to 0. Then we can obtain the following:

$$Z = \left(X^T X + X^T W W^T X + I_n\right)^{-1}\left(X^T L_1 + X^T W L_2 - L_3\right), \tag{4.13}$$

where $L_1 = X - E + \frac{M_1}{\mu}$,$L_2 = F - K + \frac{M_2}{\mu}$, $L_3 = \frac{M_3}{\mu} - S$.

The E sub-problem: When other variables are fixed except E, problem (5.9) can be converted to:

$$\min_{E} \lambda_3 \|E\|_{2,1} + \frac{\mu}{2}\left\|X - XZ - E + \frac{M_1}{\mu}\right\|_F^2. \tag{4.14}$$

The solution of E can be obtained by a shrinkage operator as follows:

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{e}^i} &= \lambda_3 \frac{1}{\|\mathbf{e}^i\|}\mathbf{e}^i + \frac{\mu}{2}\left(\mathbf{e}^i - \mathbf{a}^i\right) \quad \left(\mathbf{e}^i \text{ is the } i\text{th row of } E\right) \\ &= \left(\lambda_3 \frac{1}{\|\mathbf{e}^i\|} + \frac{\mu}{2}\right)\mathbf{e}^i - \frac{\mu}{2}\mathbf{a}^i = \mathbf{0}\end{aligned} \tag{4.15}$$

Thus,

$$\left\|\mathbf{e}^i\right\| = \left\|\mathbf{a}^i\right\| - \tfrac{2\lambda_3}{\mu} \tag{4.16}$$

Substituting the (4.16) to (4.15), under condition $\left\|\mathbf{e}^i\right\| = \left\|\mathbf{a}^i\right\| - \frac{2\lambda_3}{\mu} \geqslant 0$, we have

$$\mathbf{e}^i = \Omega_{\lambda_3/\mu}\left(\mathbf{a}^i\right) := \begin{cases} \dfrac{\left\|\mathbf{a}^i\right\| - \frac{2\lambda_3}{\mu}}{\|\mathbf{a}^i\|}\mathbf{a}^i & \left(\left\|\mathbf{a}^i\right\| \geqslant \dfrac{2\lambda_3}{\mu}\right), \\ \mathbf{0} & \text{(otherwise)} \end{cases} \tag{4.17}$$

where $\Omega(\cdot)$ denotes the shrinkage operator.

Applying this solution to (4.14), we have:

$$E = \Omega_{\lambda_3/\mu}\left(X - XZ + \frac{M_1}{\mu}\right). \tag{4.18}$$

The K sub-problem: Optimizing K using (5.9) when the other variables are fixed is equivalent to the following problem:

$$\min_K \|K\|_* + \frac{\mu}{2}\left\|F - W^T XZ - K + \frac{M_2}{\mu}\right\|_F^2. \tag{4.19}$$

Subsequently, K is solved using the Singular Value Thresholding (SVT) operator [9], as follows:

$$K = \Phi_{1/\mu}\left(F - W^T XZ + \frac{M_2}{\mu}\right), \tag{4.20}$$

where Φ is the SVT operator defined as $\Phi_\tau(A) = U\Phi_\tau(\sum)V^T, \Phi_\tau(\sum) = \mathrm{diag}(\{(\sigma_i - \lambda)_+\})$ for $A = U\sum V^T$.

The S sub-problem: When other variables are fixed except S, optimizing S by (5.9) is equivalent to the following problem:

$$\begin{aligned}
&\min_S \lambda_2 \sum_{i=1}^n \sum_{j=1}^n \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 s_{ij} + \frac{\mu}{2}\left\|Z - S + \frac{M_3}{\mu}\right\|_F^2 \\
=&\min_S \lambda_2 \sum_{i=1}^n \sum_{j=1}^n g_{ij}s_{ij} + \frac{\mu}{2}\|S - H\|_F^2 \\
&(g_{ij} = \lambda_2 \|\mathbf{f_i} - \mathbf{f_j}\|_2^2, H = Z + \frac{M_3}{\mu}) \\
=&\min_S \mathrm{tr}\left(G^T S\right) + \frac{\mu}{2}\|S - H\|_F^2, \quad (G = (\lambda_2 g_{ij})) \\
&\text{subject to } \mathrm{diag}(S) = \mathbf{0}, S \geq 0, S^T\mathbf{1} = 1.
\end{aligned} \tag{4.21}$$

This problem can be solved column-wisely as

$$\begin{aligned}
&\min_{\mathbf{s}_i} \mathbf{s}_i^T \mathbf{g}^i + \frac{\mu}{2}\|\mathbf{s}_i - \mathbf{h}_i\|^2, \\
&\text{subject to } s_{ii} = 0, \mathbf{s}_i \geqslant 0, {\mathbf{s}_i}^T\mathbf{1} = 1.
\end{aligned} \tag{4.22}$$

Therefore, forgetting condition $s_i = 0$ for some i for the time being, and assuming $\mu > 0$, we consider the following equivalent minimization problem:

$$\begin{aligned}
&\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{s}\|^2 - \mathbf{s}^T\mathbf{c}, \qquad \mathbf{c} = \mathbf{h} - \frac{1}{\mu}\mathbf{g} \\
&\text{subject to } \mathbf{s} \geqslant 0, \mathbf{s}^T\mathbf{1} = 1.
\end{aligned} \tag{4.23}$$

In the case of $\mu = 0$, the optimal solution is $\mathbf{s}$ taking one at k only such that $k = \arg\min \mathbf{g}_i$. This can be solved with the Lagrangian function:

$$J = \frac{1}{2}\|\mathbf{s}\|^2 - \mathbf{s}^T\mathbf{c} + \alpha\left(\mathbf{s}^T\mathbf{1} - 1\right) - \beta^T\mathbf{s}, \beta \geqslant 0.$$

The solution satisfies

$$\frac{\partial J}{\partial \mathbf{s}} = \mathbf{s} - \mathbf{c} + \alpha \mathbf{1} - \beta = \mathbf{0}, \tag{4.24}$$

with KKT condition

$$\mathbf{s}^T \mathbf{1} = 1, \mathbf{s} \geqslant 0, \beta^T \mathbf{s} = 0, \beta \geqslant 0. \tag{4.25}$$

Note that the third condition means $\beta_i s_i = 0$ for any i. Thus, if $\beta_i > 0$, then $s_i = 0$ and thus $c_i + \beta_i - \alpha = 0$ from (4.24). If $s_i > 0$, then $\beta_i = 0$ and thus $s_i = c_i - \alpha > 0$.

Combining those two cases we have two index sets covering $\{1, 2, ..., d\}$:

$$\begin{aligned} I &= \{i | s_i = c_i - \alpha_I > 0, \beta_i = 0\}, \\ J &= \{j | s_j = 0, \beta_j = \alpha - c_j \geq 0\}. \end{aligned} \tag{4.26}$$

In addition, from $\mathbf{s}^T \mathbf{1} = 1, \mathbf{s} \geqslant 0$, $\sum_{i=1}^{d} s_i = \sum_{i \in I} s_i = \sum_{i \in I} c_i - \alpha |I| = 1$. Thus, $\alpha = \alpha_I = \frac{1}{|I|} \sum_{i \in I} c_i - \frac{1}{|I|}$.

After all,

$$\begin{aligned} I &= \{i | s_i = c_i - \alpha > 0, \beta_i = 0\}, \\ J &= \{j | s_j = 0, -\beta_j = c_j - \alpha_I \leq 0\}. \end{aligned} \tag{4.27}$$

Note that this is a self-referenced definition, so we cannot solve this directly. Fortunately, we notice also that α_I plays a role of a threshold for I and J. So we have the following algorithm.

Algorithm 1

1. Sort c_i in decreasing order
2. Find the first k satisfying $c_k > \alpha_{\{1,2,...,k\}} \geq c_{k+1}$
3. Return s corresponding $\{c_1, ..., c_k\}$

The M_1, M_2, M_3, μ sub-problems: We update the Lagrange multipliers M_1, M_2, M_3, and parameter μ by solving the following formulas:

$$\begin{aligned} M_1 &\leftarrow M_1 + \mu (X - XZ - E), \\ M_2 &\leftarrow M_2 + \mu \left(F - W^T XZ - K\right), \\ M_3 &\leftarrow M_3 + \mu (Z - S), \\ \mu &\leftarrow \min (\rho \mu, \mu_{\max}), \end{aligned} \tag{4.28}$$

where ρ>1 and $\mu_{\max}$ are the constants.

The steps involved in the concrete solution are presented in Algorithm 2.

Algorithm 2 Iteration Algorithm of RER for (5.8)

Initialization: $k = 0$, $M_{1,k} = M_{1,k} = M_{1,k} = 0$, $\mu_{max} = 10^7$, $\mu_k = 0.01$, $\rho = 1.01$, $\eta = 10^{-7}$, $\varepsilon = 10^{-7}$

Input: original data $X \in \mathbb{R}^{d\times n}$, label matrix $Y_l \in \mathbb{R}^{c\times l}$, parameters λ_1, λ_2, and λ_3

while not converged **do**

1. Update F by solving (5.10)
2. Update W by solving (5.11)
3. Update Z by solving (5.12)
4. Update E by solving (5.13)
5. Update K by solving (5.14)
6. Update S by solving (5.15)
7. Update multipliers M_1, M_2, M_3 and μ by solving (5.16)
8. Convergence check: if $\|X - XZ_{k+1} - E_{k+1}\|_\infty < \varepsilon$ and $\left\|F_{k+1} - W_{k+1}^T XZ_{k+1} - K_{k+1}\right\|_\infty < \varepsilon$ and $\|Z_{k+1} - S_{k+1}\|_\infty < \varepsilon$, stop; else $k = k + 1$

end while

Output: $F^* \leftarrow F_{k+1}$, $W^* \leftarrow W_{k+1}$, $Z^* \leftarrow Z_{k+1}$, $E^* \leftarrow E_{k+1}$.

4.2.4 Computational Complexity and Convergence Analysis

We now analyze the computational complexity of the proposed method. Let d be the dimensionality of data. It's obvious that Steps 1, 2, and 3 have the major computational burden. In Step 1, computing F requires approximately $\mathrm{O}\left(n^3\right)$. In Step 2, computing W requires a matrix inverse operation, which costs approximately $\mathrm{O}\left(d^2n + dnc\right)$. In Step 3, computing Z requires approximately $\mathrm{O}\left(d^2n + dn^2\right)$. In summary, the total computational cost is $\mathrm{O}\left(t\left(n^3 + d^2c + d^2n + dn^2\right)\right)$, where t denotes the number of iterations. We used PCA as a preprocessing step to reduce the computational complexity in the experiments. About the convergence of our methods, it has been proven that ADMM convergences when the number of variables is not over two [71]. However, it is difficult to theoretically prove the solutions of our model converge to the global optimum or generally ensure the convergence of ADMM with over 2 variables. In fact, we will show in the experimental section that the iterative algorithm will converge very fast. Usually, the outer loop of the algorithm will converge within 3 to 10 iterations.

Table 4.2: The summary of objectives of the proposed method and the most related methods

Methods	Objectives	Properties
FME [49]	$\min_{W,F:F_l=Y_l} \left\|W^TX-F\right\|_F^2+\lambda_1\left\|W\right\|_F^2+\lambda_2\mathrm{tr}\left(FL_SF^T\right)$	**L**abel **P**ropagation **M**anifold **L**earning **D**imension **R**eduction
NNLRR [50]	$\min_{W,F,Z,E} \left\|Z\right\|_*+\lambda_2\mathrm{tr}\left(FL_ZF^T\right)+\lambda_3\|E\|_{2,1}$ subject to $F_l=Y_l, X=XZ+E, \mathrm{diag}(Z)=\mathbf{0}, Z\geq 0, Z^T\mathbf{1}=1,$	**L**abel **P**ropagation **M**anifold **L**earning **D**ata **R**ecovery **S**ample **S**election
AWSSL [81]	$\min_{\Theta,S,F} \sum_{i,j}^{n}\left\|\Theta\mathbf{x_i}-\Theta\mathbf{x_j}\right\|_2^2 s_{ij}+\lambda_1\left\|W\right\|_F^2+\lambda_2\mathrm{tr}\left(FL_SF^T\right),$ subject to $F_l=Y_l, S\geq 0, \mathrm{diag}(S)=\mathbf{0}, S^T\mathbf{1}=1, \Theta\geq 0, \Theta=\mathrm{diag}(\theta), \mathbf{1}^T\theta=1$	**L**abel **P**ropagation **M**anifold **L**earning **D**imension **R**eduction **F**eature **S**election
RER	$\min_{W,F,Z,E} \left\|W^TXZ-F\right\|_*+\lambda_1\left\|W\right\|_{2,1}+\lambda_2\mathrm{tr}\left(FL_ZF^T\right)+\lambda_3\|E\|_{2,1}$ subject to $F_l=Y_l, X=XZ+E, \mathrm{diag}(Z)=\mathbf{0}, Z\geq 0, Z^T\mathbf{1}=1,$	**L**abel **P**ropagation **M**anifold **L**earning **D**imension **R**eduction **F**eature **S**election **S**ample **S**election **D**ata **R**ecovery **RoB**ustness against noises

4.2.5 Difference From the Existing Works

For easy comparison, we summarized the objective functions of the related three methods in Table 2. Although our RER superficially seems similar to the existing methods [49, 50, 81], it is essentially different in the following points. (1) *Compared with FME:* our method imposes $L_{2,1}$-norm on both regression term and regularization term while FME utilizes Frobenius norm and L_2 norm for these terms separately, thus FME is not robust to the noise and cannot learn sparse projection to differentiate relevant and irrelevant features. (2) *Compare with NNLRR:* both RER and NNLRR exploit low-rank reconstruction property to boost label propagation and improve the robustness of the algorithm, but RER surmounts the out-of-sample problem from which NNLRR suffers. That is, NNLRR cannot handle new data unlike RER. (3) *Compare with AWSSL:* RER and AWSSL share the same idea to construct an adaptive graph for propagating label information and using special strategies for ranking the importance of features. However, RER performs better since RER introduces the low-rank property for effectively reducing the negative influence caused by noises and outliers. Besides, RER can recover clean data from noisy data owing to the low-rank property that AWSSL does not have. More quantitative comparison is made in Section 4.

4.3 Experiments

In this section, we describe the datasets, experimental settings, and experimental results. First, we evaluated the classification performance of RER compared with semi-supervised methods on benchmark image datasets. Subsequently, we verified the superiority and robustness of the proposed method on noisy face datasets. In addition, we compared the clustering results with related methods. Finally, we conducted a visualization experiment to prove the effectiveness of our algorithm.

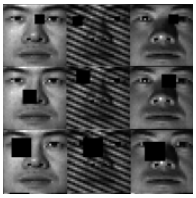
4.3.1 Datasets

Figure 4.4: Sample images. From top to bottom, AR, COIL20, LFW, CMU PIE, and YaleB datasets.

Table 4.3: Characteristics of five benchmark datasets.

Dataset	Image size	#samples	#features	#classes	#samples per class
AR	40 × 50	2400	2000	120	20
COIL20	32 × 32	1440	1024	20	72
LFW	112 × 96	4324	1024	158	>10
CMU PIE	32 × 32	1632	1024	68	24
YaleB	32 × 32	2432	1024	38	64

We evaluated the performance of the proposed RER method on five benchmark image datasets: AR [51], COIL20 [51], LFW [27], CMU PIE [63], and YaleB [19]. The face images of the AR dataset contain real occlusions, such as sunglasses and scarf occlusions. The face images of the CMU PIE, LFW, and YaleB datasets show dramatic variations in poses, expressions, and illumination. The object images of the COIL20 dataset are captured from different viewing directions. The statistics of these datasets are listed in Table 3, and some of the image samples are shown in Figure 4. For



(a) Block

(b) Salt-and-pepper

Figure 4.5: Some examples of corrupted images under varying levels of contiguous occlusions and different percentages of salt-and-pepper noises.

LFW datasets, we first exploited the deep convolutional neural network (CNN) [82] to learn deep features from the original data, and then used these features to train all the methods in our experiments. The final dimension of the LFW dataset was $d = 1024$. For the YaleB dataset, we selected the first 15 people in our experiments. In addition, the images in this dataset were added with different sizes of contiguous occlusions and percentages of the salt-and-pepper noise as follows:

1) Contiguous occlusions: We randomly chose the locations to add the block occlusions on images. The size of the added black blocks was 5×5, 7×7, and 10×10. Figure 5 (a) shows examples of the original and corrupted images.

2) Different percentages corruptions: We corrupted the images by the salt-and-pepper noises with 5%, 10%, and 15% densities. Figure 5 (b) shows examples of corrupted images.

4.3.2 Experimental Settings

For comparison, we selected five semi-supervised learning methods: linear Laplacian regularized least squares (LapRLS/L) [4], flexible manifold embedding (FME) [49], fast flexible manifold embedding (f-FME) [57], semi-supervised orthogonal graph embedding (SOGE) [39], rescaled linear square regression (RLSR) [10], non-negative low-rank representation (NNLRR) [81], auto-weighting semi-supervised learning (AWSSL)[50], robust graph learning (RGL) [31] and semi-supervised adaptive local embedding learning (SALE) with orthogonal constraint [51]. These are summarized as follows:

1. **LapRLS/L** not only captures the relationship between labeled and unlabeled data by constructing a weight graph but also exploits the linear regression function to learn a projection matrix.

2. **FME** is based on LapRLS/L but relaxes the hard linear constraint for better handling the data lying on a nonlinear manifold.

3. **f-FME** is an extended work of FME that replaces the traditional graph with an anchor graph to accelerate the speed of performance and reduce the computation cost.

4. **SOGE** aims to seek an optimal projection under the orthogonal constraint for semi-supervised regression learning.

5. **RLSR** is a semi-supervised regression method that introduces a rescale projection matrix to enhance important features.

6. **NNLRR** combines the LRR and GFHF into a unified framework to perform affinity graph construction and label propagation simultaneously and guarantees the overall optimum.

7. **RGL** learns a graph based on the clean data obtained from recovered technology and then uses the robust graph for enhancing semi-supervised classification.

8. **AWSSL** integrates the adaptive graph construction and label propagation into a unified optimization model and learns an auto-weighting matrix to select important features from all data points.

9. **SALE** with orthogonal constraint adaptively constructs a k_1 Nearest Neighbors graph based on labeled data and a k_2 Nearest Neighbors graph based on the mapped both labeled and unlabeled data for exploring the intrinsic structure of labeled data and global structure of all samples simultaneously.

In addition, we used PCA to preserve 95% of the information for each dataset. For a fair comparison to gain more efficiency and avoid singular solutions, we constructed the weight graph for the methods involving inputs Laplacian matrix L by directly following [39], such as LapRLS/L, FME, and SOGE. For each dataset, we first randomly selected 50% of the data for training and retained the remaining 50% for testing. Then, to simulate a semi-supervised scenario from a training subset, we randomly selected p samples per class as labeled and left the rest unlabeled. Consequently, each image dataset was divided into three parts: labeled, unlabeled, and testing samples. We chose p as 4, 5,

and 6 for AR, COIL20, LFW, CMU PIE, and 30 for YaleB. Such a small number of p values were adopted to clarify the effect of semi-supervised learning. All experiments were repeated ten times for stable classification rates. For the RER algorithm, we tuned all the balanced parameters using a grid search over $\{10^{-6}, 10^{-5}, ..., 10^{5}, 10^{6}\}$. The parameters for the other methods were set according to the corresponding references.

4.3.3 Semi-Supervised Classification

Table 4.4: Accuracy rate and standard deviations (%) of several methods on three image datasets with the different number of labels (p represents the number of labeled data per class and bold fonts mark the best performance).

Dataset	Method	The number of labeled data per class (p)					
		4		5		6	
		Unlabel	Test	Unlabel	Test	Unlabel	Test
AR	LapRLS/L	69.56±7.99	74.71±5.96	74.58±5.27	80.05±6.80	76.54±5.40	82.67±6.91
	FME	91.83±1.45	91.64±2.49	94.36±1.81	94.55±1.95	95.77±1.25	96.24±1.49
	f-FME	93.86±1.11	95.61±0.98	94.67±1.26	96.93±1.06	97.08±1.07	98.14±0.83
	SOGE	84.51±1.43	86.92±1.43	87.65±3.23	90.56±2.21	90.00±2.75	92.71±1.34
	RLSR	74.31±3.71	81.08±3.01	74.93±3.58	87.08±1.45	85.41±2.28	88.34±2.14
	SALE	88.67±2.14	90.78±1.71	91.63±1.89	93.67±1.12	92.56±1.52	94.82±1.20
	RER	**94.72 ± 0.89**	**96.05± 1.27**	**95.21± 0.82**	**97.85 ± 0.74**	**97.62±0.67**	**98.58±0.45**
COIL20	LapRLS/L	66.93±2.52	69.79±2.37	69.70±2.57	73.51±2.54	73.4±1.56	76.25±1.71
	FME	81.09±1.40	82.25±1.58	83.14±2.14	84.56±2.04	85.67±1.73	87.22±1.53
	f-FME	**83.43±2.14**	**84.78±1.67**	**84.23±2.03**	85.62±2.15	86.33±0.21	87.55±2.11
	SOGE	74.31±2.10	76.51±1.08	78.51±2.52	80.18±2.52	78.60±1.95	80.90±1.85
	RLSR	76.25±4.09	78.72±3.76	76.61±2.92	78.98±2.23	78.27±3.78	82.25±3.22
	SALE	79.84±2.64	81.11±1.94	82.88±2.77	84.95±2.21	85.61±2.22	87.58±1.64
	RER	80.19±1.41	81.49±1.75	84.11±1.88	**85.65±2.42**	**87.36±2.24**	**87.90±1.77**
LFW	LapRLS/L	66.09±1.48	64.51±2.73	69.93±0.89	67.88±2.05	71.32±2.00	70.62±0.93
	FME	79.68±2.34	80.53±1.86	83.86±1.43	84.17±1.43	85.44±2.25	86.45±1.06
	f-FME	96.18±0.69	96.46±0.29	96.89±1.02	97.02±0.32	97.21±1.21	97.33±0.23
	SOGE	84.47±1.50	87.03±1.17	86.77±1.19	89.19±0.56	87.02±3.64	90.88±0.77
	RLSR	74.31±2.79	81.09±3.21	78.61±2.75	83.24±1.86	80.68±4.41	85.94±3.11
	SALE	**97.93±0.48**	97.77±0.25	98.32±0.57	98.05±0.21	**98.41±0.80**	98.18±0.17
	RER	97.82±0.45	**98.16±0.16**	**98.41±0.92**	**98.46±0.36**	98.22±0.77	**98.39±0.19**

In this section, in semi-supervised classification tasks, we compared RER with representative dimension reduction methods, including LapRLS/L, FME, f-FME, SOGE, RLSR, and SALE. We employed the 1-nearest neighbor (1NN) classifier with Euclidean distance to evaluate classification accuracy after dimension reduction. Tables 4-7 show the mean classification accuracy (%) with standard deviation and highest accuracy (%) (highlighted in bold) for each method on AR, COIL20, LFW, and YaleB datasets,

respectively. The difference between different base classifiers was not much and the tendency was the same (Table 7).

Table 4.5: Accuracy rate and standard deviations (%) on the YaleB dataset with different corruption percentages ($p = 30$).

Cor.rate		LapRLS/L	FME	f-FME	SOGE	RLSR	SALE	RER
0	Unlabel	39.33±6.63	69.67±8.95	85.33±2.36	73.67±10.35	41.33±8.91	**87.03± 2.87**	86.67±10.93
	Test	69.37±4.71	82.42±5.53	87.91±1.93	85.50±4.59	87.67±3.49	86.60±3.53	**88.12±3.11**
5%	Unlabel	27.67±11.45	65.33±8.49	70.67±11.65	33.67±9.08	34.33±7.03	70.76±4.05	**73.33±11.03**
	Test	62.75±4.89	80.15±5.11	88.45±4.25	60.72±5.01	81.60±5.09	87.79±4.54	**88.50±3.61**
10%	Unlabel	26.67±4.79	60.00±6.08	63.33±8.49	22.67±6.81	25.33±6.51	63.73±4.31	**64.33±8.38**
	Test	59.08±5.48	77.72±5.39	85.02±4.05	55.81±4.89	75.81±5.14	84.08±3.21	**85.18±4.29**
15%	Unlabel	25.00±7.73	56.33±5.97	60.00±8.91	18.67±8.04	25.00±8.92	58.93±4.29	**60.33±8.52**
	Test	58.16±5.09	75.10±5.11	78.10±4.14	54.45±5.27	70.77±5.75	78.08±4.06	**78.97±5.09**

Table 4.6: Accuracy rate and standard deviations (%) on the YaleB dataset with different occlusion sizes of corruption ($p = 30$).

Occ.size		LapRLS/L	FME	f-FME	SOGE	RLSR	SALE	RER
5×5	Unlabel	31.00±8.32	64.00±8.72	**69.33±6.04**	56.00±6.67	47.00±10.71	60.28± 3.84	62.67±8.16
	Test	64.27±3.11	80.85±4.54	95.08±2.27	76.75±2.93	85.02±3.92	92.21±3.99	**95.18±2.11**
7×7	Unlabel	22.67±4.71	53.00±10.71	59.67±6.17	54.00±5.66	46.67±9.53	**60.02±4.32**	60.00±8.46
	Test	62.18±7.50	79.17±4.89	93.12±2.04	74.04±3.16	83.52±2.76	91.85±5.47	**94.37±2.59**
10×10	Unlabel	14.00±5.83	46.34±7.54	46.66±7.73	46.33±8.08	33.33±6.05	46.53±6.61	**46.67±7.69**
	Test	58.14±4.98	76.64±4.06	91.14±2.33	62.33±3.43	80.63±2.34	90.33±1.26	**92.33±1.94**

Table 4.7: Accuracy rate and standard deviations (%) on the COIL20 dataset with different base classifiers ($p = 10$). The bandwidth of RBF kernel is 3.

Classifiers		LapRLS/L	FME	f-FME	SOGE	RLSR	SALE	RER
1NN	Unlabel	83.15±3.63	85.83±1.37	90.00±1.47	83.54±1.83	81.35±1.84	90.06±1.97	**90.12±1.55**
	Test	84.83±1.88	86.43±1.65	92.23±1.24	86.59±1.60	84.44±1.12	92.35±1.00	**92.53±1.34**
SVM	Unlabel	78.09±4.07	87.32±1.77	91.15±1.68	83.09±2.41	85.30±2.59	89.50±1.76	**91.23±1.83**
	Test	80.83±2.98	87.97±1.95	91.50±1.22	86.60±1.83	86.39±3.01	91.50±1.16	**93.57±1.35**

4.3.4 Semi-Supervised Clustering

In this section, we investigated the effectiveness of our RER technique toward clustering tasks, in other words, proper representation ability. We compared RER with the other three representation methods, NNLRR, AWSSL, and RGL, on a number of benchmark datasets.

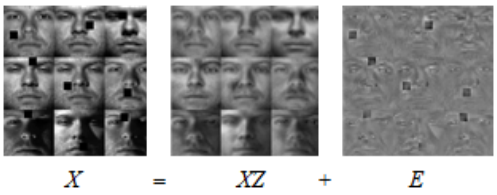


Figure 4.6: The example of using RER to recover the corrupted YaleB face images. Left: the contaminated matrix X. Middle: the corrected data XZ. Right: the error E.

Table 4.8: Predicted accuracy (PreACC) rate and standard deviations (%) of several methods on three image datasets with the different number of labels (p represents the number of labeled data per class and bold fonts mark the best performance).

Dataset	Method	The number of labeled data per class (p)		
		4	5	6
AR	NNLRR	43.33±6.73	47.55±6.17	50.58±6.88
	AWSSL	53.90±0.23	61.98±0.21	71.69±0.05
	RGL	69.24±0.06	73.77±0.05	74.79±0.07
	RER	**74.40 ± 0.02**	**80.27± 0.02**	**86.16± 0.02**
COIL20	NNLRR	75.31±2.68	82.46±2.04	84.68±1.75
	AWSSL	**85.58±0.02**	**87.05±0.02**	87.32±0.01
	RGL	85.31±0.02	86.61±0.02	88.50±0.02
	RER	85.28±0.01	85.83±0.02	**88.75±0.02**
LFW	NNLRR	96.83±0.74	97.56±0.44	97.96±1.07
	AWSSL	97.58±0.01	98.37±0.03	98.99±0.01
	RGL	97.75±0.01	98.33±0.11	**99.05±0.01**
	RER	**97.97±0.01**	**98.39±0.01**	98.80±0.92

We used the proportion of correct-classified data points, namely predicted accuracy (PreACC) to measure. After we get the optimal solution of F in Algorithm 1, the unlabeled data points can be labeled based on the following decision function:

$$\hat{\boldsymbol{f}}_i = \arg\min_{j=1,2,...,c} f_{ij} \quad \forall i = l+1, l+2, ..., n.$$

Let $\hat{\boldsymbol{f}}_{\boldsymbol{i}}$ and $\boldsymbol{y_i}$ are the predicted label and true label of $\boldsymbol{x_i}$, respectively. Then PreACC is defined as follows:

$$\text{PreACC} = \frac{\sum_{i=l+1}^{n} \delta\left(\boldsymbol{y_i}, \hat{\boldsymbol{f}}_{\boldsymbol{i}}\right)}{n},$$

where $\delta(\boldsymbol{y}, \hat{\boldsymbol{f}}) = 1$, when $\boldsymbol{y} = \hat{\boldsymbol{f}}$; $\delta(\boldsymbol{y}, \hat{\boldsymbol{f}}) = 0$, otherwise.

Table 8 shows the mean clustering accuracy with standard deviation (%) and highest accuracy (%) (highlighted in bold) for each method on AR, COIL20, and LFW datasets, respectively.

4.3.5 Experimental Results and Analysis

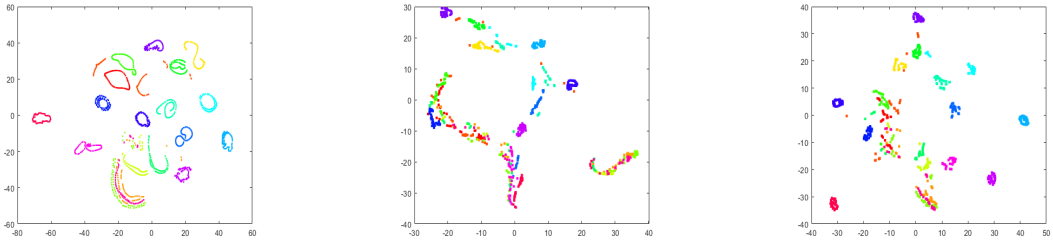
We employed our RER for the challenging task that recovering a clear face image from the images contaminated by block noises. The recovery results of our algorithm on noisy images of the YaleB dataset are shown in Figure 6, in which these images with corruption are approximately recovered.

Based on the experimental results in all the tables and figures, the following conclusions can be drawn.

- Tables 4-8 show the superiority of RER in semi-supervised learning on benchmark datasets, especially on the AR datasets. As the number of labeled samples increases, each algorithm can obtain higher classification rates, but the proposed RER method grows most rapidly among the compared methods. The major reason is that the unified optimization model for SSL significantly brings performance improvement in image classification. In addition, we can observe that RER makes a significant improvement even in clustering tasks (Table 8).

- In some cases, RER is not the best, for example, FME and f-FME work better than RER on COIL20 when $p = 4, 5$. These three methods are similar in terms but different in the norms in their objective functions (Table 2), so that they work together in COIL20. A possible reason for the inferiority of RER is sparsity: RER might impose sparsity too much in this dataset. Another reason is the property of COIL20 which has images of the same object taken from different angles. A shot from different angles is not always regarded as noise that can be appropriately dealt with by RER. SALE performs sometimes best. It constructs two graphs for labeled-labeled pairs and another for labeled-unlabeled pairs while RER constructs a single graph for unlabeled-unlabeled pairs subject to known labeled-labeled pairs. This difference makes one better than the other e.g. on the AR dataset, RER is better than SALE at 6%.

- Tables 5-6 prove the robustness of RER is robust to two kinds of noises. It is worth noting that RER performs less than 26% slight reduction in terms of accuracy as the corruptions become more severe and obvious while the performance of the competitors shows a dramatic drop, e.g. nearly 55% reduction for SOGE.



(a) Original **(b)** FME **(c)** RER

Figure 4.7: Visualization by tSNE of the COIL20 dataset of 20 classes. (a) The original sample distribution, (b) the sample distribution learned by FME, and (c) RER. Different colors represent different classes.

- In the clustering task (Table 8), RER is slightly better than RGL and AWSSL on COIL20 and LFW datasets, and best on AR dataset. The result also shows robustness against image noises because AR images have many real occlusions and illusion variations.

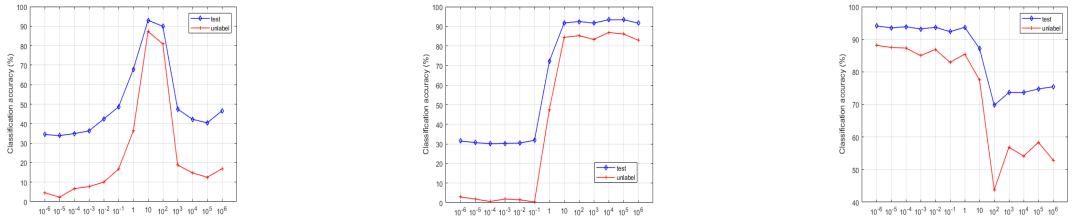
- The experimental results on the LFW dataset show that RER can work with the deeply learned features in classification.

In conclusion, RER is an effective semi-supervised method for both unlabeled training data and testing data, especially in contaminated situations.

4.3.6 Visualization Experiments

In this section, we evaluated the classification performance of the proposed RER method on a real-world image dataset.

We conducted experiments on a real-world image dataset COIL20 for the object classification task. We set $p = 6$ labeled data for each class and the rest as an unlabeled set. Figure 7 illustrates the 2D visualization of the original distribution of data points and the distribution after projecting them by FME and our RER of 20 classes (different colors represent different classes). It can be observed that RER preserves the close structure for each class better than FME in W subspace, suggesting the proposed method can effectively utilize advantages inherited from traditional methods to obtain better classification and clustering performance.



(a) λ_1 **(b)** λ_2 **(c)** λ_3

Figure 4.8: Sensitivity analysis of parameters λ_1 (regularization term), λ_2 (label propagation term), and λ_3 (error term) of RER on AR dataset ($p = 6$). One parameter was changed, whereas the other two were fixed empirically.

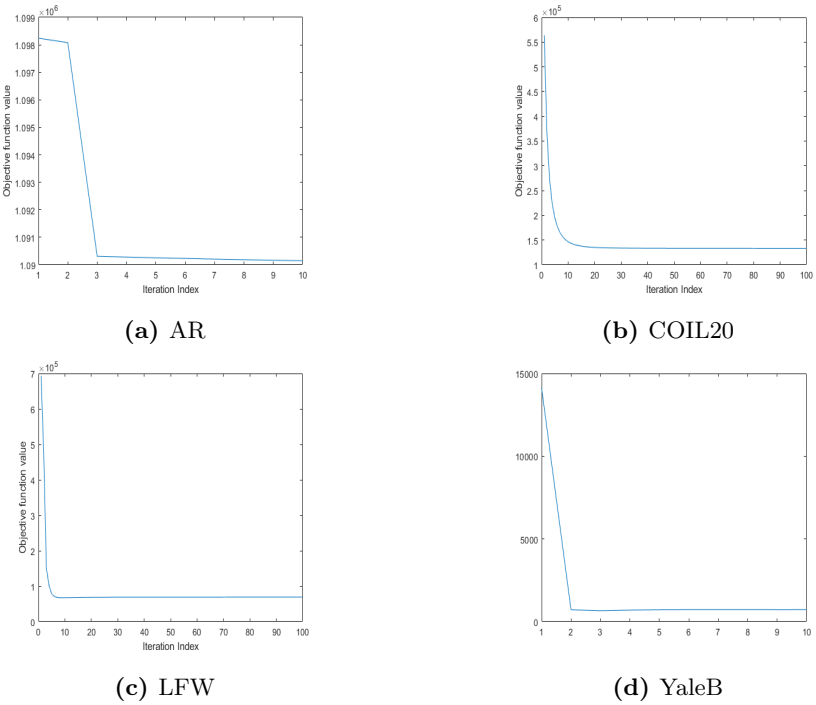
4.3.7 Parameter Sensitivity and Convergence Analysis

Here we conducted a sensitivity analysis on three parameters λ_1, λ_2, and λ_3 of RER. They are brought by sparse projection learning, label propagation, and reconstruction error, respectively. The three parameters are selected from a discrete set $S = \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}, 10^{6}\}$ for each dataset. We utilize the grid search with cross-validation to obtain the optimal parameter combinations on each dataset to obtain the highest classification rates. From Figure 8, we see the accuracy that is a little sensitive to λ_1 only. As a rule of thumb, λ_2 needs to be larger than a threshold θ_2, say $\theta_2 = 10, 100$, λ_3 to be smaller than θ_2, say $\theta_2 = 1, 0.1$, and λ_1 should be carefully chosen in the range $[10^{-3}, 10^{3}]$.

Figure 9 shows the speed of convergence curves of the objective value versus the number of iterations for the three datasets. We can see RER converged very fast within five iterations. Although the objective function of (5.7) is not convex, each sub-problem is convex with respect to variables W, F, Z, E, respectively, which is easy to prove. To the best of the author's knowledge, there is no guarantee of convergence of this type of objective function, although some studies suggest the possibility [38].

4.3.8 Ablation Analysis

For a better insight into the model, an ablation analysis was performed. We designed three inferior versions of RER. First, to analyze the effectiveness of the nuclear norm against the Frobenius norm as a measurement of regression residual, we impose the



(a) AR **(b)** COIL20

(c) LFW **(d)** YaleB

Figure 4.9: Convergence of RER on (a) AR ($p = 4$), (b) COIL20 ($p = 5$), (c) LFW ($p = 6$), and YaleB ($p = 6$).

Frobenius norm on the regression term in (5.7) as follows:

$$\begin{aligned} &\min_{W,F,Z,E} \left\|W^T XZ - F\right\|_F^2 + \lambda_1 \left\|W\right\|_{2,1} + \lambda_2 \mathrm{tr}\left(FL_Z F^T\right) + \lambda_3 \|E\|_{2,1} \\ &\text{subject to } F_l = Y_l, X = XZ + E, \mathrm{diag}(Z) = \mathbf{0}, Z \geq 0, Z^T \mathbf{1} = 1, \end{aligned} \tag{20}$$

We refer to (20) as RER-F. Second, to analyze the effectiveness of the weight graph of RER against the pre-constructed Laplacian graph, we designed a method as follows:

$$\begin{aligned} &\min_{W,F,Z,E} \left\|W^T XZ - F\right\|_* + \lambda_1 \left\|W\right\|_{2,1} + \lambda_2 \mathrm{tr}\left(FL_S F^T\right) + \lambda_3 \|E\|_{2,1} \\ &\text{subject to } F_l = Y_l, X = XZ + E, \end{aligned} \tag{21}$$

where L_S denotes a predefined Laplacian matrix. We refer to (21) as RER-pre.

To analyze the effectiveness of introducing the row-sparsity constraint on the regularization term, we set $\lambda_1 \to 0$, and RER is degraded into

$$\begin{aligned} &\min_{W,F,Z,E} \left\|W^T XZ - F\right\|_* + \lambda_2 \mathrm{tr}\left(FL_Z F^T\right) + \lambda_3 \|E\|_{2,1} \\ &\text{subject to } F_l = Y_l, X = XZ + E, \mathrm{diag}(Z) = \mathbf{0}, Z \geq 0, Z^T \mathbf{1} = 1, \end{aligned} \tag{22}$$

We refer to (22) as RER-λ. We compared three methods and RER on the CMU PIE dataset ($p = 6$) with different block sizes of corruption such as 5×5 and 7×7.

Table 4.9: Accuracy rate and standard deviations (%) on the CMU PIE dataset with different occlusion sizes of corruption.

Occ.size		RER-pre	RER-F	RER-λ	RER
0	Unlabel	71.83±8.56	81.76±3.45	82.13±7.12	**83.50±7.42**
	Test	83.55±4.19	88.28±4.44	88.83±4.02	**89.30±4.28**
5×5	Unlabel	41.69±11.84	51.02±3.45	80.63±6.53	**81.86±7.15**
	Test	58.12±15.58	67.16±3.84	86.15±3.17	**87.62±2.62**
7×7	Unlabel	35.29±13.78	30.88±7.23	80.47±6.64	**80.98±6.54**
	Test	44.73±12.34	50.62±3.73	86.27±4.52	**87.24±4.03**

The experimental results are presented in Table 9. It is evident that RER outperforms its incomplete variants, which demonstrates that every component contributes to its performance. In addition, RER-pre and RER-F perform poorly when encountering corrupted scenarios, which demonstrates that introducing the nuclear norm on the regression term significantly improves the robustness to noisy data, and fully exploiting the correlation between the weight graph and label information can result in higher classification rates.

4.4 Discussion

Although the proposed RER showed the best results in total, there are some potential limitations. 1) We experimented with image datasets only, so it is questionable how RER for the other data domains, such as speech, text, and video. 2) RER has a strong sparsity so there is a risk that RER might be too sparse to obtain sufficient discriminant features for the dataset with less noiseless or few redundant features. 3) RER is a linear method so it cannot handle nonlinearly distributed data in essence. 3) RER does not scale to the large-scale data due to the time complexity of $\hat{\mathrm{O}}(n^3)$. To cope with these limitations, we will examine the applicability of RER to domains other than images in terms of the robustness to noises in those domains. To extend RER to a nonlinear method, we will try to use kernel methods. Some techniques useful to reduce the time complexity and speed up will also be considered, e.g. anchor graph.

4.5 Conclusion

In this paper, we have presented a general framework called RER for semi-supervised learning, in which the robust linear regression, low-rank representation, and semi-supervised learning are elaborately integrated into a unified optimization objective function. RER aims to address the robustness issue of the semi-supervised learning method since the robust norms are selected for our algorithm to earn feature selection and robust regression. Furthermore, our algorithm inherits the advantages of the existing techniques, including label propagation, manifold learning, dimension reduction, feature selection, data recovery, and sample selection. An efficient algorithm based on the augmented Lagrangian method is also proposed for solving the RER model. The experimental results on some datasets show that the proposed RER model outperforms many state-of-the-art semi-supervised approaches, even on the datasets contaminated seriously.

5

Redirected Transfer Learning for Robust Multi-Layer Subspace Learning

In this section, we present the details of the paper named redirected transfer learning for robust multi-layer subspace learning [2].

5.1 Motivations

To deal with the data sharing the same in the task but different in distribution, in recent years, more research [29, 60] focuses on low-rank representation-based transfer learning. The idea is that in some low-rank subspace, the target data could be approximately reconstructed by the neighbor samples of source data, $W^t X_T \approx W^t X_S Z$ in (3.10) so that the domain distribution gap is minimized. One problem in this approach, as already stated, is that it uses zero-one indicators Y in spite of the fact that cross-domain data might distribute differently. Therefore, to cope with this problem, ε-dragging technology or margin-restriction technology is considered [86, 104]. In our model, we propose to directly learn the labels from data and adaptively arrange a more feasible value for each data by enforcing a strong marginal constraint. In addition, we impose $L_{2,1}$-norm not only on the reconstruction error but also on the regularization term, which brings the following two advantages. First, the row-sparsity requirement realized by the $L_{2,1}$-norm in the regularization term works for selecting the most dis-

criminative features. The column-sparsity requirement realized by $L_{2,1}$-norm in the regression term works for reducing the negative influence of noise and outliers residing in the source and target domains. To accomplish the above requirements, we design a multi-layer subspace structure that can decouple the input and output from distinctive distributions, so that the final classification results are not affected by a significant discrepancy of domain distributions.

5.1.1 Model Formulation

Based on the above analysis, we design a multi-layer subspace learning structure to unify two requirements of low-rank reconstruction and redirected label regression into one objective function.

In the first layer, we design a linear mapping W such that the discrepancy of different domains is minimized. In addition, the irrelevant features and noisy samples residing in the domains are eliminated in the first layer. It is realized by minimizing the following objective function:

$$\min_{W,Z} \left\| W^t X_T - W^t X_S Z \right\|_{2,1} + \lambda_1 \|Z\|_* + \lambda_2 \|W\|_{2,1}, \tag{5.1}$$

where $\lambda_1 > 0, \lambda_2 > 0$ are parameters that weigh the importance of low-rank property and sparsity.

In (5.1), the first two terms $\left\| W^t X_T - W^t X_S Z \right\|_{2,1} + \lambda_1 \|Z\|_*$ works for attaining a good domain alignment by reconstructing target data with the lowest-rank representation of source data samples. The $L_{2,1}$-norm makes the solution robust against noise and outliers compared with the Frobenius norm. The regularization term $\lambda_2 \|W\|_{2,1}$ works for feature selection, by which the redundant features and noises are suppressed or even removed.

In the second layer, RTL learns a discriminative projection P for redirected label regression based on robust and low-rank data representation. By combining these two layers into a unified objective function, we obtain the objective function of RTL:

$$\begin{aligned} \min_{W,P,Z,T} & \left\| W^t X_T - W^t X_S Z \right\|_{2,1} + \lambda_1 \|Z\|_* + \frac{1}{2} \left\| T - P W^t X \right\|_F^2 + \lambda_2 \|W\|_{2,1}, \\ & \text{subject to } t_{i,l_i} - \max_{j \neq l_i} t_{i,j} \geq 1, P^t P = I, \end{aligned} \tag{5.2}$$

where T is the redirected label matrix and $X = [X_S, X_T]$. We impose the orthogonal constraint on P to avoid a trivial solution. In this multi-layer subspace learning

structure, the final classification results are not affected by any domain distributions or noises residing in the original data space. It worth noting that the corresponding label matrix of X is defined as $Y = [Y_S, \hat{Y}_T]$ where $\hat{Y}_T \in \{0,1\}^{c\times n_T}$ is the target pseudo labels. Then we force PW^tX_S to be close to the source label Y_S and PW^tX_T to be close to the pseudo label $\hat{Y}_T$ simultaneously. Note that Y gives the basis of the margin condition of T, then we learn a redirected label matrix T with more rich and discriminative information than traditional 0-1 label matrix Y under the margin constraint, thereby boosting the classification performance.

For obtaining pseudo-target labels, we predict $\hat{Y}_T$ from X_T by a base classifier, such as SVM, for PW^tX_T, where P and W are updated to minimize (5.2). The detailed procedure of updating $\hat{Y}_T$ is shown in Algorithm 3 as the outer loop and solving the problem (5.2) is shown in Algorithm 4 as the inner loop, respectively.

Algorithm 3 Outer Loop Iteration Algorithm of RTL for Pseudo-Labels (5.2)

Require: data $X = [X_S, X_T]$, labels of source domains Y_S
Ensure: A^* and $\widehat{Y}_T$

Initialisation: $W = \mathbf{I}$, $P = \mathbf{I}$
while not converged **do**
  Update $\widehat{Y}_T$ using SVM classifier:
  $\widehat{Y}_T = SVM\ classifier(PW^tX_S, PW^tX_T, Y_S)$
  Construct label matrix $Y = [Y_S, \hat{Y}_T]$
  Fix Y and update P and W by Algorithm 4
  $A = PW^t$
  $t = t + 1$
  Check convergence: $\frac{\left\|A_t^{(t)} - A_t^{(t-1)}\right\|_F^2}{\left\|A_t^{(t-1)}\right\|_F^2} < \varepsilon$
end while

Algorithm 4 Inner Loop Iteration Algorithm of RTL for Solving Problem (5.2)

Require: source domain data $X_S \in \mathbb{R}^{m\times n_S}$, target domain data $X_T \in \mathbb{R}^{m\times n_T}$, label of source and target domains $Y \in \mathbb{R}^{c\times n}$, parameters λ_1 and λ_2

Ensure: P and W

Initialisation: $M_1 = M_2 = M_3 = 0$, $\mu_{max} = 10^7$, $\mu = 0.1$, $\rho = 1.01$, $\varepsilon = 10^{-7}$
while not converged **do**
  Update W by solving (5.6)
  Update P by solving (5.9)
  Update A by solving (5.11)
  Update Z by solving (5.13)
  Update E by solving (5.15)
  Update T by solving (5.17)
  Update J by solving (5.19)
  Update M_1, M_2, M_3 and μ by solving (5.20)
  Check convergence: $\|Z - J\|_\infty < \varepsilon$
end while

5.1.2 Optimization

We design an iterative algorithm to update each variable when other variables are fixed. First, we introduce auxiliary variables E, J and A, then convert (5.2) as follows:

$$\begin{aligned}(W^*, P^*, A^*, Z^*, E^*, T^*, J^*) = &\arg \min_{W,P,A,Z,E,T,J} \|E\|_{2,1} + \lambda_1\|J\|_* \\ &+ \frac{1}{2}\|T - AX\|_F^2 + \lambda_2\|W\|_{2,1}, \\ \text{subject to } t_{i,l_i} - \max_{j\neq l_i} t_{i,j} \geq 1, &P^tP = I, Z = J, A = PW^t, \\ &W^tX_T - W^tX_SZ = E.\end{aligned} \tag{5.3}$$

Then the optimization problem (5.3) can be solved by utilizing the alternating direction method of multipliers (ADMM) [7] algorithm. We can obtain the following augmented Lagrangian function:

$$\begin{aligned}L = &\|E\|_{2,1} + \lambda_1\|J\|_* + \frac{1}{2}\|T - AX\|_F^2 + \lambda_2\|W\|_{2,1} \\ &+ \langle M_1, W^tX_T - W^tX_SZ - E\rangle + \langle M_2, Z - J\rangle + \langle M_3, A - PW^t\rangle \\ &+ \frac{\mu}{2}\left(\|W^tX_T - W^tX_SZ - E\|_F^2 + \|Z - J\|_F^2 + \|A - PW^t\|_F^2\right),\end{aligned} \tag{5.4}$$

where M_1, M_2, M_3 are Lagrange multipliers and $\mu > 0$ is a penalty parameter. In order to solve (5.4), we adopt the alternating optimization strategy, i.e., each variable

is optimized by fixing other irrelevant variables. The predicted pseudo-labels of target data are updated in each iteration.

Update W: W can be solved by fixing other irrelevant variables and optimized by the following problem

$$\begin{aligned} W^* = \arg\min_W \lambda_2 \|W\|_{2,1} + \frac{\mu}{2} \left\| W^t X_T - W^t X_S Z - E + \frac{M_1}{\mu} \right\|_F^2 \\ + \frac{\mu}{2} \left\| A - PW^t + \frac{M_3}{\mu} \right\|_F^2 . \end{aligned} \tag{5.5}$$

Then we take partial derivative of (5.5) with respect to W equal to zero, a closed-form solution W^* can be obtained as

$$W^* = \left(2\lambda_2 G + \mu K_1 K_1^t + \mu I\right)^{-1} (\mu K_1 K_2^t + \mu K_3{}^t P), \tag{5.6}$$

where $K_1 = X_T - X_S Z, K_2 = E - \frac{M_1}{\mu}, K_3 = A - \frac{M_3}{\mu}$. $G \in \mathbb{R}^{m \times m}$ is a diagonal matrix and its each diagonal element $(G)_{ii} = \frac{1}{2\left\|(W)^i\right\|_2}$, where $(\cdot)^i$ denotes the ith column of a matrix.

Update P: P can be solved by fixing other irrelevant variables and optimized by the following problem

$$P^* = \arg\min_P \frac{\mu}{2} \left\| PW^t - A + \frac{M_3}{\mu} \right\|_F^2 \qquad \text{subject to } P^t P = I. \tag{5.7}$$

we can convert (5.7) to the following maximization problem:

$$\max_P \frac{\mu}{2} tr\left(\mu K_3 W P^t\right) \qquad \text{subject to} P^t P = I. \tag{5.8}$$

Let the SVD of $\mu K_3 W$. Then, according to [113], the optimal solution to the problem above is

$$P^* = UV^t \ , \tag{5.9}$$

where U, V is the SVD decomposition value of $\mu K_3 W$.

Update A: A can be solved by fixing other irrelevant variables and optimized by the following convex problem

$$A^* = \arg\min_A \frac{1}{2} \|T - AX\|_F^2 + \frac{\mu}{2} \left\| PW^t - A + \frac{M_3}{\mu} \right\|_F^2 . \tag{5.10}$$

Then we take partial derivative of (5.10) with respect to A equal to zero, a closed-form solution A^* can be obtained as

$$A^* = \left(TX^t + \mu\left(PW^t + \frac{M_3}{\mu}\right)\right)\left(XX^t + \mu I\right)^{-1}. \tag{5.11}$$

Update Z: Z can be solved by fixing other irrelevant variables and optimized by the following convex problem

$$Z^* = \arg\min_Z \frac{\mu}{2}\left\|W^tX_T - W^tX_SZ - E + \frac{M_1}{\mu}\right\|_F^2 + \frac{\mu}{2}\left\|Z - J + \frac{M_2}{\mu}\right\|_F^2. \tag{5.12}$$

Then we take partial derivative of (5.12) with respect to Z equal to zero, a closed-form solution Z^* can be obtained as

$$Z^* = \left(X_S^tWW^tX_S + I_{n_s}\right)^{-1}\left[X_S^tW\left(W^tX_T - E + \tfrac{M_1}{\mu}\right)^t + J - \tfrac{M_2}{\mu}\right]. \tag{5.13}$$

Update E: E can be solved by fixing other irrelevant variables and optimized by the following convex problem

$$E^* = \arg\min_E \|E\|_{2,1} + \frac{\mu}{2}\left\|E - W^tX_T + W^tX_SZ - \frac{M_1}{\mu}\right\|_F^2. \tag{5.14}$$

According to [38], the optimal E^* can be computed as

$$(E^*)_{:,i} = \begin{cases} \dfrac{\|K_{:,i}\|_2 - 1}{\|K_{:,i}\|_2}K_{:,i}, \|K_{:,i}\|_2 > \dfrac{1}{\mu}; \\ 0, \qquad \text{otherwise}, \end{cases} \tag{5.15}$$

where $K = W^tX_T - W^tX_SZ + \frac{M_1}{\mu}$.

Update J: J can be solved by fixing other irrelevant variables and optimized by the following convex problem

$$J^* = \arg\min_J \lambda_1\|J\|_* + \tfrac{\mu}{2}\left\|Z - J + \tfrac{M_2}{\mu}\right\|_F^2. \tag{5.16}$$

The optimal J^* can be computed by utilizing singular value thresholding (SVT) algorithm [9] as

$$J^* = \Omega_{\frac{\lambda_1}{\mu}}\left(Z + \tfrac{M_2}{\mu}\right) \tag{5.17}$$

where Ω is the singular value shrinkage operator.

Update T: T can be solved by fixing other irrelevant variables and optimized by the following convex problem

$$T^* = \arg\min_{T} \frac{1}{2} \|T - AX\|_F^2 \qquad \text{subject to } t_{i,l_i} - \max_{j \neq l_i} t_{i,j} \geq 1. \tag{5.18}$$

It is obvious that problem (5.18) can be decomposed into the following sub-problems

$$\min_{t_{i,l_i} - \max\limits_{j \neq l_i} t_{i,j} \geqslant 1} \|T_{i,:} - R_{i,:}\|_F^2, \tag{5.19}$$

where $R = AX$. According to **Theorem 2** in [104], after solving problem (5.19) by solving for all columns of T separately, we can obtain the optimal solution $T_{i,:}$ of problem (5.18).

We also update M_1, M_2, M_3 and μ by

$$\begin{cases} M_1 \Leftarrow M_1 + \mu \left(W^t X_T - W^t X_S Z - E\right), \\ M_2 \Leftarrow M_2 + \mu \left(Z - J\right), \\ M_3 \Leftarrow M_3 + \mu \left(PW^t - A\right), \\ \mu \Leftarrow \min\left(\rho\mu, \mu_{\max}\right), \end{cases} \tag{5.20}$$

where ρ>0 and $\mu_{\max}$ are constants.

5.1.3 Computational Complexity

In Algorithm 4, the computation cost of RTL mainly burdens in the following steps:

1) Matrix inversion and multiplication in steps 3 (optimizing W), 5 (optimizing A), and 6 (optimizing Z), which involve the computational cost of $O\left(m^3\right)$, $O\left(m^3\right)$, $O\left(n_S^3\right)$, respectively, in each iteration.

2) SVD in steps 4 (optimizing P) and 8 (optimizing J), which involve the computational cost of $O\Big(\min\left(c, d\right)^3\Big)$ and $O\left(n_S^3\right)$ respectively, in each iteration.

Suppose the number of iterations for Algorithm 3 and Algorithm 4 is t_{outer} and t, respectively. Since $d \ll m$, the total computational complexity of RTL is at most $tt_{outer}\left(O\Big(\min\left(c, d\right)^3\Big) + 2O\left(m^3\right) + 2O\left(n_S^3\right)\right)$. It is obvious that RTL is not fast enough on large-scale datasets, but it works well with deep features of large-scale datasets extracted by deep learning methods, which is proven in the following experiments.

5.1.4 Convergence Analysis

The convergence of ADMM with less than three variables has been proved in the condition that the objective function is smooth [7]. However, it is difficult to prove the convergence of our method since there are over three variables (seven variables in Algorithm 4). Also, the objective function in (5.2) is not absolutely smooth. Fortunately, according to [16, 89], there are three sufficient conditions to provide some assurances about the convergence property of the proposed method.

(1) The parameter μ in step 10 should be upper bounded.

(2) The so-called dictionary D (X_S denotes as D in our paper) should be of full column rank.

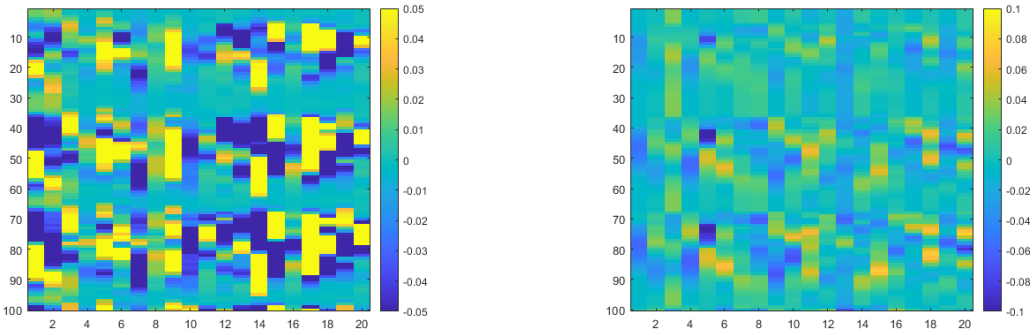
(3) For each iteration step, the optimal gap produced by $\varepsilon_k = \|(Z_k, J_k) - (Z^*, J^*)\|_F^2$ is monotonically decreasing, where Z_k and J_k denote the solutions obtained at kth iteration, respectively, and Z^* and J^* represent the optimal solutions of the model $\arg\min_{Z,J} L$.

Although the third condition is difficult to satisfy, we show some experimental evidence to prove it does hold convergence in the previous Section. Algorithm 3 as the outer loop and Algorithm 4 as the inner loop seem to be independent, but we provide sufficient evidence that these two algorithms can promote each other and gradually converge by simultaneously optimizing the variables P and W and $\widehat{Y}_T$ in Figure 5.4. When the vertical interval is quite large (right column in Figure 5.4), we can see that our objective values reduce reasonably well (less 10 iterations). When the vertical interval is very small (left column in Figure 5.4), the objective function value quickly decreases as the number of iterations increases and then has slight fluctuation.

5.1.5 Differences from Previous Work

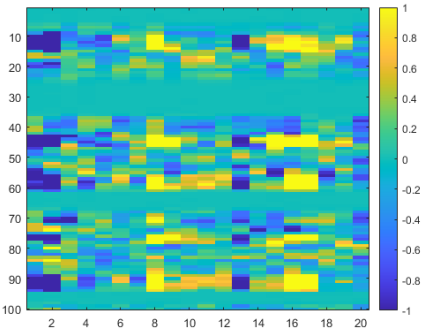
Two previous methods TSL_LRSR [89] and LET [22] are close to the proposed RTL. The objective function of TSL_LRSR is as follows:

$$\begin{aligned} \min_{W,Z,M} & \left\|W^t X_T - W^t X_S Z\right\|_F^2 + \|Z\|_* + \|Z\|_1 \\ & + \frac{1}{2}\left\|W^t X - (Y + B \odot M)\right\|_F^2, \\ & \text{subject to } M \geq 0. \end{aligned} \tag{5.21}$$



(a) TSL_LRSR

(b) LET



(c) RTL

Figure 5.1: Projections learned by TSL_LRSR, LET, and RTL for the task $C1 \rightarrow C2$. Note: all figures are shown in the HSV color space. For better comparison, we only plot the first 100 rows of these projections. From the colorbar, we can infer all element values of these projections. It is seen that the element values in some rows are equal or approximate to zeros in (c).

The objective function of LET is as follows:

$$\begin{aligned}\min_{W,P,M} \frac{1}{2}\left\|PW^tX-(Y+B\odot M)\right\|_F^2+\lambda_1\|P\|_*+\frac{1}{2}\lambda_2\|P\|_F^2\\ +\lambda_3\mathrm{tr}\left(W^tXLX^tW\right),\\ \text{subject to } M\geq 0, W^tW=I.\end{aligned} \tag{5.22}$$

They are similar to RTL in the way of domain alignment and multi-layer subspace learning, respectively, but different in some points. (1) The norm is replaced with $L_{2,1}$-norm from Frobenius norm. Therefore, RTL is more robust than TSL_LRSR. (2) RTL exploits redirected label learning technique that is not adopted in TSL_LRSR. (3) Both LET and RTL take multi-layer subspace learning, but only RTL employs the redirected label technology and feature extraction. Besides, the way of diminishing the discrepancy between the source and target domains is different, LET uses the empirical maximum mean discrepancy (MMD) for domain alignment. The low-rank reconstructions used in RTL and TSL_LRSR are more robust.

To give a better insight into the projection constrained with $L_{2,1}$-norm, in Figure 5.1c, we have plotted the 2D projection matrix $W \in \mathbb{R}^{m\times c}$ learned by TSL_LRSR, LET, and RTL for the task $C1 \rightarrow C2$, where $m = 1000$ and $c = 20$ on the COIL20 dataset. For better comparison, we only presented the first 100 rows of these projections. Since only the projection W learned by RTL has enforced sparsity by $L_{2,1}$-norm, some rows of projection W of RTL are equal or close to zeros while the other two projections do not show this phenomenon. Thus, the features of the original data corresponding to these rows will not be selected by RTL. This implies that RTL performs a self-adaptive feature selection and effectively eliminates the negative influence of redundant and noisy data.

5.2 Experiments

To evaluate the classification performance of the proposed method RTL, we conducted extensive experiments and compared RTL with the state-of-the-art approaches, including nearest neighbor (1NN), TCA [53], JDA [41], JGSA [92], TSL_LRSR [89], GSL [96], and LET [22]. We also compared RTL with some deep learning methods, e.g. AlexNet [32] and DDAN [75].

5.2.1 Datasets Introduction

We used four datasets including three image datasets: 4DA [20], CMU PIE [69], COIL20 [41], and one text dataset: Reuters-21578 [42]. We summarize the details of each dataset in Table 5.1.

1) **4DA** is usually used for object image classification, which includes four object domains with different distributions, **A** (Amazon), **W** (Webcam), **D** (DSLR) and **C** (Caltech-256). Each domain contains 10 common classes selected from the 3DA dataset [20] and an extra Caltech 256 dataset. 12 cross-domain object classification tasks are constructed by randomly deploying two domains as source and target domains alternatively, e.g., **C** $\rightarrow$ **A**, **A** $\rightarrow$ **C**, **C** $\rightarrow$ **W**, ..., **D** $\rightarrow$ **W**. We used two sets of features, e.g., shallow features (SURF) and deep features (DeCAF7) which are learned by a convolutional neural network (CNN) [32], for traditional and deep learning image classification separately. The example images of subsets of the 4DA dataset are shown in Figure 5.2 (a).

2) **CMU PIE** is also used for face image classification, which contains 68 persons with 41,368 images as a whole. We selected five subsets to test different methods: e.g., **P1** (C05) with 3332 images, **P2** (C07) with 1629 images, **P3** (C09) with 1632 images, **P4** (C27) with 3329 images and **P5** (C29) with 1632 images. All these face images were cropped to 32 x 32 pixels. 20 cross-domain face classification tasks are constructed by randomly deploying two subsets as source and target domains alternatively, e.g., **P1** $\rightarrow$ **P2**, **P2** $\rightarrow$ **P1**, **P1** $\rightarrow$ **P3**, ..., **P5** $\rightarrow$ **P4**. Furthermore, to verify the robustness of algorithms, we added 5 $\times$ 5 block size of noises on the random position of each image. The example images of subsets of the CMU PIE dataset are shown in Figure 5.2 (b).

3) **COIL20** is one more object image dataset, which contains 20 objects with 72 gray-scale images per object. The objects were placed on an electric turntable and rotated 360 degrees at five-degree intervals. Each image has 32 $\times$ 32 pixels. We selected two subsets **C1** (COIL1) and **C2** (COIL2) in our experiments. 2 cross-domain object classification tasks are constructed by randomly deploying **C1** and **C2** as source and target domains alternatively as follows: **C1** $\rightarrow$ **C2**, **C2** $\rightarrow$ **C1**. The example images of subsets of the COIL20 dataset are shown in Figure 5.2 (c).

4) **Reuters-21578** is usually used for text classification. It has three top categories i.e., *orgs*, *people*, and *place*, each of which is comprised of many subcategories. We

assume that the samples belonging to different subcategories are drawn from different domains and generated 6 cross-domain tasks, i.e., *org* → *people*, *people* → *org*, *org* → *place*, *place* → *org*, *people* → *place*, and *place* → *people*.

Table 5.1: Detailed information of different datasets

Dataset	Subset	Abbr.	Images	Features	Classes
4DA	**A**mazon	A	958	SURF(800) DeCAF7(4096)	10
	Caltech	C	1,123		
	DSLR	D	157		
	Webcam	W	295		
CMU PIE	PIE05	P1	3,332	Pixel(1,024)	68
	PIE07	P2	1,629		
	PIE09	P3	1,632		
	PIE27	P4	3,329		
	PIE29	P5	1,632		
COIL20	COIL1	C1	720	Pixel(1,024)	20
	COIL2	C2	720		
Reuters-21578	orgs	org	1,237	Pixel(4,771)	2
	people	peo	1,208		
	place	pla	1,016		

5.2.2 Experimental Setting

In this experiment, we exploited all the source domain instances during the training process and target domain instances as a testing set. We adopt the traditional 1-Nearest Neighbor classifier (1NN) with Euclidean distance and the Support Vector Machine classifier (SVM) with an RBF kernel as the baseline classifiers to calculate classification accuracy. SVM is trained on the labeled source data and then tested on the unlabeled target data. We used both classifiers 1NN and SVM only in 4DA (SURF) because they showed the same tendency in all datasets. For the other datasets, we used SVM only. SVM is used also to progressively update the pseudo labels $\hat{Y}_T$ of the target

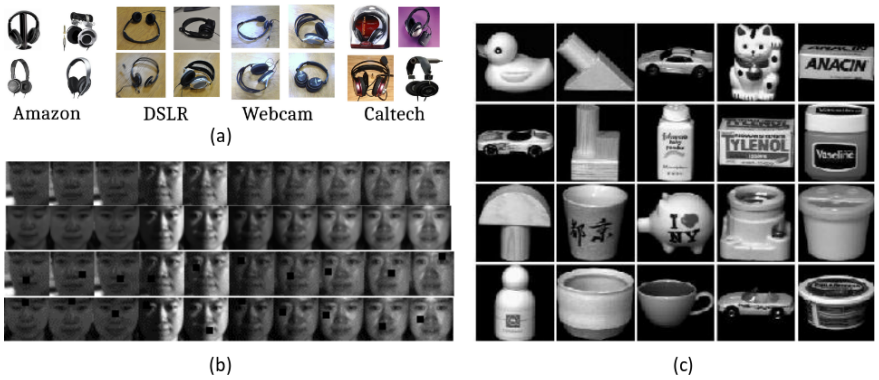


Figure 5.2: Images samples of different datasets, including (a) 4DA dataset, (b) CMU PIE dataset with clean images and corrupted images, (c) COIL20 dataset.

domain. The parameters setting of the SVM classifier is optimized by grid-search, including selecting the penalty term C and bandwidth δ of RBF kernel. For pseudo-label-based methods, JDA, JGSA, and GSL, we also tuned parameters according to their corresponding reference.

There are two parameters in RTL to be tuned, λ_1 and λ_2. We set the subspace dimensionality $d \geq c$ empirically such that the dimension is at least greater than the number of classes, for guaranteeing better classification results. We tuned the parameters λ_1 and λ_2 in the range of $\left[10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}\right]$ by gird-search strategy for all datasets. The optimal parameters in other methods are selected according to the corresponding papers. Please note that the partial results of AlexNet are directly quoted from [46]. To facilitate fairness, we modified the semi-supervised transfer learning method LET as an unsupervised version by utilizing all labeled source data for training and unlabeled target data for testing.

5.2.3 Classification and Evaluation Metric

In transfer learning, there are usually two different domains and corresponding learning tasks, i.e. a source domain $p_s(x, y)$ and learning task T_S, a target domain $p_t(x, y)$ and learning task T_T, the process of classification is using T_S (model training on source data) to help improve T_T (classification of unlabeled target data). 1NN is trained on the labeled source data, and tested on the unlabelled target data. TCA, JDA, JGSA, TSL_LRSR, GSL, LET, and the proposed RTL methods perform on source and target data as a dimensionality reduction procedure. Finally, the nearest neighbor

(NN) classifier or support vector machine (SVM) is trained on the labeled source data for classifying the unlabelled target data. The classification rate on the target task is calculated as

$$Accuracy = \frac{|x : x \in D_T \wedge \hat{y}(x) = y(x)|}{|x : x \in D_T|} \tag{5.23}$$

where D_T is the set of unlabeled target data, $y(x)$ is the truth label of x, $\hat{y}(x)$ is the label predicted by the classification methods.

Table 5.2: Accuracy(%) on the 4DA dataset with SURF features. The best is typed in boldface and * denotes the pseudo-target label-based method.

	NN classifier								SVM classifier			
Source → Target	NN	JDA*	TCA	JGSA*	TSL_LRSR	LET	GSL*	RTL*	JDA*	TSL_LRSR	GSL*	RTL*
A→D	17.83	28.66	28.03	28.66	28.66	30.57	39.49	**43.31**	38.85	33.76	40.76	**48.41**
A→W	15.93	26.10	26.78	26.10	31.53	31.53	35.93	**36.61**	36.27	33.56	36.61	**47.73**
A→C	15.32	33.04	32.06	33.04	42.12	32.77	41.41	**47.73**	42.38	43.99	43.10	**49.15**
D→A	18.79	29.96	29.33	29.96	19.83	17.54	29.02	**37.79**	25.68	27.97	35.07	**39.98**
D→W	32.88	65.42	60.34	60.00	68.81	66.44	63.05	**72.54**	45.08	74.58	**75.93**	75.25
D→C	11.93	29.21	28.58	29.21	21.55	28.41	25.82	**32.15**	28.85	27.87	30.01	**32.50**
W→A	15.66	34.66	28.71	28.50	24.32	30.48	24.95	**40.08**	37.89	33.61	34.97	**42.59**
W→D	30.57	66.88	65.61	64.97	64.33	81.53	70.06	**84.08**	68.79	80.25	83.44	**85.35**
W→C	11.58	29.30	28.05	29.30	21.10	24.31	27.43	**33.57**	33.84	28.58	31.79	**37.22**
C→A	20.35	36.74	35.80	36.74	42.90	39.35	49.69	**53.55**	51.25	51.36	51.25	**57.93**
C→D	12.10	28.03	29.30	28.03	44.59	33.76	42.04	**50.32**	43.95	44.59	45.22	**54.14**
C→W	14.24	26.44	25.42	26.44	35.59	30.85	38.31	**50.85**	45.08	40.00	45.42	**56.27**
Average	18.09	36.20	34.83	35.08	37.11	37.30	40.60	**50.69**	44.08	43.34	46.13	**51.89**

Table 5.3: Accuracy(%) on the 4DA dataset with DeCAF7 features. The best is typed in boldface, and * denotes deep learning methods.

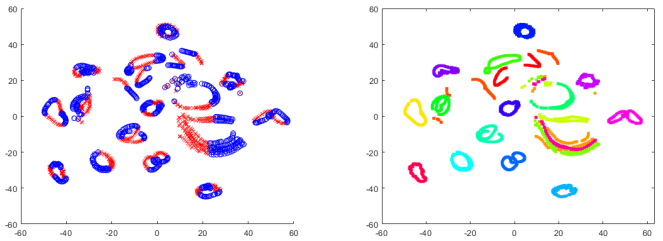
	SVM classifier								
Source → Target	JDA	TCA	JGSA	TSL_LRSR	LET	GSL	AlexNet*	DDAN*	RTL
A→D	84.08	79.62	88.54	86.62	85.99	87.26	87.41	88.53	**96.18**
A→W	74.58	75.25	83.39	76.27	77.97	78.64	79.50	90.85	**92.20**
A→C	83.62	84.24	88.60	85.04	85.49	85.40	83.01	89.10	**89.14**
D→A	87.79	37.47	37.68	38.94	75.99	82.67	87.09	**93.21**	81.11
D→W	99.32	51.53	43.05	52.88	99.32	99.66	97.65	98.64	**99.32**
D→C	79.43	36.69	37.76	39.89	69.01	74.89	79.00	**88.51**	76.22
W→A	75.16	60.44	72.34	56.47	77.66	79.02	83.81	**92.17**	89.87
W→D	**100.00**	90.45	72.61	87.90	**100.00**	**100.00**	**100.00**	93.36	**100.00**
W→C	71.06	35.80	38.82	52.00	70.97	70.79	73.03	**88.96**	84.51
C→A	91.02	76.62	80.69	83.82	92.07	92.12	91.88	93.01	**93.84**
C→D	88.54	78.34	80.25	84.08	85.99	87.26	87.10	89.81	**90.45**
C→W	79.32	64.07	65.76	68.47	78.64	77.97	83.66	**91.19**	84.07
Average	84.49	64.21	65.79	67.69	83.25	84.63	86.10	**91.45**	89.74

Table 5.4: Accuracy(%) on the CMU PIE dataset corrupted by 5×5 block size occlusions. The best is typed in boldface, and * denotes label relaxation strategy-based methods.

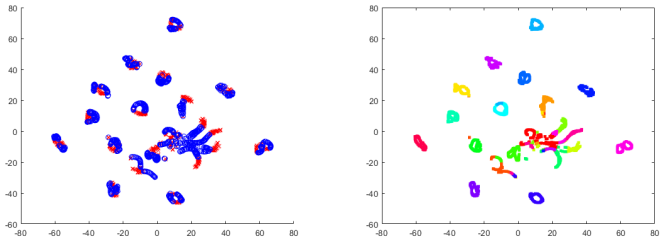
	SVM classifier						
Source → Target	JDA	TCA	JGSA	TSL_LRSR*	LET*	GSL*	RTL*
P1→P2	28.73	6.69	16.02	36.10	42.60	28.73	**51.81**
P1→P3	27.51	6.00	17.52	34.87	48.41	29.60	**50.18**
P1→P4	37.61	10.33	9.16	59.78	71.07	40.70	**82.79**
P1→P5	17.28	4.96	8.64	35.17	39.64	20.04	**43.81**
P2→P1	21.76	6.36	13.63	31.15	44.33	26.74	**54.08**
P2→P3	25.25	4.60	14.89	31.00	45.10	31.50	**48.77**
P2→P4	52.93	5.86	4.75	57.61	73.57	54.97	**76.45**
P2→P5	21.38	6.31	11.34	25.43	**37.01**	23.65	35.48
P3→P1	26.77	5.82	17.14	40.97	46.82	30.67	**58.07**
P3→P2	28.36	7.12	15.22	35.79	45.30	31.31	**48.56**
P3→P4	52.93	6.37	2.34	59.72	66.84	54.40	**70.47**
P3→P5	25.25	4.90	9.93	35.48	46.57	30.76	**52.02**
P4→P1	47.93	7.11	22.99	63.21	81.03	54.65	**87.00**
P4→P2	65.75	6.94	20.99	67.89	75.57	66.05	**79.86**
P4→P3	65.87	7.54	14.64	69.06	73.84	65.87	**78.80**
P4→P5	40.93	5.09	16.30	47.86	63.97	43.26	**67.59**
P5→P1	29.56	2.49	15.22	42.74	44.66	29.98	**51.53**
P5→P2	28.12	3.68	12.28	35.79	40.64	28.30	**44.63**
P5→P3	30.15	3.80	10.05	38.36	48.84	32.90	**53.25**
P5→P4	41.75	2.43	12.77	55.42	58.01	43.86	**67.02**
Average	35.79	5.72	26.58	45.17	54.69	45.38	**60.11**

Table 5.5: Accuracy(%) on the COIL20 dataset. The best is typed in boldface

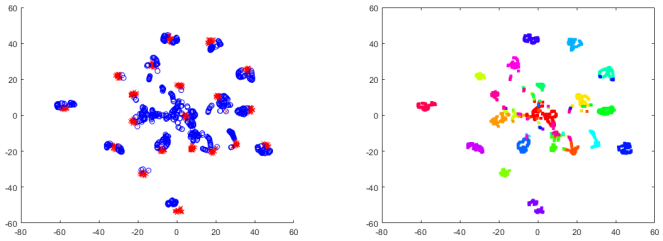
	SVM classifier						
Source → Target	JDA	TCA	JGSA	TSL_LRSR	LET	GSL	RTL
C1→C2	79.03	55.28	53.61	76.39	71.94	80.69	**84.58**
C2→C1	76.39	49.17	54.86	77.22	72.78	82.64	**85.56**
Average	77.71	52.23	54.24	76.81	72.36	81.67	**85.07**



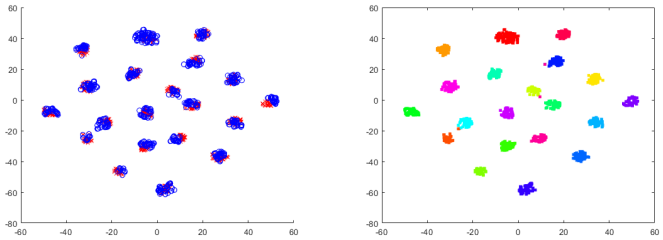
(a) COIL20



(b) TSL_LRSR



(c) LET



(d) RTL

Figure 5.3: The t-SNE visualization of (a) original data, and the extracted features generated by (b) TSL_LRSR, (c) LET, (d) RTL, on the C1→C2 task respectively. In the first column, the red cross 'x' denotes source samples of $C1$ domain, the blue hollow circle 'o' denotes target samples of $C2$ domain, and in the second column, the solid circle '•' with 20 colors denotes the samples corresponding to 20 classes.

Table 5.6: Accuracy(%) on the Reuters-21578 dataset. The best is typed in boldface

	SVM classifier						
Source → Target	JDA	TCA	JGSA	TSL_LRSR	LET	GSL	RTL
org→peo	75.99	73.76	79.80	77.40	61.01	75.50	**80.05**
peo→org	77.12	74.54	82.78	78.66	54.65	74.62	**86.42**
org→pla	71.33	69.45	56.28	60.88	52.06	69.80	**71.14**
pla→org	63.88	62.58	58.17	60.83	49.70	61.71	**85.24**
peo→pla	59.89	58.17	57.66	53.20	49.77	57.38	**63.25**
pla→peo	54.60	**60.54**	**60.54**	58.03	49.58	57.38	58.12
Average	67.14	66.51	66.37	64.83	52.80	66.07	**74.04**

5.2.4 Experimental Results

1) Results on the 4DA dataset with (shallow) SURF features: the classification results on the 4DA dataset with SURF features are shown in Table 5.2, from which we observe that RTL ranks the first (50.69% and 51.89%) in average by applying the NN classifier and SVM classifier, respectively. RTL attained about 5%-10% improvements on average compared to the second-best competitor GSL. It is noteworthy that RTL substantially promotes the classification accuracy on the hard transfer task $D \rightarrow A$. In addition, we observe that predicting pseudo labels works well, as seen in JDA, GSL, and the proposed RTL. We also observe that RTL does not much depend on the classifier, that is, whether SVM or 1NN.

2) Results on the 4DA dataset with (deep) DeCAF7 features: the classification results on the 4DA dataset when DeCAF7 features are used are shown in Table 5.3. It is seen that deep features are more discriminative than shallow features. Among all conventional methods, only RTL is comparable to two deep learning methods and outperforms the other state-of-the-art non-deep transfer learning methods. By comparing with deep transfer learning methods denoted with *, the proposed RTL is merely worse than DDAN with 1.71% in accuracy, while it outperforms the AlexNet with 3.64%. The comparison shows that the proposed RTL, as a shallow learning method, has attractive competitiveness.

3) Results on the CMU PIE dataset with noises: the classification results on the CMU PIE dataset with 5×5 block size noises are shown in Table 5.4. We can see

that the proposed RTL wins 19 out of 20 tasks and outperforms the other competitive method with 60.11% in accuracy. It is further noteworthy that the proposed RTL cannot achieve state-of-the-art performance on the complicated scenario $P2 \rightarrow P5$, mainly because the block noises hide the important features to identify faces, such as the eyes, nose, and ears, and thus the discrepancy between domains becomes large.

4) Results on the COIL20 dataset: the classification results on the COIL20 dataset are shown in Table 5.5 in which RTL shows the best performance (85.07%) in average. To figure out the reason, we visualized extracted features from TSL_LRSR, LET, GSL, and RTL on the adaptation task $C1 \rightarrow C2$ in Figure 5.3. It is worth noting that Figure 5.3 is the 2D projection of data points by applying the technique of t-SNE [67]. In the first column of Figure 5.3, the red cross 'x' denotes source samples of $C1$ domain, the blue hollow circle '$\circ$' denotes target samples of $C2$ domain, and in the second column of Figure 5.3, the solid circle '$\bullet$' with 20 colors denotes the samples corresponding to the different classes. In Figure 5, the first column intuitively reflects the results obtained by different methods by reducing the distribution gap between domains, and the second column shows the visualized results of the classification performance of data of different classes. The smaller the gap between the source domain and the target domain and the more points are clustered according to the corresponding class, the greater the performance of the method. We can see that only in Figure 5.3 (d), the source samples are uniformly distributed and close to the target samples. TSL_LRSR and LET could not divide data points into 20 clusters accurately, while RTL succeeded in making compact clusters and keeping the structure.

5) Results on the Reuters-21578 dataset: the classification results on the text Reuters-21578 dataset are shown in Table 5.6. The Reuters-21578 dataset is a challenging dataset since there are many subcategories for each main category. Thus, it is more difficult to propagate the label information between domains for knowledge transfer. Although the proposed RTL shows slightly inferior to TCA and JGSA in pla→peo task, RTL still achieves strong performance with 74.04% average accuracy on the other complex transfer tasks, which further demonstrates the efficiency of the proposed RTL when encountering a large domain discrepancy.

5.2.5 Convergence and Parameter Sensitivity Analysis

Although Algorithm 3 and Algorithm 4 are dependent, we still need to check the

convergence of RTL by running the Algorithm 3 (outer loop) and Algorithm 4 (inner loop) respectively. We show the convergence curves versus the iteration gap $\Delta A_{t_{outer}}$ in Algorithm 3 (first column of Figure 5.4), and convergence curves versus the value of the objective function in Algorithm 4 (second column of Figure 5.4), respectively. We can obviously find that the value of $\Delta A_{t_{outer}}$ is monotonically decreasing till to the stationary point with the iteration increasing, which proves the convergence property of Algorithm 3. Furthermore, we can observe that Algorithm 3 also converges very fast (less than 10 iterations).

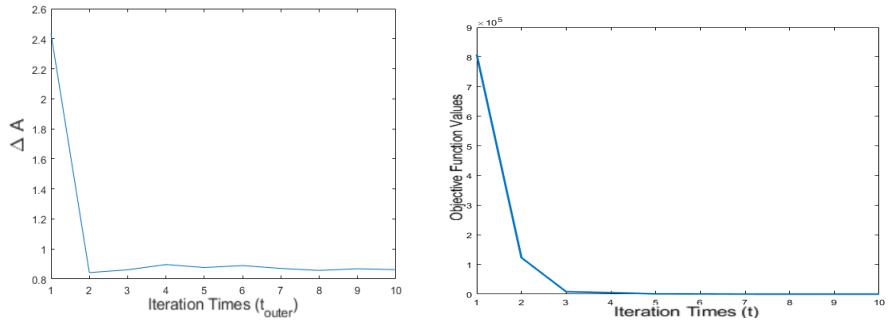
Figure 5.5 provides the classification accuracy of RTL on different cross-domain tasks when parameters λ_1 and λ_2 were selected from the set $[10^{-2}, 10^{-1}, 10^0, 10^1, 10^2]$. We can observe the classification performance of RTL is not sensitive to the variations of λ_1 and λ_2. This also proves that low-rank property (controlled by λ_1) and sparsity (controlled by λ_2) are indispensable for superior performances, and the best classification accuracies are achieved when both parameters are nonzero. In conclusion, our method is robust to the selection of λ_1 and λ_2 to some extent.

5.2.6 Time Comparison

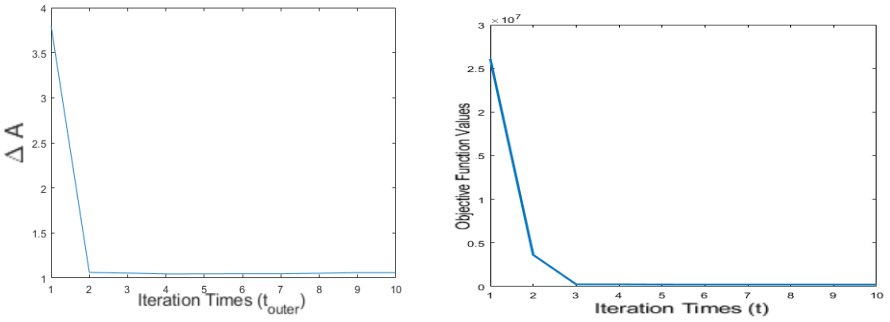
To explicitly present the computational complexity of the proposed method, we take the $C1 \rightarrow C2$ task as an example to test the efficiency of competing methods. The MATLAB codes of all algorithms are obtained from the corresponding authors, and all algorithms were implemented in MATLAB on a 2.90GHz CPU Windows 10 machine with 8GB memory. We train all the algorithms on the $C1$ subset and test on the $C2$ subset. The run time comparisons of different algorithms with respect to the training and test time are listed in Table 7. The low-rank representation-based methods, such as TSL_LRSR, GSL, and RTL, require a lot of time to solve the rankness optimization problem and thus the training speed is too slow while the performance of these methods, especially RTL, is greatly superior to the faster algorithms.

Table 5.7: Run time (s) comparisons of different methods

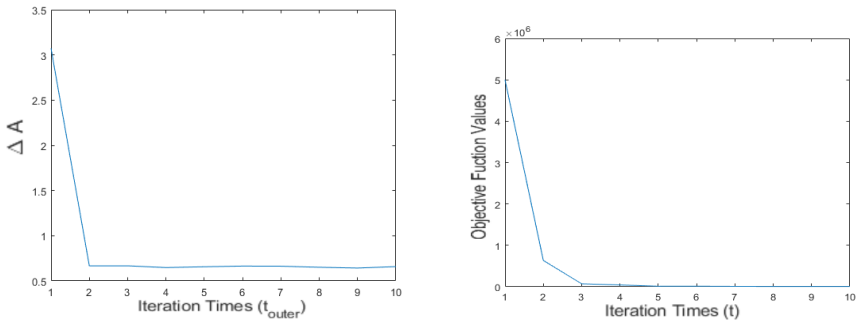
	JDA	TCA	JGSA	TSL_LRSR	LET	GSL	RTL
Train	20.21	24.45	16.07	17.04	13.36	20.71	16.96
Test	0.27	0.30	0.07	0.51	0.46	0.30	0.21



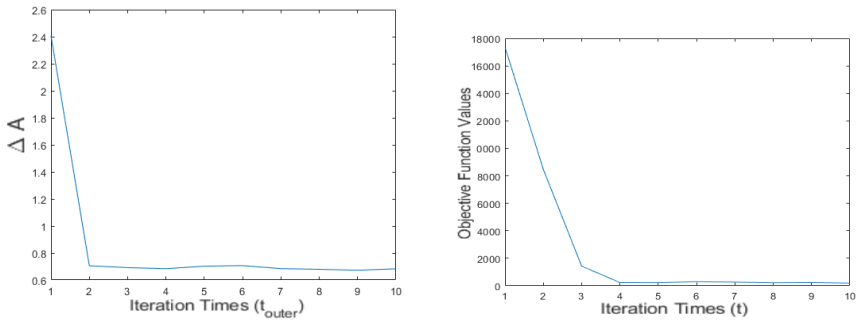
(a) D→W



(b) P3→P4

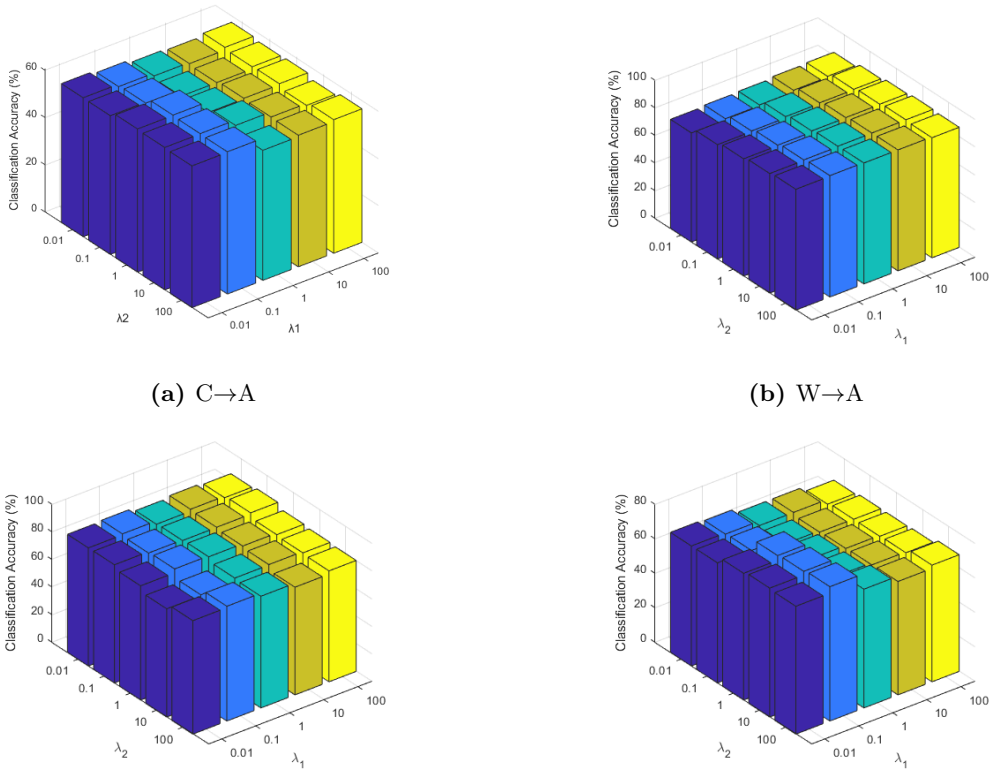


(c) C2→C1



(d) pla→peo

Figure 5.4: Convergence curves on the selected cross-domain datasets. (a) Task D→W in 4DA, (b) Task P3→P4 in CMU PIE, (c) Task C2→C1 in COIL20, (d) Task pla→peo in Reuters-21578.



(a) C→A **(b)** W→A

(c) C2→C1 **(d)** P2→P4

Figure 5.5: Sensitivity of RTL to its parameters λ_1 and λ_2. (a) 4DA (SURF feature), (b) 4DA (DeCAF7 feature), (c) COIL20, and (d) CMU PIE datasets.

5.2.7 Ablation Studies

Table 5.8: Accuracy(%) on the 4DA dataset with SURF features. The best is typed in boldface

SVM classifier				
Source → Target	RTL-A	RTL-B	RTL-C	RTL
A→D	31.21	35.03	36.94	**48.41**
D→W	64.41	66.44	65.79	**75.25**
W→D	73.89	71.97	71.97	**85.35**
C→D	42.68	42.04	40.13	**54.14**
Average	53.05	53.87	53.71	**65.79**

To gain a better insight into each subspace in RTL, we compared the proposed RTL with the single subspace learning structure learning method named RTL-A:

$$\min_{W,Z} \left\|W^t X_T - W^t X_S Z\right\|_{2,1} + \lambda_1 \|Z\|_* + \frac{1}{2}\left\|Y - W^t X\right\|_F^2 + \lambda_2 \|W\|_{2,1}. \tag{5.24}$$

RTL-A learns a single robust subspace but is limited by strict binary labels which results in performance degradation.

In addition, we have another variant named RTL-B:

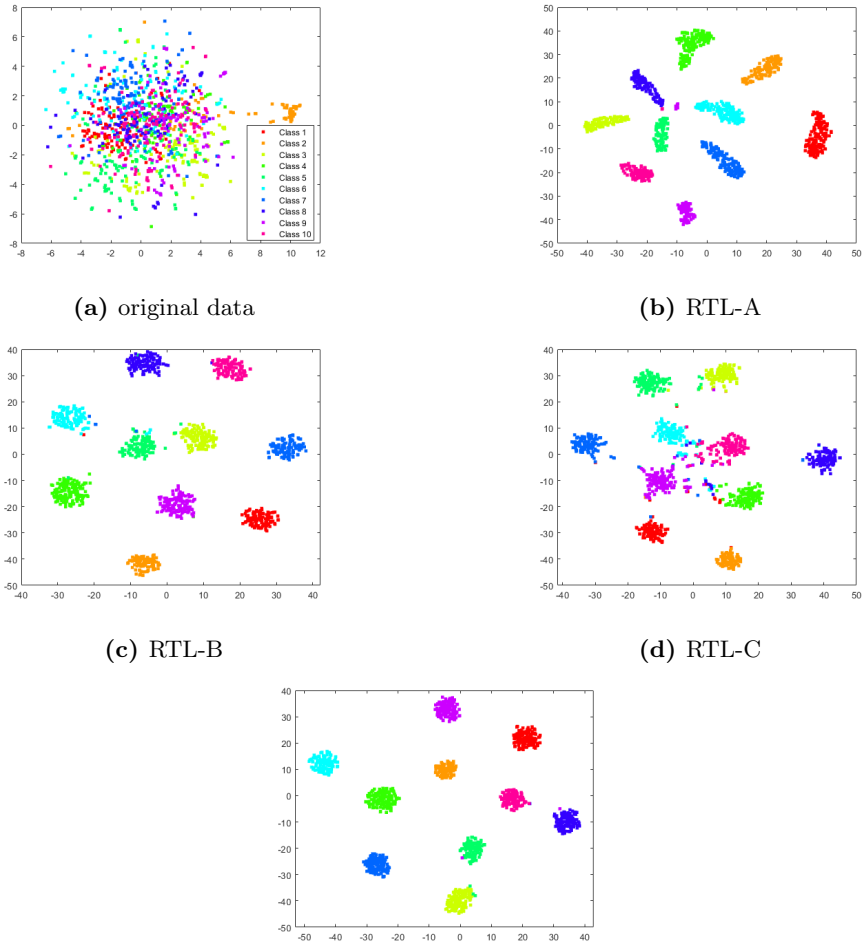
$$\begin{aligned} \min_{W,Z,T} & \left\|W^t X_T - W^t X_S Z\right\|_F^2 + \lambda_1 \|Z\|_* + \frac{1}{2}\left\|T - W^t X\right\|_F^2 + \lambda_2 \|W\|_F^2 \\ & \text{subject to } t_{i,l_i} - \max_{j \neq l_i} t_{i,j} \geq 1. \end{aligned} \tag{5.25}$$

RTL-B learns a single discriminative subspace using the Frobenius norm which results in being sensitive to the noisy data and redundant features residing in domains.

We also verify the effect of the pseudo-label in our method so we design a model named RTL-C which only uses the label of source data for training as follows:

$$\begin{aligned} \min_{W,Z,T} & \left\|W^t X_T - W^t X_S Z\right\|_{2,1} + \lambda_1 \|Z\|_* + \frac{1}{2}\left\|T - W^t X_S\right\|_F^2 + \lambda_2 \|W\|_{2,1} \\ & \text{subject to } t_{i,l_i} - \max_{j \neq l_i} t_{i,j} \geq 1. \end{aligned} \tag{5.26}$$

The results of RTL-A, B, and C of several tasks on the 4DA dataset with shallow features are shown in Table 5.8, from which we can find that each subspace is indispensable. The robust subspace learning of RTL-A has a great impact on performance since large discrepancies existing in the different domains can be efficiently minimized



(a) original data

(b) RTL-A

(c) RTL-B

(d) RTL-C

(e) RTL

Figure 5.6: The t-SNE visualization of RTL on the $A \rightarrow W$ task. The solid circle '•' with 10 colors denotes the samples corresponding to the different classes. There are 10 classes in the 4DA dataset.

and the redundant features and noisy points can be eliminated substantially. Then we observe that the discriminative subspace of RTL-B is also important because the minimum discrepancy is not enough to guarantee the best classification performance. In addition, we find that the performance of RTL-C shows a clear drop compared with RTL, which proves the effectiveness of the pseudo labels.

To obtain a deeper insight into these methods, in Figure 5.6, we present the visualized results for classifying 10 classes samples of task $A \rightarrow W$ from the 4DA dataset. It is worth noting that Figure 5.6 is the 2D projection of data points by applying the technique of t-SNE. The dataset of $A \rightarrow W$ is complex in which there are many noisy and irrelevant data points and the data from different classes overlap together (see Figure 5.6(a)). RTL-A can alleviate the problem with robust norm selection, however, the margins between different classes are not large enough (see Figure 5.6(b)). RTL-B reduces the margins of the intra-class as much as possible and enlarges the margins of the inter-class simultaneously, however, compared to RTL-A, the marginal distribution of RTL-B is somewhat more discrete and more misclassified points, which is due to the lack of redirected label strategy (see Figure 5.6(c)). The class boundary obtained by RTL-C is inferior since only source data labels are available is not enough for unsupervised scenarios while the pseudo-label technique fully considers the labels from all domains and benefits the classification accuracy (see Figure 5.6(d)). As shown in Figures 5.6(e), the data from the same class exhibits a compact and clean structure which proves that lacking any component of RTL will lead to worse classification performance than the full model of (5.2).

5.3 Conclusion

We have proposed a novel transfer learning method named RTL for unsupervised classification. RTL surmounts the disadvantages of existing transfer learning methods in the following three aspects. First, RTL overcomes the limitation of the rigid 0-1 labels by adopting a redirected label matrix of continuous values. This improved the discriminative ability and robustness. Second, RTL employs a multi-layer subspace learning structure so that the final classification results are not affected by the differences in domain distributions or noises residing in the original data space. Third, RTL

pre-constructs the pseudo-label for target domain data to further improve the discriminative ability. The classification experimental results of the proposed method on image datasets show can reach 89.74%, and even on text dataset can reach 74.04%, which is 6.9% higher than the competing methods. Furthermore, we overcome the shortcomings of the existing methods that they are sensitive to noises and achieve a great effect of more than 60% on the corrupted dataset.

In our experiments, we show the computation time of competing methods. However, the training speed is fast only when the number of training samples is relatively small. The computation complexity of RTL is cubic in the size n of training samples. Therefore, the scalability is a limitation of the current RTL. In the future, we will explore more efficient algorithms to improve computational efficiency.

6

Partial Multi-label Learning with Adaptive Dual Graph Disambiguation

6.1 Preliminaries

In Section 3.4, we scrutinize the inherent limitations of conventional graph-based label disambiguation methods in Partial Multi-Label Learning (PML). We identify two primary shortcomings in the existing approaches. Firstly, these methods generally adhere to a two-stage process, segregating the construction of the graph structure from model training. This segregation leads to inaccuracies in label confidence determination and suboptimal classifier performance. Secondly, these methods exhibit a significant vulnerability to the complexities of real-world data, especially in the presence of substantial noise and outliers.

Recent research for partial label learning [72] has suggested using adaptive graph embedding as a more effective alternative for single partial label tasks. This approach is deemed superior in capturing the intrinsic relationship between data and labels and shows enhanced robustness against noisy data, compared to fixed graph-based disambiguation methods. However, in the context of partial multi-label learning tasks, which confront challenges in both high-dimensional feature and label spaces, there is a pressing need for a more robust and efficient method. This method should be capable of simultaneously exploring the feature correlation and label correlation as well as the

correlation between the feature space and the label space.

In response to these challenges, instead of a two-stage learning strategy, we proposed a unified framework, named Adaptive Dual Graph Disambiguation (ADGD), which jointly performs adaptive dual graph learning, candidate label disambiguation, and predictive model induction in an objective function. Furthermore, we integrated two different graphs, a data-based graph and a label-based graph to guide the disambiguation in the PML problem. Finally, we employ $L_{2,1}$-norm on both regression and regularization terms to improve the robustness of noises and reduce the dimensionality of feature space and complexity of label space. In the following parts, we will present the details of the proposed ADGD and introduce an alternative optimization algorithm to solve the objective function.

6.2 Formulation of ADGD

The proposed ADGD method simultaneously learns dual adaptive graphs and a sparse projection matrix in a joint PML framework. These graphs, one capturing instance interrelationships and the other focusing on label correlations, are dynamically updated to better handle label noise and enhance label confidence. Our objective function is illustrated as follows:

$$\begin{aligned}
&\min_{W,F,S^X,S^Y} \left\|X^TW - F\right\|_F^2 + \lambda_1\|W\|_{2,1} + \lambda_2 \sum_{i,j}^{n} s_{ij}^X \left\|\boldsymbol{f_i} - \boldsymbol{f_j}\right\|_2^2 + \lambda_3 \sum_{i,j}^{q} s_{ij}^Y \|\boldsymbol{w_i} - \boldsymbol{w_j}\|_2^2, \\
&\text{subject to } {S^X}^T \mathbf{1}_d = \mathbf{1}_d, 0_{d\times d} \leq S^X \leq n, \\
&\qquad\qquad {S^Y}^T \mathbf{1}_q = \mathbf{1}_q, 0_{q\times q} \leq S^Y \leq n,
\end{aligned} \tag{6.1}$$

where λ_1, λ_2, and λ_3 are the trade-off parameters for sparse regularization, label-based regularization, and instance-based regularization. $W \in \mathbb{R}^{d\times q}$ is a mapping from the feature space into the label space.

However, the loss function is in the form of a squared Frobenius norm, which inevitably results in sensitivity to outliers or noise. Therefore, to address this issue, motivated by (3.3), which avoids the dilemma by jointly minimizing the $L_{2,1}$-norms.

The objective function of RFS with our notations can be written as follows:

$$\begin{aligned}\min_{W,F,S^X,S^Y} &\left\|X^TW-F\right\|_{2,1}+\lambda\left\|F-Y\right\|_{2,1}+\lambda_1\|W\|_{2,1}\\ &+\lambda_2\sum_{i,j}^{n}s_{ij}^X\left\|\boldsymbol{f_i}-\boldsymbol{f_j}\right\|_2^2+\lambda_3\sum_{i,j}^{q}s_{ij}^Y\left\|\boldsymbol{w_i}-\boldsymbol{w_j}\right\|_2^2,\\ \text{subject to } &{S^X}^T\mathbf{1}_d=\mathbf{1}_d, 0_{d\times d}\le S^X\le n,\\ &{S^Y}^T\mathbf{1}_q=\mathbf{1}_q, 0_{q\times q}\le S^Y\le n,\end{aligned}\tag{6.2}$$

From (6.2), we can find ADGD explores both instance graph regularization and label graph regularization, as well as a robust loss function, which has excellent guiding significance for conducting effective partial multi-label feature selection. the first and second terms minimize the discrepancy between the predicted label and the actual label, ensuring the model accurately maps instances to labels. Besides, the joint $L_{2,1}$-norm is also performed to resist outliers. The third term ensures that the projection matrix is sparse, which helps in feature selection and improves the model's interpretability and robustness against irrelevant or noisy features. Regularization in the fourth term performs label disambiguation based on the similarity between data and regularization in the fifth term maintains the local geometric structure of labels according to the manifold assumption, that is if the similarity of label $\boldsymbol{y_i}$ and $\boldsymbol{y_j}$ is large, then $\boldsymbol{w_i}$ and $\boldsymbol{w_j}$ should be similar as well.

6.3 Alternative Optimization

As shown in the previous subsection, the joint optimization problem (6.2) contains four sets of variables with different regularizations and constraints, thus it is hard to be tackled directly. In this subsection, we show that this problem can be solved by applying the alternative optimization. Specifically, each set of variables will be iteratively optimized by fixing other sets of variables until convergence or the maximum number of iterations is reached.

Firstly, through some algebra formulations, the objective function in (6.2) can be transformed into

$$\begin{aligned}\min_{W,F,S^X,S^Y} &\left\|X^TW-F\right\|_{2,1}+\lambda\left\|F-Y\right\|_{2,1}+\lambda_1\|W\|_{2,1}\\ &+\lambda_2\mathrm{tr}(F^TL^YF)+\lambda_3\mathrm{tr}(WL^XW^T),\\ \text{subject to } &{S^X}^T\mathbf{1}_d=\mathbf{1}_d, 0_{d\times d}\le S^X\le n,\\ &{S^Y}^T\mathbf{1}_q=\mathbf{1}_q, 0_{q\times q}\le S^Y\le n,\end{aligned}\tag{6.3}$$

where $L^X = D^X - S^X$ (resp. $L^Y = D^Y - S^Y$) is the graph Laplacian matrix of the instance-level graph (resp. label-level graph), D^X (resp. D^Y) is a diagonal degree matrix whose entries are given by $D^X = \text{diag}\left(\sum_i^d \left(S^X\right)_{ij}\right)$ (resp. $D^Y = \text{diag}\left(\sum_i^q \left(S^Y\right)_{ij}\right)$).

Update W with fixed F, S^X, and S^Y: Since F, S^X, and S^Y are fixed, (6.3) can be reduced to

$$\min_W \left\|X^T W - F\right\|_{2,1} + \lambda_1 \|W\|_{2,1} + \lambda_3 \text{tr}(W L^X W^T). \tag{6.4}$$

Take partial derivative of (6.4) with respect to W and setting it to 0, we obtain

$$\left(X G_1 X^T + \lambda_1 G_3\right) W + \lambda_3 W L_y = X G_1 F, \tag{6.5}$$

where $G_1 \in \mathbb{R}^{n\times q}$ is the diagonal matrix with the iith entries $1/\left\|(X^T W - F)_{i,:}\right\|_2$ and $G_3 \in \mathbb{R}^{d\times d}$ is the diagonal matrix with the iith entries $1/\|W_{i,:}\|_2$. (6.5) is a Sylvester equation [28] and we use the Lyapunov function [8] to optimize W as follows:

$$W \leftarrow \left(X^T G_1 X + \lambda_1 G_3 + \alpha I_{d\times d}\right)^{-1} X G_1 F, \tag{6.6}$$

where α is a tradeoff parameter.

Update F with fixed W, S^X, and S^Y: By fixing other variables, the optimal F can be obtained by minimizing the following problem:

$$\min_F \left\|X^T W - F\right\|_{2,1} + \lambda \left\|F - Y\right\|_{2,1} + \lambda_2 \text{tr}(F L^X F^T). \tag{6.7}$$

Take partial derivative of (6.7) with respect to F and setting it to 0, we obtain

$$F = \left(G_1 + G_2 + \lambda_2 L_X\right)^{-1} \left(G_1 X^T W + G_2 Y\right). \tag{6.8}$$

where $G_2 \in \mathbb{R}^{n\times q}$ is the diagonal matrix with the iith entries $1/\left\|(F - Y)_{i,:}\right\|_2$

Update S^X with fixed W, F, and S^Y: By fixing other variables, the optimal S^X can be obtained by minimizing the following problem:

$$\begin{aligned} &\min_{S^X} \lambda_2 \sum_{i,j}^{d} s_{ij}^X \left\|\boldsymbol{x_i} - \boldsymbol{x_j}\right\|_2^2, \\ &\text{subject to } {S^X}^T \mathbf{1}_d = \mathbf{1}_d, 0_{d\times d} \le S^X \le n. \end{aligned} \tag{6.9}$$

which means that each subproblem of i and j is independent of each other. Then, problem (6.9) can be further simplified to

$$\begin{aligned} &\min_{S^X} \lambda_2 \sum_{i,j}^{d} s_{ij}^X \boldsymbol{t_i}, \\ &\text{subject to } {S^X}^T \mathbf{1}_d = \mathbf{1}_d, 0_{d\times d} \leq S^X \leq n. \end{aligned} \tag{6.10}$$

Accordingly, the optimal solution of (6.10) can be directly determined by k nonzero smallest values in vector $\boldsymbol{t_i}$ according to Section 4.2.3.

Update S^Y with fixed W, F, and S^X: By fixing other variables, the optimal S^Y can be obtained by minimizing the following problem:

$$\begin{aligned} &\min_{S^Y} \lambda_3 \sum_{i,j}^{q} s_{ij}^Y \left\| \boldsymbol{f_i} - \boldsymbol{f_j} \right\|_2^2, \\ &\text{subject to } {S^Y}^T \mathbf{1}_n = \mathbf{1}_n, 0_{n\times q} \leq S^Y \leq n. \end{aligned} \tag{6.11}$$

which means that each subproblem of i and j is independent of each other. Then, problem (6.11) can be further simplified to

$$\begin{aligned} &\min_{S^Y} \lambda_3 \sum_{i,j}^{d} s_{ij}^Y \boldsymbol{h_i}, \\ &\text{subject to } {S^Y}^T \mathbf{1}_q = \mathbf{1}_q, 0_{q\times q} \leq S^Y \leq n. \end{aligned} \tag{6.12}$$

Accordingly, the optimal solution of (6.12) can be directly determined by k nonzero smallest values in vector $\boldsymbol{h_i}$ according to Section 4.2.3.

The pseudocode of ADGD is illustrated in Algorithm 5.

6.3.1 Convergence Proof

In this section, we prove that the iterative optimization converges due to the fact that the optimization is convex in terms of W, F, S^X, and S^Y. The proof of the convergence of Algorithm 5 is as follows:

Theorem 1. *Algorithm 5 converges.*

Proof. For convenience, we denote (6.2) as

$$\begin{aligned} \phi(W, F, S^X, S^Y) = &\left\| X^T W - F \right\|_{2,1} + \lambda \left\| F - Y \right\|_{2,1} + \lambda_1 \|W\|_{2,1} \\ &+ \lambda_2 \mathrm{tr}(F^T L^Y F) + \lambda_3 \mathrm{tr}(W L^X W^T) \end{aligned} \tag{6.13}$$

Algorithm 5 The ADGD Algorithm

Require: Instance data $X \in \mathbb{R}^{d\times n}$, candidate label matrix $Y \in \{0,1\}^{n\times q}$, parameters λ, λ_1, λ_2 and λ_3

1: Initialize $G_1 \in \mathbb{R}^{n\times q}$, $G_2 \in \mathbb{R}^{n\times q}$ and $G_3 \in \mathbb{R}^{d\times q}$ as the identity matrices, build S^X and S^Y, initialize W and label confidence F.
2: **while** not converged **do**
3: Update W by solving (6.6).
4: Update F by solving (6.8).
5: Update S^X by solving (6.10).
6: Update S^Y by solving (6.12).
7: Update $(G_1)_{ii} = 1/\left\|(X^TW - F)_{i,:}\right\|_2$.
8: Update $(G_2)_{ii} = 1/\|(F - Y)_{i,:}\|_2$.
9: Update $(G_3)_{ii} = 1/\|W_{i,:}\|_2$.
10: Convergence or reaching the maximum number of iterations.
11: **end while**

Ensure: Label confidence matrix F, feature weight matrix W, similarity matrices S^X and S^Y of dual graph.

Suppose W_t, F_t, S_t^X and S_t^Y are prepared in the tth iteration. With fixing the F, S^X and S^Y, solving W to yield the optimal W^{t+1} via

$$\phi(W) = \mathrm{tr}(X^TW - F)^TG_1(X^TW - F) + \lambda_1\mathrm{tr}(W^TG_3W) + \lambda_3\mathrm{tr}(WL^XW^T) \quad (6.14)$$

As we know, the standard $L_{2,1}$-norm regularization is convex. Thus, the sum of the three functions is also a convex function. Therefore, we have:

$$\phi(W_{t+1}) \leq \phi(W_t) \quad (6.15)$$

With fixing the W, S^X and S^Y, solving F yields the optimal F_{t+1} via

$$\phi(F) = \mathrm{tr}(X^TW - F)^TG_1(X^TW - F) + \lambda_2\mathrm{tr}(F^TL^YF) \quad (6.16)$$

Since this sub-problem is convex, naturally, we have

$$\phi(F_{t+1}) \leq \phi(F_t) \quad (6.17)$$

Similarly, L^X and L^Y are positive semi-definite, and $F^TL_XF \geq 0$ and $WL_YW^T \geq 0$ hold for any non-zero F and W hold. Thus, the sub-problem (6.10) and (6.12) to variable S^X and S^Y are convex, we can easily have the following inequality:

$$\phi(S_{t+1}^X, S_{t+1}^Y) \leq \phi(S_t^X, S_t^Y) \quad (6.18)$$

Thus, the objective function in (6.2) monotonically decreases and Theorem 6.5 is proved. □

6.3.2 Time Complexity Analysis

In this section, we analyze the computational complexity of Algorithm 5 using big $\mathcal{O}$ notation. We denote T the number of iterations, N is the number of all samples, D the dimensionality of the subspace, and Q the number of classes. The time complexity of ADGD comes from the time spent on constructing feature weight W, similarity matrices S^X and S^Y of the dual graph, which needs totally $\mathcal{O}(N^2M + NM^2)$. The update of label confidence F has a complexity of $\mathcal{O}(N^3 + N^2Q)$. The total complexity for the iterations is $\mathcal{O}(T(N^2M + NM^2 + N^3 + N^2Q))$.

6.4 Experiments

Table 6.1: Characteristics of the PML experimental datasets.

Dataset	#Examples	#Features	#Labels	#Cardinality	Domain
emotions	593	72	6	1.86	music
arts	5,000	462	26	1.64	text
scene	2,407	294	6	1.07	image
birds	645	260	19	1.014	audio
yeast	2,417	103	14	4.23	biology
music_emotion	6,833	98	11	2.42	music
mirflickr	10,433	100	7	1.77	image

6.4.1 Experimental setup

In this section, we conduct experiments on seven representative datasets [99]. These were sourced from diverse real-life applications: scene and mirflickr for annotating images, emotions for categorizing music, arts for text categorization, birds for audio recognition, and yeast for biology classification. The properties of these datasets are summarized in Table 6.1, where # stands for the number of and cardinality represents the average number of labels per instance. For each synthetic data set, we construct partial multi-label assignment by randomly adding the irrelevant noisy labels of each

sample $\boldsymbol{x_i}$ with $\theta\%$ number of ground-truth labels and $\theta\%$ is also randomly assigned by one of 100%, 150%, 200%, and we directly conduct feature selection for the real-world partial multi-label data sets, "Music emotion" and "Mirflickr", without any processing. We select the top $p\%$ of features, the select rate varies from 0% to 50% and the value of p is obtained by dividing 50 into 20 equal parts. To evaluate the robustness of our experiments, we add random feature noise which is defined as the ratio between the number of noisy features with the number of clean features of each dataset. Following [94], we use ML-KNN ($k = 10$) as the classifier to evaluate the performance of the selected feature subset. Finally, we adopt five-fold cross-validation to train the model and record average results and standard deviations. The evaluation metrics we take are average precision (AP), coverage, hamming loss (HL), ranking loss (RL), and coverage (CV) [84]. For the first metric, the value is larger with better performance, and the following three metrics are the opposite.

Let $D = \{(\boldsymbol{x_i}, \boldsymbol{y_i}) \,|1 \leq i \leq n\}$ be a testing set, and $f\left(\boldsymbol{x_i}\right)$ be a multi-label classifier's predicted label set for unseen instance $\boldsymbol{x_i}$.

- Hamming loss:

$$\text{Hamming loss} = \frac{1}{t}\sum_{i=1}^{t}\frac{|f(\boldsymbol{x_i})\oplus \boldsymbol{y_i}|}{q}, \tag{6.19}$$

  where $\oplus$ denotes the symmetric difference between two sets (XOR operation). This metric evaluates the average error rate over all the binary labels.

- Ranking loss:

$$\text{Ranking loss} = \frac{1}{t}\sum_{i=1}^{t}\frac{\left|\left\{\left(\boldsymbol{l_j}, \boldsymbol{l_k}\right) | f_j\left(\boldsymbol{x_i}\right) \leq f_k\left(\boldsymbol{x_i}\right), \left(\boldsymbol{l_j}, \boldsymbol{l_k}\right) \in \boldsymbol{y_i} \times \bar{\boldsymbol{y}}_i\right\}\right|}{\left|\boldsymbol{y_i}\right|\left|\bar{\boldsymbol{y}}_i\right|}, \tag{6.20}$$

  where $f_j\left(\boldsymbol{x_i}\right)$ denotes the jth entry of $f\left(\boldsymbol{x_i}\right)$, and $\bar{\boldsymbol{y}}_i$ denotes the complementary set of $\boldsymbol{y_i}$ in the label set L. This metric evaluates the fraction of reversely ordered label pairs.

- Average precision:

$$\text{Average precision} = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{|\boldsymbol{y_i}|}\sum_{\boldsymbol{l_j}, \boldsymbol{l_k} \in \boldsymbol{y_i}}\frac{|L_i = \{\boldsymbol{l_j} | rank(\boldsymbol{x_i}, \boldsymbol{l_j}) \leq rank(\boldsymbol{x_i}, \boldsymbol{l_k})\}|}{rank(\boldsymbol{x_i}, \boldsymbol{l_k})}, \tag{6.21}$$

  This metric evaluates the average fraction of relevant labels ranked higher than a particular label $\boldsymbol{l_k} \in \boldsymbol{y_i}$.

- Coverage:

$$\text{Coverage} = \frac{1}{q}\left(\frac{1}{t}\sum_{i=1}^{t}\max_{\boldsymbol{l_k}\in\boldsymbol{y_i}} rank(\boldsymbol{x_i}, \boldsymbol{l_k}) - 1\right) \tag{6.22}$$

  This metric evaluates how many steps are needed, on average, to go down the label ranking list so as to cover all the ground-truth labels of the instance. Assume that $rank(\boldsymbol{x_i}, \boldsymbol{l_k}) = \sum_{j=1}^{q} [\![f_j(\boldsymbol{x_i}) \geq f_k(\boldsymbol{x_i})]\!]$ returns the rank of $\boldsymbol{l_k}$ when all labels in L are sorted in L descending order based on f.

6.4.2 Competing Algorithms

ADGD couples with multi-label feature selection methods and partial multi-label learning methods for evaluation. We implemented one partial-label learning method, four state-of-the-art partial multi-label learning methods, and three multi-label feature selection methods, whose detailed definitions are as follows:

- PARTICLE [97] proposes a two-stage PML approach by firstly eliciting labeling confidence through a label propagation procedure and secondly inducing two multi-label predictors named PAR-VLS and PAR-MAP. The parameters are set as $k = 10$, $\alpha = 0.95$, $thr = 0.9$.

- PMLNI [88] jointly learns a noisy label identifier, which identifies feature-induced noisy labels, as well as a multi-label classifier for prediction. The parameters are set as $\lambda = 10$, $\beta = 0.5$, $\gamma = 0.5$.

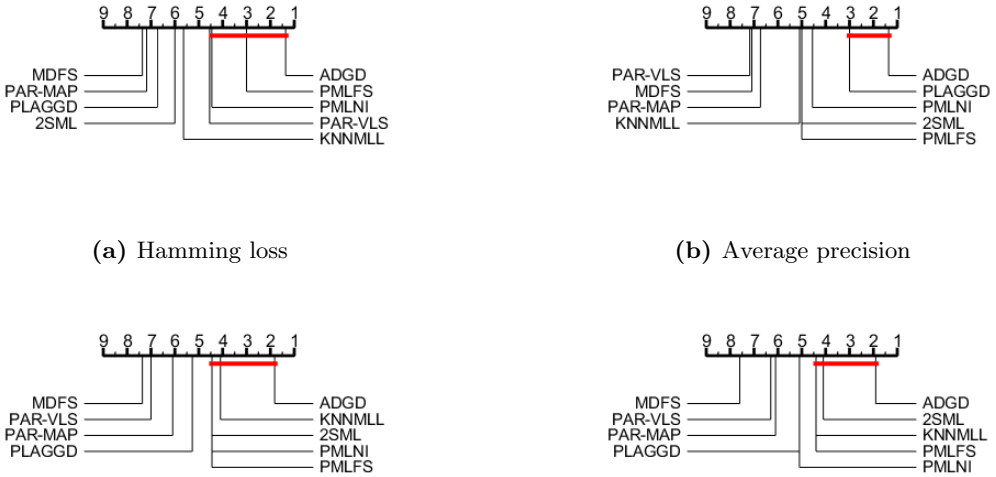
- MLKNN [98] is a nearest neighbor-based multi-label classification method. ML-kNN is a very popular baseline method in multi-label learning literature owing to its simplicity. The parameters are set as $num = 10$, $smooth = 1$.

- MDFS [93] proposes an embedded feature selection method via manifold regularization to select discriminative features for multi-label learning. The parameters are set as $\beta = 10$, γ=0.1.

- 2SML [56] proposes a shared weight matrix with low-rank and sparse regularization for multi-label learning. It utilizes both the feature manifold and label manifold to guide the shared weight learning process. The parameters are set as $\lambda_1 = 10^{-3}, \lambda_2 = 10^{-3}, \lambda_3 = 10^{-4}, \alpha = 0.6, \beta = 1 - \alpha$.

- PL_AGGD [72] proposes to utilize adaptive graph construction to guide label disambiguation and predictive model learning for partial label learning. The parameters are set as $k = 10$, $T = 20$, $\lambda = 1$, $\mu = 1$ and $\gamma = 0.05$.

- PMLFS [76] proposes a partial multi-label feature selection method by combining the $L_{2,1}$-norm regularization and noisy label distinguished strategy. The parameters are set as $\lambda_1 = 2^6$, $\lambda_2 = 2^{-8}$, $\lambda_3 = 2^{-2}$, and $\lambda_4 = 2^{-2}$.



(a) Hamming loss

(b) Average precision

(c) Coverage

(d) Ranking loss

Figure 6.1: Comparison of ADGD against comparing methods with the Nemenyi test. Methods not connected with ADGD in the CD diagram are considered to have a significantly different performance from ADGD (CD = 3.29 at 0.05 significance level)

We use the Friedman test [14] to analyze the significance performance between our proposed method and the other comparative algorithms. Table 6.2 shows the Friedman statistics F_F and each evaluation metric corresponding critical value, which demonstrates the null hypothesis that all the comparing algorithms have an equal performance is rejected at the significant level $\alpha = 0.05$. In other words, there are significant differences between comparison approaches. Further, we employ the Nemenyi test to test whether our method ADGD achieves a competitive performance against the other comparing methods, where ADGD is the control approach.

For Nemenyi test, $q_\alpha = 2.81$, CD =3.29 ($K = 9, N = 11$) at significant level $\alpha = 0.05$. The difference between the average ranking of the two algorithms at least the critical difference CD $= q_\alpha \sqrt{K(K+1)/6N}$ manifests the performance between the two algorithms is significantly different. The CD diagrams of four evaluation metrics are shown in Figure 6.1, in each sub-figure, any algorithm not connected with ADGD implies its performance is significantly different from our proposed method in that metric. Otherwise, if an algorithm is connected to ADGD, it means that the average ranking difference between them is less than one CD, and they would be considered to have no significant difference.

Table 6.2: Friedman statistics F_F in terms of each evaluation metric and the critical value at 0.05 significance level (# comparing algorithms $K = 9$, # datasets $N = 11$).

Evaluation Metric	F_F	Critical Value
Hamming Loss	16.5858	2.0534
Ranking Loss	7.2138	
Coverage	3.9069	
Average Precision	5.0513	

6.4.3 Experimental Results

We report detailed results of each comparing method in terms of Hamming loss, ranking loss, average precision, and coverage in Tables 6.3-6.6. The metrics for each data set are recorded in the form of mean and standard deviation among different percentages. If two algorithms obtain the same performance on one dataset for a given evaluation metric, their ranks are assigned with their average rank value. We report the average rank of each algorithm on all datasets. We also tested the feature selection ability and robustness on the Birds dataset and Yeast dataset separately in Figure 6.3 and Figure 6.2. In summary, we have the following observations:

- As can be seen in 6.3-6.6, ADGD is better than other comparing algorithms on the whole. Concretely, on all datasets (11), across all the evaluation metrics (4), ADGD ranks first in 50.0% cases and ranks second in 27.3% cases. For the two data sets (emotions and scene), ADGD outperforms all comparing state-of-the-art methods. And for arts, music_emotion, and mirflickr datasets, it can also

Table 6.3: Comparison results of methods (mean±std.deviation) in terms of hamming loss. The best results are highlighted in bold, and the number in the bracket indicates the ranking of this algorithm. The last row shows the average ranking of each algorithm on each evaluation metric

Data		ADGD	MLKNN	MDFS	2SML	PMLNI	PAR-VLS	PAR-MAP	PL_AGGD	PMLFS
scene	100%	0.1286+0.0095 (2)	0.2321+0.0056 (6)	0.1797+0.0018 (4)	0.6519+0.0042 (9)	0.1604+0.0076 (3)	0.2139+0.0546 (5)	0.3643+0.0718 (7)	0.6175+0.0052 (8)	**0.1076+0.0035 (1)**
	150%	**0.1532+0.0168 (1)**	0.2682+0.0478 (6)	0.1823+0.0021 (2)	0.6148+0.0074 (8)	0.2196+0.0215 (3)	0.2202+0.0522 (4)	0.4007+0.0692 (7)	0.6943+0.0033 (9)	0.2576+0.0147 (5)
	200%	0.1947+0.1514 (2)	0.3327+0.0163 (6)	**0.1814+0.0026 (1)**	0.6124+0.0138 (8)	0.2690+0.0172 (4)	0.2140+0.0498 (3)	0.3942+0.0667 (7)	0.6975+0.0013 (9)	0.3023+0.0441 (5)
emotions	100%	**0.1935+0.0173 (1)**	0.5442+0.0326 (6)	0.3160+0.0165(2)	0.6000+0.0113 (9)	0.3725+0.0453 (4)	0.3225+0.0131 (3)	0.4177+0.1075 (5)	0.5940+0.0135 (8)	0.5490+0.114 (7)
	150%	**0.1940+0.0234 (1)**	0.6853+0.0062 (9)	0.3108+0.0151 (2)	0.5910+0.0159 (6)	0.4741+0.0092 (5)	0.3345+0.0308 (3)	0.4708+0.0679 (4)	0.6459+0.0053 (7)	0.6806+0.0041 (8)
	200%	**0.1966+0.0101 (1)**	0.6849+0.0102 (9)	0.3095+0.0096 (2)	0.5846+0.0258 (5)	0.6455+0.0285 (6)	0.3158+0.0200 (3)	0.5794+0.0489 (4)	0.6550+0.0122 (7)	0.6840+0.0100 (8)
arts	100%	0.0580+0.0122 (2)	0.0615+0.0019 (5)	0.0629+0.0010 (6)	0.7630+0.1177 (9)	0.0585+0.0011 (4)	**0.0579+0.0007 (1)**	0.0791+0.0083 (7)	0.6880+0.0043 (8)	0.0582+0.0014 (3)
	150%	0.0600+0.0012 (3)	0.0613+0.0005 (4)	0.0629+0.0015 (5)	0.7621+0.3498 (9)	0.0595+0.0014 (2)	0.0632+0.0013 (6)	0.0812+0.0014 (7)	0.7455+0.0044 (8)	**0.0585+0.0007 (1)**
	200%	0.0627+0.0013 (5)	0.0616+0.0005 (4)	0.0629+0.0006 (6)	0.7590+0.3483 (9)	0.0594+0.0019 (2)	0.0598+0.0016 (3)	0.0824+0.0028 (7)	0.7523+0.0036 (8)	**0.0588+0.0018 (1)**
music_emotion		0.1935±0.0035 (2)	0.1975±0.0022 (4)	0.2383±0.0018 (5)	0.7793+0.0014 (9)	0.1960+0.0049 (3)	0.2761+0.0211 (6)	0.3426+0.0034 (7)	0.5961+0.0011 (8)	**0.0588+0.0019 (1)**
mirflickr		**0.0892+0.0033 (1)**	0.1604+0.0049 (4)	0.2574+0.0126 (6)	0.7465+0.0028 (9)	0.1037+0.0050 (3)	0.2567+0.0015 (5)	0.4642+0.1101 (8)	0.4493+0.0036 (7)	0.0953+0.0020 (2)
Avg.Rank		**1.9 (1)**	5.72 (6)	3.72 (3)	8.18 (9)	3.55 (2)	3.83 (5)	6.36 (7)	7.91 (8)	3.82 (4)

Table 6.4: Comparison results of methods (mean±std.deviation) in terms of average precision. The best results are highlighted in bold, and the number in the bracket indicates the ranking of this algorithm. The last row shows the average ranking of each algorithm on each evaluation metric

Data		ADGD	MLKNN	MDFS	2SML	PMLNI	PAR-VLS	PAR-MAP	PL_AGGD	PMLFS
scene	100%	**0.8220+0.0209 (1)**	0.8030±0.0113 (3)	0.5617±0.0308 (7)	0.7956±0.0079 (5)	0.7847±0.0140 (6)	0.4100±0.0090 (9)	0.4259±0.0058 (8)	0.7971±0.0061 (4)	0.8161±0.0078 (2)
	150%	0.7962±0.0450 (2)	0.7870±0.0114 (4)	0.5610±0.0506 (7)	**0.8040+0.0130 (1)**	0.7258±0.0260 (6)	0.3949±0.0344 (9)	0.4247±0.012 (8)	0.7899±0.0090 (3)	0.7738±0.0085 (5)
	200%	**0.8058+0.0359 (1)**	0.7584±0.0141 (4)	0.5352±0.0402 (7)	0.8008±0.0080 (2)	0.7142±0.0170 (6)	0.4308±0.0185 (9)	0.4311±0.0082 (8)	0.7698±0.0076 (3)	0.7581±0.0134 (5)
emotions	100%	**0.8019+0.0272 (1)**	0.7176±0.0254 (5)	0.6535±0.0515 (7)	0.7715±0.0258 (2)	0.7250±0.0346 (4)	0.5452±0.0393 (9)	0.5969±0.0681 (8)	0.7581±0.0207 (3)	0.7061±0.0367 (6)
	150%	**0.7966+0.0282 (1)**	0.6681±0.0188 (6)	0.6230±0.0421 (7)	0.7571±0.0176 (2)	0.6878±0.0103 (4)	0.5325±0.0367 (9)	0.5723±0.0258 (8)	0.7298±0.0172 (3)	0.6736±0.0277 (5)
	200%	**0.7846+0.0120 (1)**	0.6384±0.0269 (4)	0.5861±0.0215 (7)	0.7516±0.0179 (2)	0.6127±0.0270 (6)	0.5131±0.0155 (9)	0.5639±0.0220 (8)	0.6567±0.0207 (3)	0.6271±0.0213 (5)
Arts	100%	**0.6186+0.0128 (1)**	0.4963±0.0084 (7)	0.2026±0.0269 (9)	0.3291±0.0592 (8)	0.5957±0.0068 (3)	0.5956±0.0067 (4)	0.5919±0.0083 (5)	0.6068±0.0094 (2)	0.5292±0.0054 (6)
	150%	**0.6169+0.0082 (1)**	0.4890±0.0109 (7)	0.2024±0.0288 (9)	0.3307±0.0663 (8)	0.5737±0.0093 (5)	0.5919±0.0094 (2)	0.5792±0.0040 (4)	0.5859±0.0115 (3)	0.5051±0.0023 (6)
	200%	**0.6110+0.0073 (1)**	0.4874±0.0094 (7)	0.2161±0.0671 (9)	0.3369±0.0707 (8)	0.5691±0.0105 (5)	0.5987±0.0146 (2)	0.5785±0.0058 (4)	0.5861±0.0076 (3)	0.5172±0.0079 (6)
music_emotion		0.6404± 0.0012 (2)	0.5060±0.009 (5)	0.6140±0.005 (3)	0.4144±0.0053 (8)	0.6050±0.006 (4)	0.3810±0.028 (9)	0.4729±0.021 (6)	**0.6616± 0.0029 (1)**	0.4470±0.0061 (7)
mirflickr		0.8636±0.0101 (3)	0.8615±0.0055 (4)	0.7545±0.0144 (6)	0.4701±0.0055 (9)	**0.9066±0.0052 (1)**	0.4775±0.0840 (8)	0.5144±0.1101 (7)	0.8566±0.0029 (5)	0.8949±0.0060 (2)
Avg.Rank		**1.36 (1)**	5.09 (6)	7.09 (8)	5 (4)	4.55 (3)	7.18 (9)	6.73 (7)	3 (2)	5 (4)

Table 6.5: Comparison results of methods (mean±std.deviation) in terms of ranking loss. The best results are highlighted in bold, and the number in the bracket indicates the ranking of this algorithm. The last row shows the average ranking of each algorithm on each evaluation metric

Data		ADGD	MLKNN	MDFS	2SML	PMLNI	PAR-VLS	PAR-MAP	PL_AGGD	PMLFS
scene	100%	**0.1079±0.0144 (1)**	0.1220±0.0123 (3)	0.3703±0.0142 (7)	0.1178±0.0054 (2)	0.1393±0.0132 (6)	0.4780±0.2214 (8)	0.4871±0.0259 (9)	0.1378±0.0082 (5)	0.1333±0.0069 (4)
	150%	0.1251±0.0472 (2)	0.1462±0.0185 (3)	0.3566±0.0361 (7)	**0.1167±0.0074 (1)**	0.1853±0.0161 (6)	0.6723±0.1065 (9)	0.4929±0.0187 (8)	0.1834±0.0064 (5)	0.1603±0.0043 (4)
	200%	0.1261±0.0287 (2)	0.1562±0.0138 (4)	0.3528±0.0423 (8)	**0.1193±0.0083 (1)**	0.1886±0.0154 (7)	0.1443±0.1271 (3)	0.4773±0.0120 (9)	0.1829±0.0090 (6)	0.1776±0.0108 (5)
emotions	100%	**0.1628±0.0265 (1)**	0.2624±0.0290 (5)	0.3346±0.0435 (7)	0.1663±0.0224 (2)	0.2484±0.0376 (4)	0.5592±0.0497 (9)	0.3932±0.0919 (8)	0.2158±0.0239 (3)	0.2659±0.0319 (6)
	150%	**0.1625±0.0228 (1)**	0.3275±0.0213 (6)	0.3661±0.0429 (7)	0.1738±0.0219 (2)	0.2856±0.0209 (4)	0.5576±0.3564 (9)	0.4132±0.0368 (8)	0.2515±0.0165 (3)	0.3092±0.0293 (5)
	200%	**0.1614±0.0098 (1)**	0.3620±0.0432 (4)	0.4037±0.0172 (7)	0.1759±0.0129 (2)	0.3744±0.0411 (5)	0.6164±0.0508 (9)	0.4448±0.0526 (8)	0.3429±0.0220 (3)	0.3847±0.0182 (6)
arts	100%	0.1240±0.0068 (2)	0.1604±0.0049 (5)	0.4241±0.0498 (9)	0.3647±0.3177 (8)	0.1616±0.0022 (6)	0.1327±0.0043 (3)	**0.1158±0.0036 (1)**	0.1674±0.0047 (7)	0.1540±0.0041 (4)
	150%	0.1342±0.0069 (3)	0.1648±0.0067 (5)	0.4756±0.0509 (9)	0.3634±0.3183 (8)	0.1761±0.0041 (6)	0.1332±0.0053 (2)	**0.1215±0.0042 (1)**	0.1845±0.0076 (7)	0.1616±0.0012 (4)
	200%	0.1396±0.0049 (3)	0.1695±0.0092 (5)	0.4730±0.0789 (9)	0.3633_0.3185 (8)	0.1812±0.0055 (6)	0.1307±0.0047 (2)	**0.1230±0.0021 (1)**	0.1831±0.0076 (7)	0.1649±0.0066 (4)
music_emotion		0.2378±0.0027 (3)	0.2451±0.0086 (4)	0.3614±0.0140 (6)	0.4487±0.0082 (8)	**0.2045±0.0032 (1)**	0.5544±0.0350 (9)	0.3822±0.0270 (7)	0.2304±0.0022 (2)	0.2840±0.0042 (5)
mirflickr		0.0771±0.0058 (3)	0.1007±0.0043 (4)	0.1997±0.0170 (6)	0.4144±0.0045 (7)	**0.0540±0.0037 (1)**	0.6733±0.1113 (9)	0.4194±0.0761 (8)	0.1161±0.0019 (5)	0.0664±0.0055 (2)
Avg.Rank		**2 (1)**	4.36 (2)	7.45 (9)	4.45 (3)	4.73 (5)	6.55 (8)	6.18 (7)	4.83 (6)	4.45 (3)

Table 6.6: Comparison results of methods (mean±std.deviation) in terms of coverage. The best results are highlighted in bold, and the number in the bracket indicates the ranking of this algorithm. The last row shows the average ranking of each algorithm on each evaluation metric

Data		ADGD	MLKNN	MDFS	2SML	PMLNI	PAR-VLS	PAR-MAP	PL_AGGD	PMLFS
scene	100%	**0.1085±0.0101 (1)**	0.1147±0.0129 (3)	0.3246±0.0109 (7)	0.1123±0.0055 (2)	0.1322±0.0099 (5)	0.4493±0.0198 (9)	0.4126±0.010 (8)	0.1377±0.0059 (6)	0.1186±0.0077 (4)
	150%	0.1252±0.0397 (2)	0.1478±0.0056 (3)	0.3122±0.0294 (7)	**0.1114±0.0048 (1)**	0.1707±0.0101 (5)	0.4441±0.0116 (9)	0.4161±0.0132 (8)	0.1834±0.0064 (6)	0.1511±0.0046 (4)
	200%	0.1223±0.0219 (2)	0.1479±0.0128 (3)	0.3090±0.0348 (7)	**0.1139±0.0083 (1)**	0.1766±0.0154 (5)	0.4365±0.0353 (9)	0.4091±0.0137 (8)	0.1859±0.0113 (6)	0.1648±0.0096 (4)
emotions	100%	**0.2985±0.0181 (1)**	0.3954±0.0279 (5)	0.4452±0.0233 (7)	0.3187±0.0179 (2)	0.3787±0.0393 (4)	0.5351±0.0522 (9)	0.4832±0.0812 (8)	0.3594±0.0172 (3)	0.3957±0.0189 (6)
	150%	**0.2989±0.0118 (1)**	0.4483±0.0129 (6)	0.4656±0.0426 (7)	0.3136±0.0237 (2)	0.4134±0.0318 (4)	0.5490±0.0693 (9)	0.4969±0.0350 (8)	0.3893±0.0162 (3)	0.4331±0.0204 (5)
	200%	**0.2977±0.0101 (1)**	0.4744±0.0348 (4)	0.5087±0.0151 (7)	0.3233±0.0148 (2)	0.4785±0.0317 (5)	0.5628±0.0442 (9)	0.5193±0.0263 (8)	0.4590±0.0234 (3)	0.4939±0.0036 (6)
arts	100%	0.1917±0.0098 (3)	0.2206±0.0064 (5)	0.5027±0.0529 (9)	0.3310±0.1407 (8)	0.2378±0.0038 (6)	0.1781±0.0067 (2)	**0.1705±0.0054 (1)**	0.2490±0.0057 (7)	0.2157±0.0070 (4)
	150%	0.2047±0.0124 (3)	0.2252±0.0086 (4)	0.5488±0.0508 (9)	0.3276±0.1308 (8)	0.2539±0.0069 (6)	0.1793±0.0066 (2)	**0.1773±0.0066 (1)**	0.2674±0.0100 (7)	0.2273±0.0032 (5)
	200%	0.2128±0.0083 (3)	0.2331±0.0090 (5)	0.5487±0.0717 (9)	0.3276±0.1316 (8)	0.2586±0.0057 (6)	0.1773±0.0039 (1)	0.1797±0.0033 (2)	0.2665±0.0089 (7)	0.2304±0.0102 (4)
music_emotion		0.4023±0.0075 (2)	0.4031±0.0093 (3)	0.5154±0.0130 (6)	0.6047±0.0110 (8)	**0.3683±0.0043 (1)**	0.6470±0.0220 (9)	0.5548±0.0250 (7)	0.4200±0.0037 (5)	0.4037±0.0023 (4)
mirflickr		**0.0790±0.0069 (1)**	0.2198±0.0049 (4)	0.3055±0.0140 (6)	0.4478±0.0032 (7)	0.1675±0.0025 (2)	0.5070±0.0609 (9)	0.4642±0.1032 (8)	0.2596±0.0040 (5)	0.1805±0.0047 (3)
Avg.Rank		**1.82 (1)**	4.09 (2)	7.36 (9)	4.45 (3)	4.45 (3)	7 (8)	6.09 (7)	5.27 (6)	4.45 (3)

achieve superior performance than most comparing methods. That demonstrates the effectiveness of the proposed ADGD method.

- ADGD consistently achieves low hamming and ranking loss while maintaining high average precision across various datasets. This indicates the method's exceptional ability to not only accurately identify the most relevant labels for each instance but also to rank these labels effectively according to their relevance, which is crucial for applications that rely on the precision of label predictions, such as content recommendation and information retrieval systems.

- From the four tables, the performance of all comparing methods and ADGD gradually drop as the levels of noisy label increase, but the downtrend of ADGD is more gentle than other comparing methods. It well demonstrates the robustness of the proposed ADGD method.

- In order to further analyze the performance of all comparing algorithms varying with the number of selected features, the corresponding experimental results are shown in Figure 6.2. With the increasing number of selected features, the performance of ADGD first has a remarkable improvement and then keeps stable or even degrades. This observation reveals that it is meaningful to conduct feature selection for partial multi-label learning. In addition, MDFS as an effective feature selection method performs inferior to other methods since it cannot handle the noisy labels in practice. PMLFS is better than others but its performance is limited by the ignoring of the local and global structure of data and labels.

- In order to further analyze the performance of all comparing algorithms varying with noisy features, the corresponding experimental results are shown in Figure 6.3. We can find that ADGD significantly outperforms these comparing methods since most PML and MLL methods ignore the case with serious noises in feature space. Thus, we can conclude that the proposed method benefits the performance with feature selection and robustness of $L_{2,1}$-norm loss function.

6.4.4 Sensitivity Analysis

At last, we study the influences of the four parameters, λ, λ_1, λ_2, and λ_3 for the proposed method on the Mirflickr dataset. Our experiment is accomplished by using the

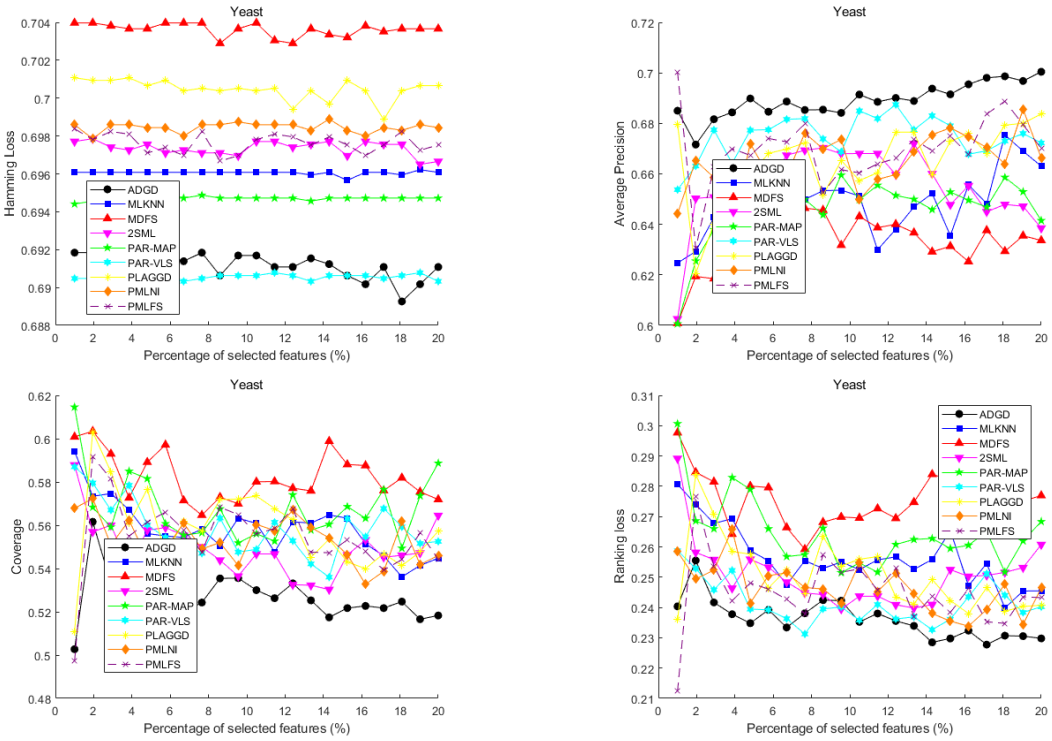


Figure 6.2: Experimental results on data set Yeast with 200% noisy labels.

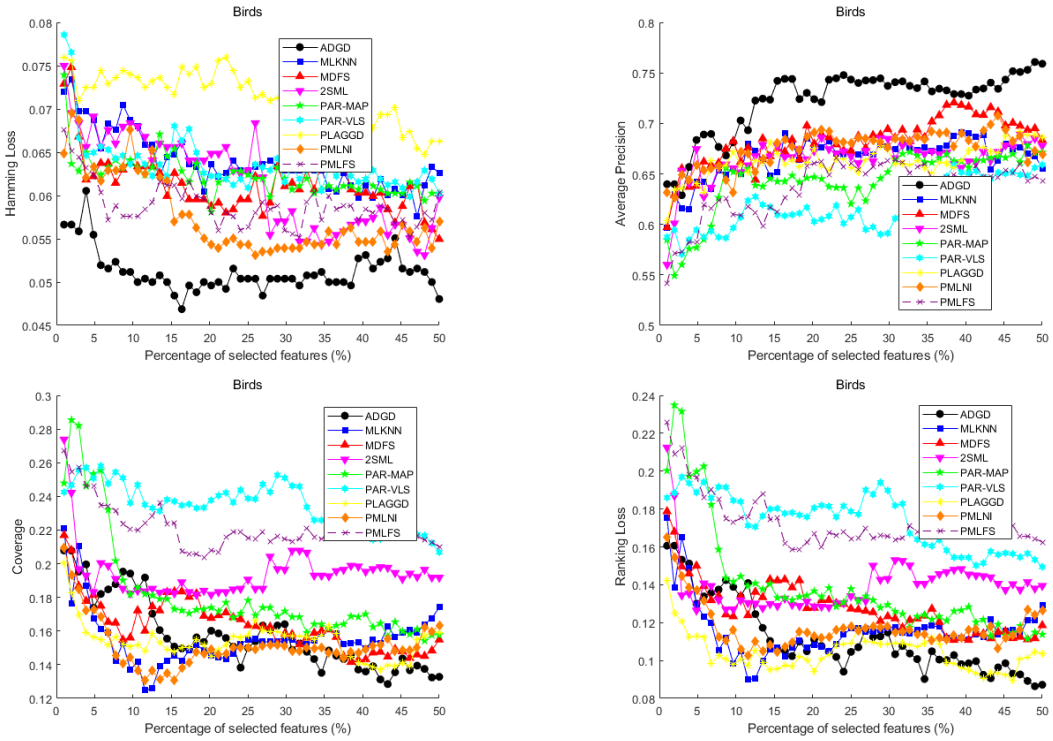
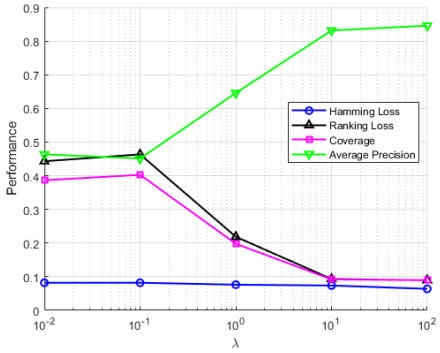
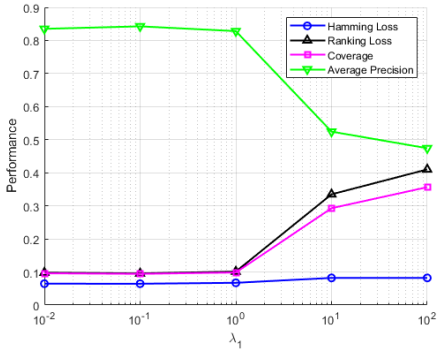


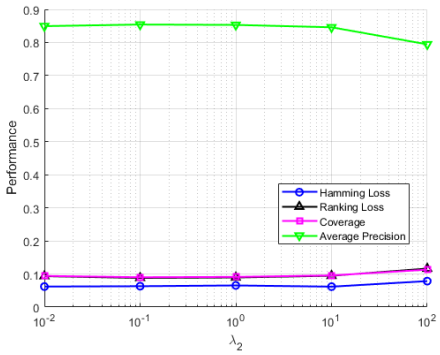
Figure 6.3: Experimental results on data set Birds with 10% feature noises.



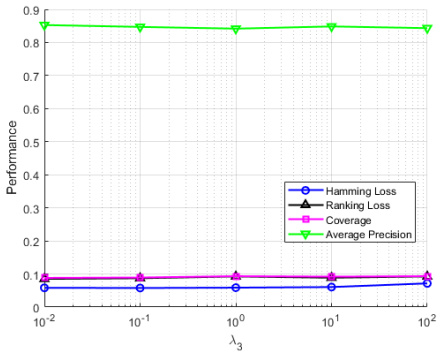
(a) performance curve with λ changes



(b) performance curve with λ_1 changes



(c) performance curve with λ_2 changes



(d) performance curve with λ_3 changes

Figure 6.4: Results of ADGD with varying value of trade-off parameters on Mirflickr.

grid search method which conducts the parameter analysis by varying four parameters simultaneously. The experimental results are shown in Figure 6.4 which are measured by the four evaluation metrics. It can be seen that how the performance of our algorithm varies as these parameters change. Therefore we should safely set them in a wide range in practice. From this figure, we can notice that better performances are gained when $\lambda = 100$, $\lambda_1 = 0.1$, $\lambda_2 = 0.1$ and $\lambda_3 = 0.01$.

6.4.5 Convergence and Complexity Analysis

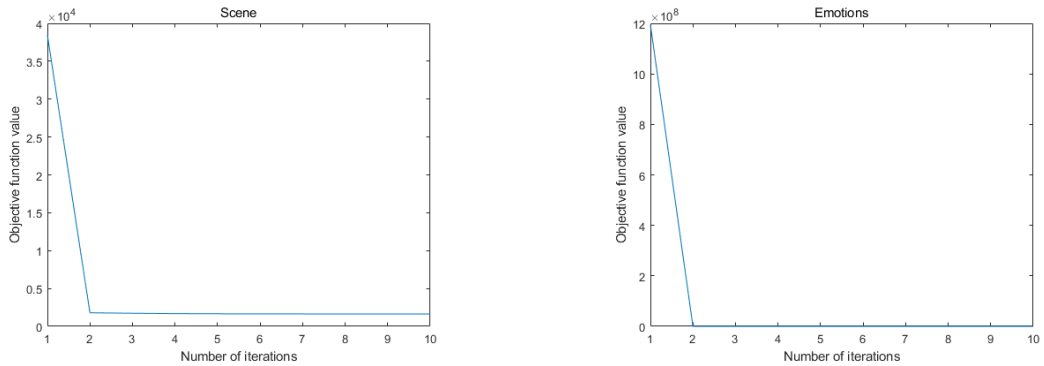


Figure 6.5: convergence curve on scene and emotions datasets with 100% label noises.

In the previous section, we have proved that the proposed ADGD algorithm will converge. In this subsection, we conduct experiments on the Scene and Emotions datasets to demonstrate this. The convergence curves of ADGD are shown in Figure 6.5, where we see that ADGD converges within 10 iterations on the two datasets.

6.4.6 Ablation Study

In this subsection, the ablation study on several variants of the proposed method is further conducted to analyze the contributions of its essential components. We choose the Entertainment dataset with 10% feature noises and 200% label noises to verify the proposed ADGD method and its variant methods.

Impact of learning label correlations. ADGD learns dual graphs to exploit both instance and label correlations to improve the partial multi-label feature selection. To demonstrate the effectiveness of using the single graph of label correlations, we compare ADGD with ADGD-NoLabelCorrelations, a variant of ADGD that does not

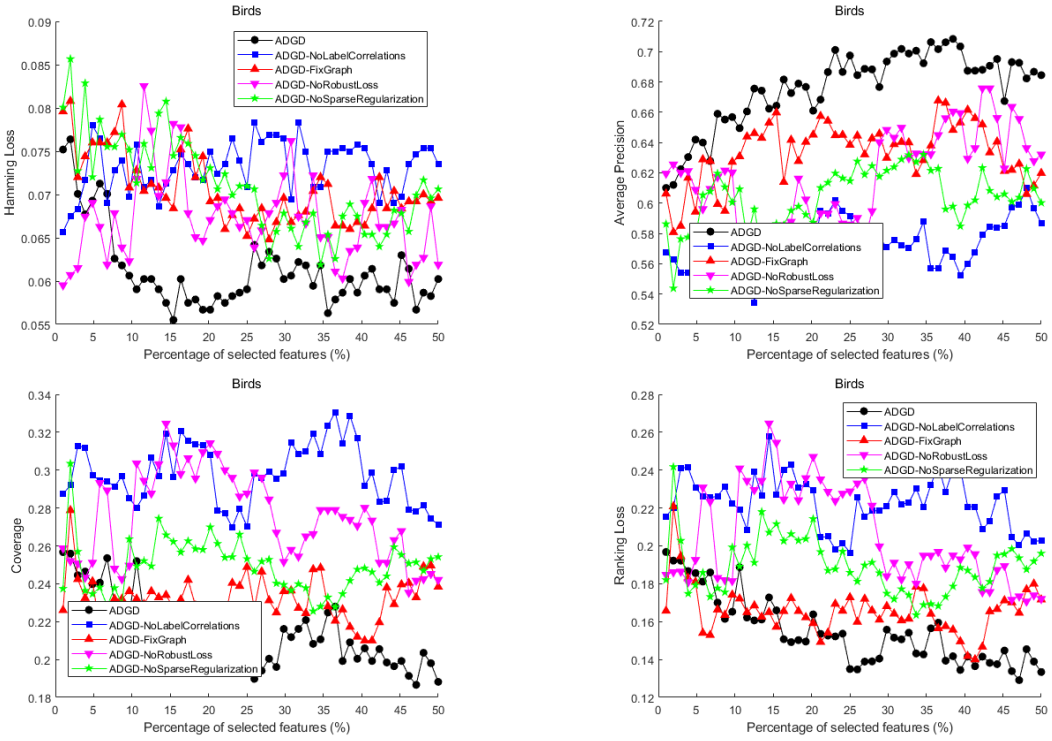


Figure 6.6: ADGD and its variant methods on Birds dataset with 20% feature noises and 200% label noises.

use label correlations (i.e., λ_3 is set to zero). As shown in Figure 10, we can see that ADGD outperforms ADGD-NoLabelCorrelations, showing that exploiting label correlations can improve our model.

Impact of learning adaptive dual graphs. To show the impact of adaptive dual graphs estimation, we compare ADGD with ADGD-FixGraph, a variant of ADGD that utilizes pre-defined instance graph S^X and label correlations graph S^Y indicated by heat-kernel function. As shown in Figure 10, ADGD outperforms ADGD-FixGraph, showing that the dual adaptive graphs based on both instance and label space information are better than the dual fixed similarity graphs and can help label disambiguation.

Impact of robust loss. To prove the effectiveness of $L_{2,1}$-norm imposed on regression loss terms, a variant of ADGD named ADGD-NoRobustNorm is proposed by imposing Frobenius norm on both $(X^TW - F)$ and $(F - Y)$ loss terms. As shown in Figure 10, ADGD outperforms ADGD-NoSparseRegularization since $L_{2,1}$-norm on loss terms can effectively improve the robustness of noisy features.

Impact of $L_{2,1}$-norm sparse regularization. To demonstrate the impact of the $L_{2,1}$-norm sparse regularization, we compare ADGD with ADGD-NoSparseRegularization, a variant of ADGD that does not use sparse regularization (i.e., λ_1 is set to zero). As shown in Figure 10, ADGD outperforms ADGD-NoSparseRegularization, showing the effectiveness of $L_{2,1}$-norm sparse regularization.

6.5 Conclusion

In this section, we introduce Adaptive Dual Graph Disambiguation (ADGD) by adaptively employing both instance-space graph regularization and label-space graph regularization to preserve the geometric structure of features and labels. This innovative framework unifies adaptive dual graph learning, candidate label disambiguation, and predictive model induction within a single objective function. Moreover, by applying the $L_{2,1}$-norm to both regression and regularization terms, ADGD not only improves resilience against data noise but also reduces the feature and label space dimensions. Experiments are conducted on datasets with both feature noise and label noise, which demonstrate the advantages of the proposed method ADGD compared with multi-label, partial label, and partial multi-label methods.

7

Conclusion and Future Work

7.1 Summary of the Thesis

This thesis addresses the critical need for accurate and reliable machine learning (ML) models in application areas where errors can have severe consequences, such as healthcare, finance, and autonomous vehicles. The focus is on overcoming challenges posed by imperfect data, which is often compromised by factors like insufficient information, data bias, label noise, and vulnerability to attacks.

The research is structured into three distinct parts, each targeting specific imperfections in real-world data:

- Part I - Semi-Supervised Learning (SSL): This section tackles the challenge of limited labeled data and data noise in SSL. The proposed Robust Embedding Regression (RER) method introduces a robust graph construction that adaptively adjusts weights for each data point, reducing the influence of noise. A low-rank representation enhances the utilization of limited labeled data, and appropriate norms in reconstruction and regularization terms aid in feature and sample selection. RER demonstrates a significant improvement in classification accuracy in noisy datasets.

- Part II - Transfer Learning (TL): The focus is on addressing domain shift and its resulting data imperfections. The Redirected Transfer Learning (RTL) approach mitigates the impact of domain shift by reconstructing target samples using low-rank representation from source samples. RTL employs $L_{2,1}$-norm sparsity and

a redirected label strategy for robust adaptation to diverse data distributions, showing substantial improvements in cross-domain classification tasks.

- Part III - Learning with Label Noise: This part addresses Partial Multi-Label Learning (PML) challenges, where incomplete labeling hinders accurate model building. Traditional methods are limited in adaptability and effectiveness under label uncertainty. The proposed Adaptive Dual Graph Disambiguation (ADGD) framework learns dual adaptive graphs and a sparse projection matrix, improving label confidence and reducing ambiguity in PML. The integration of $L_{2,1}$-norm enhances model robustness against feature noise.

The methodologies developed in this thesis contribute to the precision and generalizability of ML models, demonstrating notable improvements over existing methods in handling real-world data imperfections. The results of extensive experiments validate the effectiveness of the proposed approaches, marking a significant advancement in the field of machine learning under imperfect data conditions.

7.2 Future Work

In future work, we plan to expand the main three techniques employed in this thesis, including robust norm selection, low-rank representation, and adaptive graph construction, to more real-world applications and design more robust and reliable models.

- Robust Norm Selection: This approach is pivotal for feature selection in deep learning, particularly in bioinformatics and image processing. By emphasizing joint $L_{2,1}$-norm minimization, it offers robustness against outliers and noise, which is crucial for accurate feature extraction in high-dimensional data like genomic sequences or medical images.

- Low-Rank Representation: In deep learning models like convolutional neural networks, low-rank representation can be used to enhance the efficiency and interpretability of the model. It reduces the complexity of the model while maintaining performance, which is particularly useful in applications like image and video processing, where reducing computational load without sacrificing accuracy is key.

- Adaptive Graph Construction: In the context of graph neural networks, adaptive graph construction allows for dynamic learning of graph structures, enhancing the model's ability to handle complex, evolving data. This is particularly relevant in social network analysis, recommendation systems, and any other domain where the underlying data relationships are non-static and intricate.

References

[1] Ahmed, Z., Mohamed, K., Zeeshan, S. & Dong, X. (2020). Artificial intelligence with multi-functional machine learning platform development for better healthcare and precision medicine. *Database*, **2020**, baaa010. 1

[2] Bao, J., Kudo, M., Kimura, K. & Sun, L. (2024). Redirected transfer learning for robust multi-layer subspace learning. *Pattern Analysis and Application (In print)*. 17, 55

[3] Bao, J., Kudo, M., Kimura, K. & Sun, L. (2024). Robust embedding regression for semi-supervised learning. *Pattern Recognition*, **145**, 109894. 17, 31

[4] Belkin, M., Niyogi, P. & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, **7**, 2399–2434. 9, 44

[5] Bharilya, V. & Kumar, N. (2023). Machine learning for autonomous vehicle's trajectory prediction: A comprehensive survey, challenges, and future research directions. 1

[6] Blum, A. & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, 92–100, Association for Computing Machinery, New York, NY, USA. 1

[7] Boyd, S., Parikh, N., Chu, E., Peleato, B. & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, **3**, 1–122. 58, 62

[8] Branicky, M. (1998). Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, **43**, 475–482. 84

[9] Cai, J.F., Candes, E.J. & Shen, Z. (2008). A singular value thresholding algorithm for matrix completion. 39, 60

[10] Chen, X., Yuan, G., Nie, F. & Ming, Z. (2020). Semi-supervised feature selection via sparse rescaled linear square regression. *IEEE Transactions on Knowledge and Data Engineering*, **32**, 165–176. 9, 44

[11] Chen, Z.S., Wu, X., Chen, Q.G., Hu, Y. & Zhang, M.L. (2020). Multi-view partial multi-label learning with graph-based disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence*, **34**, 3553–3560. 27

[12] Cour, T., Sapp, B. & Taskar, B. (2011). Learning from partial labels. *J. Mach. Learn. Res.*, **12**, 1501–1536. 14

[13] De la Torre, F. (2012). A least-squares framework for component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**, 1041–1055. 8

[14] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, **7**, 1–30. 90

[15] Deng, W., Liao, Q., Zhao, L., Guo, D., Kuang, G., Hu, D. & Liu, L. (2021). Joint clustering and discriminative feature alignment for unsupervised domain adaptation. *IEEE Transactions on Image Processing*, **30**, 7842–7855. 11

[16] Eckstein, J. & Bertsekas, D. (1992). On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, **55**, 293–318. 62

[17] Feng, L. & An, B. (2018). Leveraging latent label distributions for partial label learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 2107–2113, International Joint Conferences on Artificial Intelligence Organization. 27

[18] Frenay, B. & Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, **25**, 845–869. 3

[19] Georghiades, A., Belhumeur, P. & Kriegman, D. (2001). From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**, 643–660. 43

[20] Gong, B., Shi, Y., Sha, F. & Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2066–2073. 65

[21] Guo, W., Wang, Z. & Du, W. (2023). Robust semi-supervised multi-view graph learning with sharable and individual structure. *Pattern Recognition*, **140**, 109565. 10

[22] Han, N., Wu, J., Fang, X., Xie, S., Zhan, S., Xie, K. & Li, X. (2020). Latent elastic-net transfer learning. *IEEE Transactions on Image Processing*, **29**, 2820–2833. 12, 26, 62, 64

[23] Hao, P., Hu, L. & Gao, W. (2023). Partial multi-label feature selection via subspace optimization. *Information Sciences*, **648**, 119556. 16

[24] Hu, L., Li, Y., Gao, W., Zhang, P. & Hu, J. (2020). Multi-label feature selection with shared common mode. *Pattern Recognition*, **104**, 107344. 16

[25] Hu, Y., Zhang, D., Ye, J., Li, X. & He, X. (2013). Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**, 2117–2130. 26

[26] Huang, D., Cabral, R. & Torre, F.D.l. (2016). Robust regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**, 363–375. 8

[27] Huang, G.B., Mattar, M., Berg, T. & Learned-Miller, E. (2008). Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments. In *Workshop on Faces in 'Real-Life' Images: Detec-*

tion, Alignment, and Recognition, Erik Learned-Miller and Andras Ferencz and Frédéric Jurie, Marseille, France. 43

[28] Huang, J., Li, G., Huang, Q. & Wu, X. (2015). Learning label specific features for multi-label classification. In *2015 IEEE International Conference on Data Mining*, 181–190. 84

[29] Jhuo, I.H., Liu, D., Lee, D.T. & Chang, S.F. (2012). Robust visual domain adaptation with low-rank reconstruction. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2168–2175. 55

[30] Kan, M., Wu, J., Shan, S. & Chen, X. (2013). Domain adaptation for face recognition: Targetize source domain bridged by common subspace. *International Journal of Computer Vision*, **109**, 94–109. 10

[31] Kang, Z., Pan, H., Hoi, S.C.H. & Xu, Z. (2020). Robust graph learning from noisy data. *IEEE Transactions on Cybernetics*, **50**, 1833–1843. 10, 44

[32] Krizhevsky, A., Sutskever, I. & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, 1097–1105, Curran Associates Inc., Red Hook, NY, USA. 64, 65

[33] Lei, W., Ma, Z., Lin, Y. & Gao, W. (2021). Domain adaption based on source dictionary regularized rkhs subspace learning. *Pattern Analysis and Applications*, **24**, 1513–1532. 11

[34] Li, C. & Tam, P. (2000). A global energy approach to facet model and its minimization using weighted least-squares algorithm. *Pattern Recognition*, **33**, 281–293. 8

[35] Li, Q., He, H., Lai, H., Cai, T., Wang, Q. & Gao, Q. (2022). Enhanced nuclear norm based matrix regression for occluded face recognition. *Pattern Recognition*, **126**, 108585. 10, 33

[36] Li, X. & Wang, Y. (2020). Recovering accurate labeling information from partially valid data for effective multi-label learning. 15

[37] Lin, Y., Hu, Q., Liu, J. & Duan, J. (2015). Multi-label feature selection based on max-dependency and min-redundancy. *Neurocomputing*, **168**, 92–103. 15

[38] Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y. & Ma, Y. (2013). Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**, 171–184. 51, 60

[39] Liu, H., Han, J. & Nie, F. (2017). Semi-supervised orthogonal graph embedding with recursive projections. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2308–2314. 9, 44, 45

[40] Liu, J., Lin, M., Mingbo, Z., Choujun, Z., Bing, L. & Tai, C.J.K. (2023). Person re-identification via semi-supervised adaptive graph embedding. *Applied Intelligence*, **53**, 2656–2672. 8

[41] Long, M., Wang, J., Ding, G., Sun, J. & Yu, P.S. (2013). Transfer feature learning with joint distribution adaptation. In *2013 IEEE International Conference on Computer Vision*, 2200–2207. 11, 64, 65

[42] Long, M., Wang, J., Sun, J. & Yu, P.S. (2015). Domain invariant transfer kernel learning. *IEEE Transactions on Knowledge and Data Engineering*, **27**, 1519–1532. 65

[43] Lu, Y., Lai, Z., Wong, W.K. & Li, X. (2020). Low-rank discriminative regression learning for image classification. *Neural Networks*, **125**, 245–257. 10, 33, 35

[44] Lv, J., Feng, L., Xu, M., An, B., Niu, G., Geng, X. & Sugiyama, M. (2021). On the robustness of average losses for partial-label learning. *CoRR*, **abs/2106.06152**. 15

[45] Lyu, G., Feng, S. & Li, Y. (2020). Partial multi-label learning via probabilistic graph matching mechanism. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, 105–113, Association for Computing Machinery, New York, NY, USA. 15

[46] Ma, X., Zhang, T. & Xu, C. (2019). Gcan: Graph convolutional adversarial network for unsupervised domain adaptation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8258–8268. 67

[47] Mashrur, A., Luo, W., Zaidi, N.A. & Robles-Kelly, A. (2020). Machine learning for financial risk management: A survey. *IEEE Access*, **8**, 203203–203223. 1

[48] Nie, F., Huang, H., Cai, X. & Ding, C. (2010). Efficient and robust feature selection via joint $l_{2,1}$-norms minimization. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, NIPS'10, 1813–1821, Curran Associates Inc., Red Hook, NY, USA. 8, 22

[49] Nie, F., Xu, D., Tsang, I.W.H. & Zhang, C. (2010). Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE Transactions on Image Processing*, **19**, 1921–1932. 9, 23, 42, 44

[50] Nie, F., Shi, S. & Li, X. (2020). Semi-supervised learning with auto-weighting feature and adaptive graph. *IEEE Transactions on Knowledge and Data Engineering*, **32**, 1167–1178. 9, 42, 44

[51] Nie, F., Wang, Z., Wang, R. & Li, X. (2022). Adaptive local embedding learning for semi-supervised dimensionality reduction. *IEEE Transactions on Knowledge and Data Engineering*, **34**, 4609–4621. 9, 43, 44

[52] Pan, S.J. & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, **22**, 1345–1359. 3, 10

[53] Pan, S.J., Tsang, I.W., Kwok, J.T. & Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, **22**, 199–210. 11, 64

[54] Peng, Z., Zhang, W., Han, N., Fang, X., Kang, P. & Teng, L. (2020). Active transfer learning. *IEEE Transactions on Circuits and Systems for Video Technology*, **30**, 1022–1036. 24

[55] Prabono, A.G., Yahya, B.N. & Lee, S.L. (2021). Hybrid domain adaptation for sensor-based human activity recognition in a heterogeneous setup with feature commonalities. *Pattern Anal. Appl.*, **24**, 1501–1511. 11

[56] Qian, K., Min, X.Y., Cheng, Y. & Min, F. (2023). Weight matrix sharing for multi-label learning. *Pattern Recognition*, **136**, 109156. 14, 89

[57] Qiu, S., Nie, F., Xu, X., Qing, C. & Xu, D. (2019). Accelerating flexible manifold embedding for scalable semi-supervised learning. *IEEE Transactions on Circuits and Systems for Video Technology*, **29**, 2786–2795. 9, 44

[58] Read, J., Pfahringer, B., Holmes, G. & Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, **85**, 333–359. 14

[59] Shao, M., Castillo, C., Gu, Z. & Fu, Y. (2012). Low-rank transfer subspace learning. In *2013 IEEE 13th International Conference on Data Mining*, 1104–1109, IEEE Computer Society, Los Alamitos, CA, USA. 12, 25

[60] Si, S., Tao, D. & Geng, B. (2010). Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering*, **22**, 929–942. 11, 55

[61] Si, Y., Pu, J., Zang, S. & Sun, L. (2021). Extreme learning machine based on maximum weighted mean discrepancy for unsupervised domain adaptation. *IEEE Access*, **9**, 2283–2293. 11

[62] Silva, I.O.e., Soares, C., Sousa, I. & Ghani, R. (2024). Systematic analysis of the impact of label noise correction on ml fairness. In T. Liu, G. Webb, L. Yue & D. Wang, eds., *AI 2023: Advances in Artificial Intelligence*, 173–184, Springer Nature Singapore, Singapore. 3

[63] Sim, T., Baker, S. & Bsat, M. (2003). The cmu pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**, 1615 – 1618. 43

[64] Song, Z., Yang, X., Xu, Z. & King, I. (2023). Graph-based semi-supervised learning: A comprehensive review. *IEEE Transactions on Neural Networks and Learning Systems*, **34**, 8174–8194. 1

[65] Sun, L., Feng, S., Wang, T., Lang, C. & Jin, Y. (2019). Partial multi-label learning by low-rank and sparse decomposition. *Proceedings of the AAAI Conference on Artificial Intelligence*, **33**, 5016–5023. 15

[66] Sun, L., Feng, S., Liu, J., Lyu, G. & Lang, C. (2022). Global-local label correlation for partial multi-label learning. *IEEE Transactions on Multimedia*, **24**, 581–593. 15

[67] van der Maaten, L. & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, **9**, 2579–2605. 72

[68] Wan, J., Chen, Y. & Bai, B. (2021). Joint feature extraction and classification in a unified framework for cost-sensitive face recognition. *Pattern Recognition*, **115**, 107927. 8

[69] Wan, M., Chen, X., Zhao, C., Zhan, T. & Yang, G. (2022). A new weakly supervised discrete discriminant hashing for robust data representation. *Information Sciences*, **611**, 335–348. 65

[70] Wan, M., Yao, Y., Zhan, T. & Yang, G. (2022). Supervised low-rank embedded regression (slrer) for robust subspace learning. *IEEE Transactions on Circuits and Systems for Video Technology*, **32**, 1917–1927. 12

[71] Wan, M., Chen, X., Zhan, T., Yang, G., Tan, H. & Zheng, H. (2023). Low-rank 2d local discriminant graph embedding for robust image feature extraction. *Pattern Recognition*, **133**, 109034. 10, 12, 23, 36, 41

[72] Wang, D.B., Zhang, M.L. & Li, L. (2022). Adaptive graph guided disambiguation for partial label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44**, 8796–8811. 29, 81, 90

[73] Wang, F. & Zhang, C. (2008). Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, **20**, 55–67. 9

[74] Wang, H., Liu, W., Zhao, Y., Zhang, C., Hu, T. & Chen, G. (2019). Discriminative and correlative partial multi-label learning. In *Proceedings of the*

Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, 3691–3697, International Joint Conferences on Artificial Intelligence Organization. 15, 27

[75] Wang, J., Chen, Y., Feng, W., Yu, H., Huang, M. & Yang, Q. (2020). Transfer learning with dynamic distribution adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, **11**, 1–25. 64

[76] Wang, J., Li, P. & Yu, K. (2022). Partial multi-label feature selection. In *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–9. 16, 90

[77] Wang, L., Ding, Z. & Fu, Y. (2018). Adaptive graph guided embedding for multi-label annotation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 2798–2804, International Joint Conferences on Artificial Intelligence Organization. 28

[78] Wang, X., Lu, S., Zhou, R. & Wang, H. (2023). Skeleton estimation of directed acyclic graphs using partial least squares from correlated data. *Pattern Recognition*, **139**, 109460. 8

[79] Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G. & Liu, Y. (2022). Learning with noisy labels revisited: A study using real-world human annotations. 3

[80] Wen, J., Fang, X., Xu, Y., Tian, C. & Fei, L. (2018). Low-rank representation with adaptive graph regularization. *Neural Networks*, **108**, 83–96. 10, 24

[81] Wen, J., Zhang, B., Xu, Y., Yang, J. & Han, N. (2018). Adaptive weighted nonnegative low-rank representation. *Pattern Recognition*, **81**, 326–340. 10, 42, 44

[82] Wen, Y., Zhang, K., Li, Z. & Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In B. Leibe, J. Matas, N. Sebe & M. Welling, eds., *Computer Vision – ECCV 2016*, 499–515, Springer International Publishing, Cham. 44

[83] Wu, S., Gao, G., Li, Z., Wu, F. & Jing, X.Y. (2020). Unsupervised visual domain adaptation via discriminative dictionary evolution. *Pattern Analysis and Applications*, **23**, 1665–1675. 11

[84] Wu, X.Z. & Zhou, Z.H. (2017). A unified view of multi-label performance measures. 88

[85] Wu, Z., Lv, J. & Sugiyama, M. (2022). Learning with proper partial labels. 15

[86] Xiang, S., Nie, F., Meng, G., Pan, C. & Zhang, C. (2012). Discriminative least squares regression for multiclass classification and feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, **23**, 1738–1754. 8, 12, 55

[87] Xie, M.K. & Huang, S.J. (2018). Partial multi-label learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, **32**. 15, 27

[88] Xie, M.K. & Huang, S.J. (2022). Partial multi-label learning with noisy label identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44**, 3676–3687. 15, 89

[89] Xu, Y., Fang, X., Wu, J., Li, X. & Zhang, D. (2016). Discriminative transfer subspace learning via low-rank and sparse representation. *IEEE Transactions on Image Processing*, **25**, 850–863. 12, 26, 62, 64

[90] Yin, M., Gao, J. & Lin, Z. (2016). Laplacian regularized low-rank representation and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**, 504–517. 24

[91] You, C.Z., Wu, X.J. & Palade, V. (2018). Graph regularized low-rank representation for semi-supervised learning. In *2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, 92–95. 10

[92] Zhang, J., Li, W. & Ogunbona, P. (2017). Joint geometrical and statistical alignment for visual domain adaptation. 11, 64

[93] Zhang, J., Luo, Z., Li, C., Zhou, C. & Li, S. (2019). Manifold regularized discriminative feature selection for multi-label learning. *Pattern Recognition*, **95**, 136–150. 14, 16, 89

[94] Zhang, J., Lin, Y., Jiang, M., Li, S., Tang, Y. & Tan, K.C. (2020). Multi-label feature selection via global relevance and redundancy optimization. In C. Bessiere, ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 2512–2518, International Joint Conferences on Artificial Intelligence Organization, main track. 88

[95] Zhang, L., Wang, S., Huang, G.B., Zuo, W., Yang, J. & Zhang, D. (2019). Manifold criterion guided transfer learning via intermediate domain generation. *IEEE Transactions on Neural Networks and Learning Systems*, **30**, 3759–3773. 11

[96] Zhang, L., Fu, J., Wang, S., Zhang, D., Dong, Z. & Chen, C.L.P. (2020). Guide subspace learning for unsupervised domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, **31**, 3374–3388. 12, 26, 64

[97] Zhang, M.L. & Fang, J.P. (2021). Partial multi-label learning via credible label elicitation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **43**, 3587–3599. 15, 27, 89

[98] Zhang, M.L. & Zhou, Z.H. (2007). Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, **40**, 2038–2048. 14, 89

[99] Zhang, M.L. & Zhou, Z.H. (2014). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, **26**, 1819–1837. 87

[100] Zhang, M.L., Zhou, B.B. & Liu, X.Y. (2016). Partial label learning via feature-aware disambiguation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 1335–1344, Association for Computing Machinery, New York, NY, USA. 15

[101] Zhang, M.L., Yu, F. & Tang, C.Z. (2017). Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, **29**, 2155–2167. 15

[102] Zhang, M.L., Li, Y.K., Liu, X.Y. *et al.* (2018). Binary relevance for multi-label learning: an overview. *Frontiers of Computer Science*, **12**, 191–202. 14

[103] Zhang, P., Liu, G. & Gao, W. (2019). Distinguishing two types of labels for multilabel feature selection. *Pattern Recognition*, **95**, 72–82. 15

[104] Zhang, X.Y., Wang, L., Xiang, S. & Liu, C.L. (2015). Retargeted least squares regression algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, **26**, 2206–2213. 27, 55, 61

[105] Zhang, Y., Ye, H. & Davison, B.D. (2021). Adversarial reinforcement learning for unsupervised domain adaptation. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 635–644. 11

[106] Zhang, Z., Lai, Z., Xu, Y., Shao, L., Wu, J. & Xie, G.S. (2017). Discriminative elastic-net regularized linear regression. *IEEE Transactions on Image Processing*, **26**, 1466–1481. 26

[107] Zhao, M., Liu, J., Zhang, Z. & Fan, J. (2021). A scalable sub-graph regularization for efficient content based image retrieval with long-term relevance feedback enhancement. *Knowledge-Based Systems*, **212**, 106505. 8

[108] Zhou, D., Bousquet, O., Lal, T., Weston, J. & Schölkopf, B. (2004). Learning with local and global consistency. *Advances in neural information processing systems*, **16**, 321–328. 9

[109] Zhu, P., Xu, Q., Hu, Q., Zhang, C. & Zhao, H. (2018). Multi-label feature selection with missing labels. *Pattern Recognition*, **74**, 488–502. 16

[110] Zhu, R., Dornaika, F. & Ruichek, Y. (2020). Semi-supervised elastic manifold embedding with deep learning architecture. *Pattern Recognition*, **107**, 107425. 9

[111] Zhu, X. & Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. 1

[112] Zhu, X., Ghahramani, Z. & Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, 912–919, AAAI Press. 9, 22

[113] Zou, H., Hastie, T. & Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, **15**, 265–286. 59