UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Transfer Learning with Imprecise Observations: Theory and Algorithms

by Guangzhi Ma

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy in Computer Science

under the supervision of Distinguished Professor Jie Lu, A/Prof Guangquan Zhang and Dr Feng Liu

University of Technology Sydney
Faculty of Engineering and Information Technology

August 2024

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Guangzhi Ma*, declare that this thesis is submitted in fulfilment of the requirements for the award of *Doctor of Philosophy*, in the *School of Computer Science*, *Faculty of Engineering and Information Technology* at the University of Technology Sydney. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note: Signature removed prior to publication.

[Guangzhi Ma]

DATE: 15th August, 2024

PLACE: Sydney, Australia

Abstract

Transfer learning aims to leverage previously-acquired knowledge from domains with abundant labels (i.e., source domains) to help train a classifier or predictor for the domain with insufficient labels (i.e., target domain). Although transfer learning has achieved significant advancements in many areas, most existing methods share a common assumption that the observations in the source and target domains are precise. Unfortunately, precise observations are often unavailable in real-world scenarios. For example, the readings on many measuring devices are not exact numbers but intervals, as there are only a limited number of decimals available on most measuring devices.

In this research, we consider a new, realistic problem called *transfer learning with imprecise observations* (TLIMO), where the source or target domains only contain imprecise observations. To develop new theories and construct algorithms for addressing TLIMO problem in various real-world scenarios, this thesis intends to address four orthogonal problems: 1) How to construct a theoretical foundation for imprecise data analysis and handle a simple problem called *multi-class classification with imprecise observations* (MCIMO); 2) How to handle TLIMO problem in single-source domain scenario; 3) How to handle the multi-source transfer learning problem when the instances in the source or target domains are imprecise; and 4) How to handle the universal domain adaptation (UniDA) problem when the instances in the source or target domains are imprecise.

To address Problem 1), this thesis develops a theoretical foundation for imprecise

data analysis based on fuzzy random variables and provides a theoretical analysis of MCIMO problem (Chapter 3). This theoretical analysis ensures that we can always train a fuzzy classifier with high classification accuracy when infinite imprecise instances can be collected. Two new frameworks are constructed for addressing MCIMO problem. The first integrates defuzzification methods with support vector machines and neural networks (Chapter 3), while the second applies multi-view learning and fuzzy techniques to analyze interval-valued data (Chapter 4).

To address Problem 2), we extend the theoretical analysis of MCIMO to develop theory for the TLIMO problem. This theory derives a generalization bound to guide model construction. A novel transfer learning approach is then proposed to transfer knowledge from a single-source domain to a single-target domain with imprecise data (Chapter 5).

To address Problem 3), Chapter 6 presents two domain adaptation models to transfer knowledge from multiple source domains with crisp-valued data to a single-target domain with imprecise data. The first model designs a fuzzy relation-based approach to appropriately combine multiple classifiers trained on multiple source domains for enhanced adaptation performance, while the second model uses a fuzzy distance-based approach to achieve the same purpose.

In Chapter 7, we develop a novel dynamic reweighted loss learning strategy to tackle an unsolved problem in transfer learning, where the distribution discrepancy between the source domain and the target domain in different categories may be significantly different. Then, we can apply this proposed strategy and our previous designed model via fuzzy techniques to address Problem 4).

In summary, this thesis not only contributes to the theory of transfer learning when the source domain or target domain contains imprecise observations but also proposes a set of effective algorithms for different transfer learning scenarios.

Dedication

To myself . . .

ACKNOWLEDGMENTS

I am deeply grateful to all those who have supported me throughout my PhD journey at the University of Technology Sydney. Although these three and a half years of pursuing a PhD abroad have been very challenging, they have undoubtedly been exciting and memorable.

First and foremost, I would like to express my heartfelt gratitude to my principal supervisor, Distinguished Professor Jie Lu. From the very beginning of my PhD study, she guided me step-by-step on how to complete my PhD program diligently. During times of difficulty, she always believed in my research abilities and provided me with all the necessary support unconditionally in the past three and half years. Without her dedication and encouragement, I would not have been able to smoothly complete my PhD project. Additionally, she offered professional and visionary suggestions for my future research path. What she taught me and what I learned from her in the past three and half years will benefit me for a lifetime.

I also extend my sincere thanks to my co-supervisors, Associate Professor Guangquan Zhang and Dr. Feng Liu. They provided valuable suggestions for my research direction and patiently guided me from reading literature to finally completing a qualified paper during the early stages of my PhD study. Their excellent advice was indispensable for all the papers I completed during my PhD period. Moreover, their passion for research and rigorous attitude have greatly inspired me.

I am also grateful to the University of Technology Sydney and Australian Research

Council (grand under FL190100149), who provided financial support for my PhD research. This financial support basically covered my living expenses during my three and a half years of studying in Sydney.

I would like to express my gratitude to every member of the Decision Systems & e-Service Intelligence Lab (DeSI) in the Australian Artificial Intelligence Institute (AAII). It was a wonderful experience to spend four years with these dedicated researchers. I especially thank Dr. Zhen Fang, Dr. Keqiuyin Li, Dr. Hua Zuo, Dr. Yi Zhang, and En Yu who provided insightful comments related to my research problem during my Ph.D. candidature; Dr. Yiliao Song, Dr. Bin Zhang, Dr. Kun Wang, Ming Zhou, Wei Duan, Zihe Liu, Xinheng Wu and Dr. Mengjia Wu who provide great help and support in my overseas life, and Ming Zhou, Wei Duan and Zhaoqing Liu who shared my joys and sadness.

Meanwhile, I genuinely thank Jemima Moore, Sue Felix and Michele Mooney for language proofreading of all my publications. Their meticulous and rigorous work attitude has improved the quality of my publications.

Lastly, I want to express my heartfelt appreciation and gratitude to my parents, girlfriend, and family members for their unwavering love and support.

List of Publications

1. **Guangzhi Ma**, Jie Lu, Zhen Fang, Feng Liu, Guangquan Zhang, Multi-view Classification through Learning from Interval-valued Data, *IEEE Transactions on Neural Networks and Learning Systems* (IEEE-TNNLS), 2024. DOI: 10.1109/TNNLS.2024.3421657. [ERA&CORE: **A***, JCR Q1]

2. **Guangzhi Ma**, Jie Lu, Feng Liu, Zhen Fang, Guangquan Zhang, Domain Adaptation With Interval-Valued Observations: Theory and Algorithms, *IEEE Transactions on Fuzzy Systems* (IEEE-TFS), Vol. 32, no. 5, pp. 3107-3120, 2024. DOI: 10.1109/TFUZZ.2024.3367460. [ERA&CORE: **A***, JCR Q1]

3. **Guangzhi Ma**, Jie Lu, Guangquan Zhang, Multisource Domain Adaptation With Interval-Valued Target Data via Fuzzy Neural Networks, *IEEE Transactions on Fuzzy Systems* (IEEE-TFS), Vol. 32, no. 5, pp. 3094-3106, 2024. DOI: 10.1109/TFUZZ.2024.3367456. [ERA&CORE: **A***, JCR Q1]

4. Jie Lu, **Guangzhi Ma**, Guangquan Zhang, Fuzzy Machine Learning: A Comprehensive Framework and Systematic Review, *IEEE Transactions on Fuzzy Systems* (IEEE-TFS), Vol. 32, no. 7, pp. 3861-3878, 2024. DOI: 10.1109/TFUZZ.2024.3387429. [ERA&CORE: **A***, JCR Q1]

5. **Guangzhi Ma**, Jie Lu, Guangquan Zhang, Interval-Valued Observations-Based Multi-Source Domain Adaptation Using Fuzzy Neural Networks, *Proceedings of*

the 2023 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2023), pp. 1-6, 2023. [ERA&CORE: A]

6. **Guangzhi Ma**, Jie Lu, Feng Liu, Zhen Fang, Guangquan Zhang, Multiclass Classification With Fuzzy-Feature Observations: Theory and Algorithms, *IEEE Transactions on Cybernetics* (IEEE-TCYB), Vol. 54, no. 2, pp. 1048-1061, 2022. DOI: 10.1109/TCYB.2022.3181193. [ERA&CORE: A, JCR Q1]

7. **Guangzhi Ma**, Feng Liu, Guangquan Zhang, Jie Lu, Learning from imprecise observations: An estimation error bound based on fuzzy random variables, *Proceedings of the 2021 IEEE International Conference on Fuzzy Systems* (FUZZ-IEEE 2021), pp. 1-8, 2021. [ERA&CORE: A]

8. **Guangzhi Ma**, Jie Lu, Feng Liu, Zhen Fang, Guangquan Zhang, Distraction-control for Universal Domain Adaptation, *IEEE Transactions on Neural Networks and Learning Systems* (IEEE-TNNLS). [ERA: A, CORE: **A***, JCR Q1] (submitted)

Table of Contents

List of Figures

Figure **Page**

List of Tables

Table **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

In the dynamic realm of technology, machine learning has profoundly transformed various sectors. It drives innovation by decoding complex data patterns, advancing artificial intelligence, and influencing how we engage with information and understand the capabilities of computational systems. However, most well-known machine learning approaches, such as *support vector machines* (SVM) [18] and *neural networks* [153], operate under a common assumption: the training data (source domain) and test data (target domain) are drawn from identical feature spaces and identical distributions. Therefore, when the target domain has a different feature space or distribution from the source domain, models constructed in the source domain cannot be directly applied; they must be rebuilt and retrained from scratch using newly gathered instances in the target domain. Unfortunately, in many real-world scenarios, gathering sufficient labeled instances to construct a learning-based model for the target domain is difficult, time-consuming, and sometimes even impossible. As a result, researchers have considered

transferring and utilizing knowledge from the source domain to guide the construction of the model in the target domain, a process referred to as transfer learning.

The mechanism of transfer learning [112] has garnered significant attention in various fields, including computer vision [45, 86], biology [35, 143], and finance [46, 61]. Transfer learning can be categorized into several types: multitask learning [92], domain adaptation [183], cross-domain adaptation [166], and heterogeneous learning [178, 180]. Unlike traditional machine learning algorithms, transfer learning addresses scenarios where the domains, tasks, and distributions of the training and test data may differ. This situation is quite common in the real world; for instance, learning English vocabulary often aids in learning French vocabulary, or learning to play the guitar facilitates learning the violin. The concept of transfer learning, inspired by the notion that utilizing previously acquired knowledge in the source domain to solve new but similar problems in the target domain can enhance both efficiency and accuracy, is thus introduced.

Recently, the integration of fuzzy techniques with transfer learning methods has garnered increasing attention. Behbood *et al.* [5] proposed a novel fuzzy-based transfer learning method for long-term bank failure prediction. Deng *et al.* [35, 63, 64, 161, 164] introduced a series of new transfer learning approaches that integrate the Takagi-Sugeno-Kang fuzzy system with transfer learning to recognize epileptic electroencephalogram signals. To address heterogeneous unsupervised domain adaptation problems in classification tasks, Liu *et al.* [87, 90] developed two new transfer learning approaches utilizing shared fuzzy equivalence relations via fuzzy geometry. Furthermore, for regression tasks, [186] presented an innovative fuzzy rule-based transfer learning model that combines an infinite Gaussian mixture model with active learning. Lu *et al.* [97] proposed a novel fuzzy rule-based transfer learning approach that merges fuzzy rules from multi-source domains in both homogeneous and heterogeneous scenarios. These works illustrate that integrating fuzzy techniques with existing transfer learning algorithms can effectively

address various types of uncertainty issues.

1.2 Motivation

Most existing transfer learning works predominantly concentrate on addressing large-scale image data characterized by crisp-valued features. However, in real-world applications, datasets often encapsulate uncertainties and imprecisions that cannot be adequately represented by single-point values. For instance, interval-valued observations, which express a range or uncertainty associated with each data point, offer a more faithful representation of such inherent uncertainties. Consider medical data where patient health parameters fluctuate within a certain range, or environmental monitoring data capturing fluctuating sensor readings. In these contexts, interval-valued observations become indispensable. Therefore, the utilization of interval-valued datasets not only aligns with the inherent nature of uncertainties present in many real-world scenarios but also facilitates more accurate and robust analyzes. Consequently, in this paper, we focus on a more realistic and challenging problem named *transfer learning with imprecise observations* (TLIMO). Within the TLIMO context, we confront the scenario of having a source domain enriched with an adequate quantity of labeled observations and an unlabeled target domain, where the instances from both the source or target domain are characterized by imprecise features.

To solve the TLIMO problem and move forward to more realistic scenarios, there are four orthogonal problems need to be solved. The orthogonal problems are: 1) how to construct a theoretical foundation for imprecise data analysis and address MCIMO problem; 2) how to handle single-source domain adaptation problem with imprecise observations; 3) how to handle the multi-source transfer learning problem when the instances in the source or target domains are imprecise; 4) how to tackle the *universal domain adaptation* (UniDA) problem when the instances in the source or target domains

are imprecise. This thesis provides a comprehensive analysis and solutions to all the aforementioned challenges.

1.3 Research Questions and Objectives

This research aims to develop a set of theory and methods towards transfer learning with imprecise observations and will answer the following research questions:

1.3.1 Research Questions

This research has four main Research Questions (RQ) as follows:

RQ1: How to construct a theoretical foundation for imprecise data analysis and address MCIMO problem?

RQ2: How to handle single-source domain adaptation problem with imprecise observations?

RQ3: How to handle the multi-source transfer learning problem when the instances in the source or target domains are imprecise?

RQ4: How to handle the UniDA problem when the instances in the source or target domains are imprecise?

1.3.2 Research Objectives

To answer these research questions, we set up the corresponding Research Objectives (**RO**):

RO1: Develop a theoretical foundation for imprecise data analysis and some algorithms to solve the MCIMO problem (aims to answer RQ1).

The theoretical analysis of multi-class classification has proved that the existing multi-class classification methods can train a classifier with high classification accuracy,

as long as the instances in the training and test sets are precisely drawn from the same distribution and the size of the training set approaches infinity. These theoretical analysis are based on some different measures, such as Rademacher complexity [70, 102, 105], VC-dimension [1, 27], stability and PAC-Bayesian [57, 103], and local Rademacher Complexity [78, 162]. In this thesis, we use fuzzy random variable, which was proposed in [73, 119, 157], to represent the imprecise feature of the instances and we give a formal definition of fuzzy distribution. Then, the estimation error bounds, based on fuzzy Rademacher complexity, are presented to provide a theoretical analysis of the MCIMO problem. Finally, some new algorithms based on fuzzy techniques are constructed to solve the MCIMO problem.

RO2: Develop a theoretical analysis of domain adaptation problem with imprecise observations and novel models for addressing this problem in single-source scenario (aims to answer RQ2).

Recently, Ben-David *et al.* [6] proposed the learning bounds of the traditional domain adaptation problem with crisp-valued observations, which illustrates that the risk in the target domain is upper bounded by three terms: the risk in the source domain, the marginal distribution discrepancy, and the combined risk. Most existing domain adaptation models utilize some metrics to estimate the distribution discrepancy between source domain and target domain. For example, *Maximum Mean Discrepancy* (MMD) [53]. However, when the instances in source and target domains are all imprecise, the existing metrics could not be used. Therefore, to address this problem, we should construct a new metric which can estimate the fuzzy distribution discrepancy between the source domain and the target domain when the instances in both domains are imprecise.

RO3: Develop new models for addressing multi-source transfer learning problem when the instances in the source or target domains are imprecise (aim to answer RQ3).

Transfer learning for single-source scenarios has been extensively studied. Re-

cent works in transfer learning have increasingly focused on multi-source scenarios [84, 90, 97], aiming to leverage knowledge from multiple source domains to enhance adaptation performance in the target domain. However, existing multi-source transfer learning algorithms fail when instances in the source or target domains are imprecise. Additionally, most existing methods ignore the inherent uncertainty correlation between different source domains and the target domain, leading to an inability to effectively combine knowledge from multiple source domains. Therefore, it is imperative to develop new multi-source transfer learning models to address these issues.

RO4: Develop a new framework for addressing the UniDA problem with imprecise data (aim to answer RQ4).

Most existing domain adaptation methods are proposed for the closed-set scenario [48], where the source and target domains completely share the class of their samples. However, closed-set domain adaptation assumes that the source and target domains share a common label set, an assumption that does not always hold in real-world scenarios. Consequently, three special cases of domain adaptation have been identified: 1) partial domain adaptation, where the source domain contains private categories [15]; 2) open-set domain adaptation, where the target domain contains private categories [44]; and 3) open-partial domain adaptation, where both the source and target domains contain private categories [72]. Recently, a more general case called Universal Domain Adaptation [171] has been proposed to address scenarios with no prior knowledge of the label set in the target domain. In this thesis, we consider a more unique scenario where both the source and target domains contain domain shift and category shift, and instances in the source or target domains are imprecise. We will develop a new domain adaptation model to handle this scenario.

1.4 Research Contribution

The main contributions of this study are summarised as follows:

Contribution 1. Two fuzzy technique-based machine learning algorithms called DF-SVM and DF-MLP are constructed to address the MCIMO problem, which combine fuzzy techniques with SVM and neural networks.

1) This study is the first to build a theoretical foundation for imprecise data analysis.

2) This study provides a theoretical analysis of the MCIMO problem based on fuzzy Rademacher complexity, demonstrating that it is possible to train a fuzzy classifier with high classification accuracy. This theoretical framework establishes a solid foundation for fuzzy data analysis.

3) The two algorithms significantly enhance classification accuracy by utilizing fuzzy vectors to represent the distribution of imprecise data and applying various defuzzification methods to extract crisp-valued information from imprecise observations.

Contribution 2. A new algorithm called Mv-IIE is developed by using fuzzy techniques and multi-view learning to solve a new classification problem called learning from interval-valued data.

1) The estimation error bounds for this problem, based on Rademacher complexity, are provided, ensuring that a classifier can always be trained on the interval-valued data with high classification accuracy. Additionally, the learnability of the underlying problem under perfect observations is also discussed.

2) This study is the first to construct a novel framework that integrates fuzzy techniques with multi-view learning to extract crisp-valued information from interval-valued features and applies this framework to enhance classification accuracy when only interval-valued observations are available.

3) A novel framework for protecting data privacy called INPP is presented to show

an application of Mv-IIE. This is the first to explore the use of interval-valued data properties for achieving data privacy protection.

Contribution 3. A new model called SP-TSF is developed to solve the domain adaptation problem with imprecise observations problem.

1) This study is the first to formalize a more realistic and challenging setting compared to traditional domain adaptation problems. Additionally, a new theoretical bound is established to provide a theoretical foundation for this problem.

2) SP-TSF utilizes the Takagi-Sugeno fuzzy rule-based framework as its foundational structure to capture the intrinsic uncertainty inherent in imprecise data.

3) It introduces a novel integral probability metric design to align the distribution characteristics between the source and target domains with imprecise observations.

4) A deep clustering-based self-supervised pseudo-labeling strategy is developed to enhance class discriminability of the target domain.

Contribution 4. Two fuzzy techniques-based frameworks are constructed to effectively address the multi-source domain adaptation with interval-valued target data problem.

1) This study is the first to address the more realistic and challenging problem, where the aim is to enhance prediction performance on interval-valued target data by leveraging knowledge derived from multiple source domains with crisp-valued features.

2) The first framework, called FUMAT-Net, applies fuzzy transformation function, fuzzy relation, and adversarial training to enhance adaptation performance.

3) The second framework, called FDIM-Net, employs two fuzzy techniques, namely a fuzzy transformation function and fuzzy distances, to tackle the significant extent of uncertainty present in the problem.

Contribution 5. A novel dynamic reweighted loss learning strategy called Distraction-control is developed to handle the unresolved issue in UniDA that the existing UniDA methods can not achieving satisfactory adaptation performance in the hard transfer categories.

1.5 Research Significance

The theoretical and practical significance of this thesis is summarised as follows:

Theoretical significance: This thesis is the first to to build a theoretical foundation for imprecise data analysis and provides theoretical analysis of MCIMO and TLIMO problems. Exploring these areas can lead to the development of unified theoretical frameworks that integrate the handling of imprecise data within the contexts of multi-class classification and transfer learning, thereby offering a more comprehensive understanding of machine learning under uncertainty.

Many real-world datasets contain imprecise or uncertain data due to measurement errors, data entry mistakes, or inherent variability. Developing robust multi-classification algorithms capable of handling such imprecise data ensures that machine learning models can be effectively applied in practical scenarios. Addressing imprecision necessitates sophisticated mathematical and statistical models, such as fuzzy logic and interval analysis, which push the boundaries of current theoretical frameworks and foster the development of new theories and methodologies.

Transfer learning involves adapting models trained on one domain to another. Handling imprecise data in this context ensures that models can be effectively transferred even when the target domain has uncertain data, expanding the applicability of transfer learning. Moreover, transfer learning with imprecise data allows leveraging pre-trained models on large, high-quality datasets to improve performance on smaller, noisier datasets. This maximizes the utility of available data and reduces the need for extensive,

high-quality labeled data in every new domain.

Practical significance: This study presents several models to address different transfer learning scenarios when the observations are imprecise, including single-source, multi-source, and open-set scenarios. The findings help resolve real-world challenges in transfer learning. Additionally, there is significant potential for many other applications to benefit from this study, such as medical diagnostics, natural language processing, and recommender systems. Furthermore, developing methods that handle imprecise data contributes to creating more ethical and fair artificial intelligence systems. By ensuring that models perform reliably despite data uncertainties, we can reduce biases and enhance the trustworthiness of artificial intelligence systems.

1.6 Thesis Structure

The structure of the thesis is shown in Figure 1.1 and the chapters are organised as follows:

- CHAPTER 2 presents the literature of transfer learning and imprecise data analysis, thereby revealing limitations of current research.
- CHAPTER 3 presents a more realistic and challenging problem, which considers addressing the multi-class classification problem when only imprecise observations are available and develops a new framework to handle this problem. This chapter addresses RQ1 to achieve RO1.
- CHAPTER 4 presents 1) a comprehensive theoretical analysis of learning from interval-valued data; 2) a new algorithm to solve this problem via fuzzy techniques and multi-view learning; and 3) a novel framework for protecting data privacy to show an application of the proposed algorithm. This chapter addresses RQ1 to achieve RO1 when data feature is interval-valued.

- CHAPTER 5 presents a new model to solve a more realistic and challenging problem: domain adaptation problem with imprecise observations. This model uses the Takagi-Sugeno fuzzy rule-based model as its foundational structure, and constructs a novel integral probability metric to decrease the distribution discrepancy between the source and target domains with imprecise observations. This chapter addresses RQ2 to achieve RO2.

- CHAPTER 6 presents two fuzzy techniques-based frameworks to effectively address the multi-source domain adaptation with interval-valued target data problem. These two frameworks respectively apply the new designed fuzzy relation and fuzzy distances to measure the correlation between the multiple source domains and the target domain. This chapter addresses RQ3 to achieve RO3.

- CHAPTER 7 presents a novel dynamic reweighted loss learning strategy to focus on the hard transfer categories during adaptation process. With this strategy, the performance of the existing UniDA methods can be significant improved. Then, we can apply fuzzy transformation function and this proposed strategy to address UniDA with imprecise observations problem. This chapter addresses RQ4 to achieve RO4.

- CHAPTER 8 summarises the findings of this thesis and points to directions for future work.

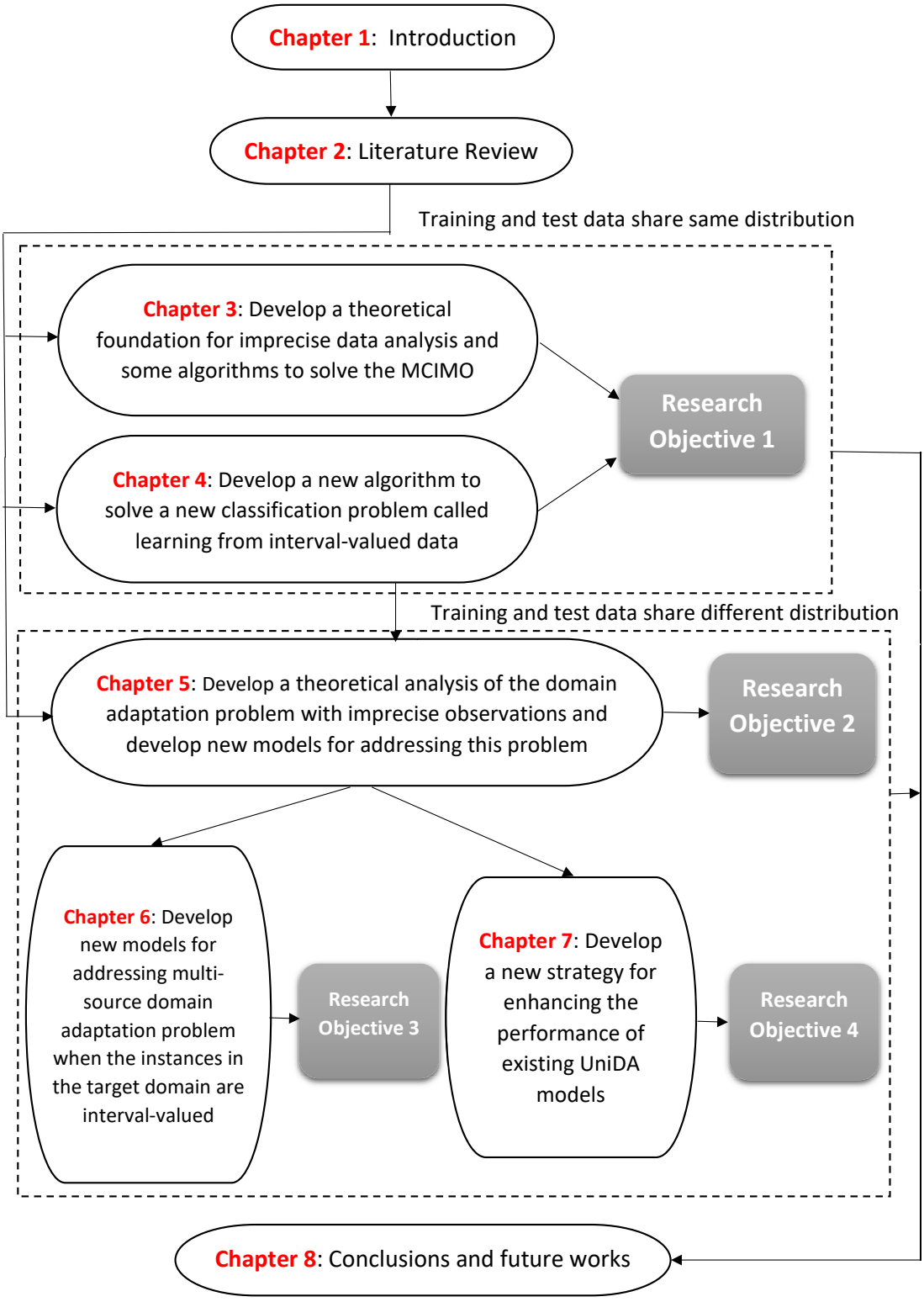


Figure 1.1: Thesis structure

CHAPTER 2

LITERATURE REVIEW

In this chapter, we review recent papers related to this thesis, which includes two main parts: transfer learning and imprecise data analysis.

2.1 Transfer Learning

Transfer learning [112] aims to train a well-performed model in one domain (target) by leveraging knowledge from another domain (source) that has different distribution or learning tasks compared with the previous one. Transfer learning [112] mechanism has drawn great attention in many areas. Based on different perspectives, transfer learning works can be divided into different categories. Firstly, considering the main research streams, the works contains: transfer knowledge of instances [141, 174], transfer knowledge of feature representations [147], transfer knowledge of model parameters [40], and transfer relational knowledge [36]. From the problem setting viewpoint, existing developments can be categorized into multi-task learning [92], domain adaptation [183], cross-domain adaptation [166], and heterogeneous learning [178, 180]. From an applica-

tion perspective, the works can be generally categorized into three tasks: classification [7], unsupervised learning (clustering [26], dimensionality deduction [113]), and regression [10].

In this section, we review two main parts : transfer learning with fuzzy techniques and domain adaptation, which are most related to this thesis.

2.1.1 Transfer Learning with Fuzzy Techniques

Notably, most of the existing transfer learning methods more or less have some limitations when dealing with some unique situations. For example, when few labeled instances can be collected or even only unlabeled instances available in the target domain, this will incur a high degree of uncertainty because there is an obvious interdependence between the certainty level of the learning task and the amount of information available. Therefore, to enhance the capability of existing transfer learning methods and address more realistic problems in many real-world scenarios, many researchers have focused on using fuzzy sets and fuzzy logic [42] to eliminate these limitations. We have divided these recent works into three areas based on the fuzzy technique used. These are fuzzy sets, fuzzy systems, and fuzzy relations.

2.1.1.1 Transfer Learning Based on Fuzzy Sets

Behbood *et al.* [5] proposed an innovative fuzzy-based transfer learning framework to predict long-term bank failures. The framework relies on fuzzy sets, as well as similarity and dissimilarity, to modify the labels of target instances predicted by an fuzzy classifier. Wu *et al.* [156] developed OwARR, a new algorithm that combines fuzzy sets with domain adaptation. The aim is to reduce the amount of object-specific calibration data so as to solve the important regression problem of estimating online drowsiness in drivers from EEG signals in brain-computer interfaces. Gargees *et al.* [49] proposed a transfer

learning method for the possibilistic c-means clustering problem with insufficient data, overcoming a crucial problem for clustering tasks where the source and target domains have a different number of clusters. Based on the idea of fuzzy sets, the proposed algorithm employs historical cluster centers of the data in the source domain as a reference to guide the clustering of data in the target domain.

In terms of applying type-2 fuzzy sets to transfer learning models, Sun *et al.* [139] proposed a new transfer learning model to address the uncertainty caused by conflicting implications in text sequence recognition. The proposed model uses *fuzzy c-means* (FCM) to transform the correspondences among words into information granules. By integrating type-2 fuzzy sets into a hidden Markov model, this granular information can be used for sequence recognition. To reliably estimate GDP from only CO_2 emission data, Shukla *et al.* [137] proposed a new approach to a *kernel extreme learning machine* (KELM) that combines transfer learning with interval type-2 fuzzy sets. Interval type-2 fuzzy sets are used to improve the efficiency of the knowledge transfer. To consider the uncertainty in input datasets, Kumar *et al.* [71] presented a novel transfer learning approach that incorporates type-1 and interval type-2 fuzzy sets into a KELM framework. The aim is to predict GDP based on uncertain carbon emissions data.

In general, fuzzy sets have been widely applied to address uncertainty in data in transfer learning scenarios, and, experimentally, they have been shown to improve both the efficiency and accuracy of knowledge transfer in comparison to non-fuzzy methods.

2.1.1.2 Transfer Learning Based on Fuzzy Systems

Most existing transfer learning methods have a number of drawbacks. For instance, the performance of model-based transfer learning algorithms is heavily dependent on the selected classifier. Additionally, feature-based transfer learning methods can negatively impact the discriminant information and geometric properties of instances from both the source and target domains. Further, the lack of interpretability and an inability to

handle uncertainty are two significant flaws. To address these issues, researchers have turned to fuzzy rule-based systems to improve interpretability and handle uncertainty. Notably, *Takagi-Sugeno-Kang Fuzzy System* (TSK-FS) [140] has received significant attention in this regard.

Shell *et al.* [134] proposed FuzzyTL, a novel structure that combines transfer learning with a fuzzy rule-based system. This structure is designed to bridge the knowledge gap between contexts that lack prior direct contextual knowledge. Meher *et al.* [104] developed an interpretable domain adaptation method, named the rule-based fuzzy ELM classification model, that uses a fuzzy inference system to design an ELM architecture for remote sensing image classification. The model uses the maximum fuzzy membership grade of features, which is characterized by class-belonging fuzzification, to construct the fuzzy rules and two rule extraction matrices. Moreover, Deng *et al.* [32, 33] proposed two novel transfer learning approaches for regression tasks using the Mamdani-Larsen fuzzy system and TSK-FS coupled with a new fuzzy logic algorithm and its objective functions. However, they noticed that the antecedent parameters of the TSK-FS model constructed in the target domain were directly inherited from the source domain, which meant that they could not leverage enough knowledge from the source domain. To address this problem, Deng *et al.* [34] proposed a new transfer learning method that contains two knowledge-leveraging strategies to better learn the antecedent and consequent parameters in the TSK-FS model. First, they applied an FCM-based clustering transfer technique to the antecedent parameters, which means that the antecedent parameters can be learned from both the source and target domains. Second, they introduced an enhanced knowledge-leverage mechanism to learn the consequent parameters. Another knowledge-leverage term is then introduced to make more effective use of the knowledge in the source domain. Further, they applied and modified these methods so that they could be used for analysis in scenarios with insufficient data, such as recognizing EEG

signals [35, 62–64] or with situations involving multiple-source domains [62]. The aim of transfer representation learning is to learn a shared space that matches the distributions of instances from both domains. However, transfer representation learning based on kernels suffers from some shortcomings, such as a lack of interpretability and difficulties with selecting a kernel function. To overcome these issues, Xu *et al.* [163] proposed a new transfer representation learning method that uses TSK-FS instead of kernel functions to realize nonlinear transformations. In this approach, instances from both domains are transformed into a fuzzy feature space to minimize the differences between the distributions. Meanwhile, any discriminant information or geometric properties are preserved using latent Dirichlet allocation and principal component analysis.

Notably, Zuo *et al.* [185] devised a new way of constructing a TSK-FS model for regression tasks. This model uses data from the source domain to construct fuzzy rules and then modifies these rules using a nonlinear continuous function based on sigmoid functions to estimate values in the target domain. To address any significant difference in the label distribution between the source and target domains, Zuo *et al.* [184] developed some fuzzy system-based domain adaptation models for classification tasks. In [186], they applied granular computing techniques to transfer learning and proposed a comprehensive domain adaptation framework based on a *Takagi-Sugeno* (T-S) fuzzy model to handle three different regression scenarios: one where the source and target domains share different conditions, one where they share different conclusions, and one where both apply. Moreover, they identified two issues in fuzzy transfer learning that had not yet been resolved: how to choose an appropriate source domain and how to efficiently select labeled data for the target domain when the target data structure is unbalanced. The solutions, which involve an innovative method again based on a T-S fuzzy model [181], combine an infinite Gaussian mixture model with active learning to improve the performance and generalizability of the initial model. To address a more challenging

problem in multi-source domain adaptation where no source data is available, Li *et al.* [80] proposed a new model based on a deep neural network with fuzzy rules.

Importantly, all the domain adaptation studies mentioned so far only work when both domains have identical feature spaces and the same number of fuzzy rules, i.e., they are all methods of homogeneous domain adaptation. Zuo *et al.* [182], however, devised a novel approach to heterogeneous scenarios based on a T-S fuzzy model. In this framework, fuzzy rules are constructed in the source domain and then transferred to the target domain using canonical correlation analysis so as to minimize the discrepancy between the feature spaces of the two domains. This was the first article to solve heterogeneous domain adaptation problems using a fuzzy rule-based system. Subsequently, Lu *et al.* [97] addressed the more challenging scenario of when the only available instances to build the model span multiple source domains. They proposed two novel transfer learning methods for regression tasks based on a T-S fuzzy model - one for when the feature spaces are homogeneous and one for when the spaces are heterogeneous. In the former, knowledge from multiple source domains is merged in the form of fuzzy rules, while, in the latter, knowledge is merged in the form of both data and fuzzy rules.

In summary, most of the above methods share a common model construction framework: they begin by constructing a fuzzy rule-based model on the source data (e.g., a TSK-FS) and subsequently modify the existing model (fuzzy rules) to establish a new fuzzy model for the target domain. Fuzzy rule-based systems provide a linguistic representation of knowledge, enabling generalization and adaptation, while also making the model more robust to domain shift. Their power to transfer relevant knowledge also helps to improve a model's interpretability. All these characteristics make fuzzy rule-based systems well-suited to transfer learning tasks - particularly, the more challenging tasks, such as heterogeneous domain adaptation and source-free domain adaptation.

2.1.1.3 Transfer Learning Based on Fuzzy Relations

Most studies mentioned so far focus on supervised or semi-supervised transfer learning in homogeneous scenarios, where both the source and target domains have labeled instances and only their data distributions are different. However, it is not uncommon in the real-world for there to be no available labeled instances in the target domain. Further, the feature spaces of the source and target domains will usually be different. This scenario, which is characterized by a high degree of uncertainty, is commonly referred to as *heterogeneous unsupervised domain adaptation* (HeUDA). Recently, researchers have developed n-dimensional fuzzy geometry theory [110] and fuzzy equivalence relations [165] to analyze and handle such problems with uncertainty.

Liu *et al.'s* [87] solution to HeUDA problems, called F-HeUDA, is to use fuzzy geometry to measure the similarity of features between the source and target domains. Shared fuzzy equivalence relations are then introduced, which means both domains will share the same number of clustering categories. Hence, knowledge can be transferred from a heterogeneous source domain to a target domain with only unlabeled data. Using these techniques, F-HeUDA outperformed the SOTA models on four real datasets, and performed especially well when the target domain had very few instances. Moreover, Liu *et al.* [88, 90] focused on a more realistic problem called the multi-source HeUDA problem. Solving this problem involves transferring knowledge from several different source domains that have labeled data but heterogeneous dimensions and one target domain with unlabeled data. Their approach, called a shared fuzzy equivalence relations neural network, improves upon previous work in shared fuzzy equivalence relations to extract the shared fuzzy information contained in multiple heterogeneous domains.

In summary, because there is a high degree of uncertainty when transferring knowledge from a heterogeneous source domain to a target domain with only unlabeled data, non-fuzzy models will not usually perform well. Fuzzy relations offer a flexible, in-

terpretable, and adaptable framework for representing and transferring knowledge between such domains. Hence, researchers tend to apply fuzzy relations to improve transfer efficiency in heterogeneous situations.

2.1.2 Domain Adaptation

Domain adaptation [93] (DA) refers to the process of adapting a machine learning model trained on a source domain to achieve good performance on a different target domain. In this section, we review five main parts : DA theory, single-source DA, *multi-source domain adaptation* (MSDA), *source-free domain adaptation* (SFDA), and *universal domain adaptation* (UniDA), which are most related to this thesis.

2.1.2.1 Domain Adaptation Theory

Ben-David *et al.* [6] proposed the learning bounds of traditional DA problems, which illustrates that the risk in the target domain is upper bounded by three terms: the risk in the source domain, the marginal distribution discrepancy, and the combined risk. In pursuit of refining and tightening these bounds, numerous researchers have explored diverse loss functions [100], devised alternative distribution distance metrics [135, 175], and applied the PAC-Bayes framework [50, 51] for DA problem analysis. Based on these theoretical works, many well-known UDA models have been developed to improve performance in the target domain - see, e.g., [48, 93, 94, 130].

2.1.2.2 Single-source Domain Adaptation

Traditional single-source DA methods primarily gravitate towards three principal strategies for domain alignment. The first strategy involves minimizing the distributional disparities between the source and target domains, by employing an array of integral probability metrics. For instance, *deep adaptation network* [93] and *joint adaptation*

networks [95] use maximum mean discrepancy to measure the distribution discrepancy, while Shen *et al.* [135] are guided by Wasserstein distances. The second strategy is the adversarial training directly inspired by domain adaptation theory [6]. For example, *domain adversarial training of neural networks* [48] applies a domain discriminator with a gradient reversal layer to obtain domain-invariant features. Similarly, *conditional adversarial domain adaptation* [94] uses the cross-covariance between feature and classifier predictions to capture multimodal information along with an entropy condition to guarantee transferability. The last strategy is a pseudo-labeling strategy that trains a classifier to get pseudo labels for unlabeled target data. Saito *et al.* [130] introduced an asymmetric tri-training methodology, utilizing two networks for labeling unlabeled target samples and a third network trained on pseudo-labeled samples to enhance discriminative capabilities within the target domain.

Most recently, Kang *et al.* [66] consider cross-domain discrepancy distance and aim to minimize the difference through the competition between a generator and discriminator, reducing the cross-domain distribution disparity. Li *et al.* [77] propose a new domain adaptation method founded on adversarial training and a new developed metric named maximum density divergence. Na *et al.* [107] introduced a new *unsupervised domain adaptation* (UDA) approach by combining a fixed ratio-based mixup and confidence-based learning methodologies to address problems with large domain discrepancies. Here, Xiao *et al.* [159] proposed a sample weighting method to balance the sample size of the source and target domains and dynamic weighted learning to make a tradeoff between domain alignment and class discrimination. Further, Xie *et al.* [160] propose a unified framework, called Collaborative Alignment Framework, which simultaneously reduces the global domain discrepancy and preserves the local semantic consistency for cross-domain knowledge transfer in a collaborative manner.

Limitations: Unfortunately, these theories and methods cannot be directly applied to

imprecise data. Consequently, this thesis delves into an extensive theoretical analysis of our preceding research and subsequently formulates a model, guided by this theoretical foundation, specifically tailored to address the challenges of domain adaptation with imprecise observation problem (See Chapter 5).

2.1.2.3 Multi-source Domain Adaptation

MSDA aims to leverage knowledge from multiple source domains to enhance adaptation performance. Peng *et al.* [117] applied moment matching to dynamically align the feature distributions of multiple source and target domains. Yang *et al.* [167] designed a dynamic curriculum for source samples, iteratively learning to identify the most suitable domains or samples for alignment with the target. Wang *et al.* [150] presented a new MSDA framework based on a knowledge graph constructed from the prototypes of multiple domains, providing guidance for target prediction. Additionally, Ren *et al.* [124] proposed a novel strategy for MSDA, involving the construction of pseudo target domains using each pair of source and target domains. These pseudo target domains are then aligned with the remaining source domains to improve adaptation performance.

Limitations: However, most of these MSDA methods ignore the inherent uncertainty correlation between different source domains and the target domain, leading to the inability to effectively combine knowledge from multiple source domains. To tackle this issue, we introduce a fuzzy distance-based method designed to measure the distribution discrepancy between different source domains and the target domain (See Chapter 6).

2.1.2.4 Source-free Domain Adaptation

SFDA, where source data is inaccessible during the adaptation process, was first proposed in [83] to address the problem of source private data leakage. In that work, information maximization and self-supervised pseudo-labeling techniques were employed to achieve domain alignment. Recently, other strategies have also been explored to tackle SFDA

problems. For instance, adversarial learning [81] and local structure clustering strategies [168] have been employed. Furthermore, [39, 80] focused on an even more challenging scenario: Multi-Source-Free Domain Adaptation. [39] utilized a confident-anchor-induced pseudo-labeling strategy, while [80] applied fuzzy rule-based deep neural networks to handle data uncertainty in MSDA without the availability of source data.

Limitations: Unfortunately, the majority of existing MSDA and SFDA works assume that the data from the target domain are crisp-valued, rendering them unsuitable for addressing more realistic problems where the target data is imprecise. Consequently, this thesis proposes the use of fuzzy techniques-based neural networks to bridge this gap (See Chapter 6).

2.1.2.5 Universal Domain Adaptation

UniDA is one of the most realistic yet challenging domain adaptation scenarios [148], as both the source and target domains may contain private categories. In recent years, several innovative models have been proposed to tackle domain alignment and the identification of unknown samples in the UniDA setting. UAN [171], for example, quantifies sample-level transferability to distinguish common and private categories and identify unknown samples. CMU [47] introduces a novel transferability measure by combining multiple complementary uncertainty measures to detect unknown samples in the target domain. DANCE [128] employs neighborhood clustering and entropy separation loss functions to reduce the category shift between the source and target domains. DCC [76] leverages domain consensus knowledge to support target clustering and address discrepancies between the private categories. OVANet [129] introduces a one-vs-all open-set classifier for domain alignment and unknown sample identification. The classifier is trained on labeled source data and unlabeled target data for hard negative classifier sampling and open-set entropy minimization. Chen *et al.* [20, 21] propose two novel domain alignment strategies based on intrinsic manifold structure relationships for

domain alignment and develop an energy-based universal classification paradigm to detect unknown samples. Chang *et al.* [17] designed a unified framework for UniDA scenarios using an optimal transport-based partial alignment with adaptive filling to detect common classes without predefined threshold values.

Limitations: These existing UniDA methods encounter a significant challenge when attempting to handle hard transfer categories, i.e., where a category in the source and target data have substantially different distributions. In Chapter 7, we introduce a new reweighted learning strategy to address this challenging problem effectively.

2.2 Imprecise Data Analysis

Imprecise data analysis [85] involves the examination and interpretation of data that contains elements of uncertainty, vagueness, or ambiguity. Unlike traditional data analysis, which assumes precise and well-defined data points, imprecise data analysis acknowledges that real-world data often comes with inherent imprecision due to factors like measurement errors, incomplete information, or subjective judgment.

On the one hand, uses fuzzy sets to handle data that is not precise but rather falls into a range of values. This allows for reasoning about data in a way that mimics human reasoning, which is often approximate rather than exact. Colubi *et al.* [22] integrated fuzzy L_2 metrics [138] with the discriminant analysis approach to analyze fuzzy data. Yang *et al.* [170] proposed a novel fuzzy SVM algorithm based on a kernel fuzzy c-means clustering method to deal with the classification problems with outliers or noises. Rong *et al.* [126] introduced a new classification method, which applies the defuzzified Choquet integral to address heterogeneous fuzzy data classification issues. Wang *et al.* [149] presented a novel deep-ensemble-level-based TSK fuzzy classifier to address imbalanced data classification tasks, which achieve both promising classification performance and high interpretability of zero-order TSK fuzzy classifiers.

Moreover, interval-valued data, where all of the observations' features are described by intervals, is also a common data type in real-world scenarios. For example, the data extracted by many measuring devices are not exact numbers but intervals. Francisco *et al.* [111] designs a new approach for interval-valued data principal component analysis based on midpoints and radii of the intervals. In addition, they [28] present a novel partitioning dynamic clustering method for interval-valued data based on suitable adaptive quadratic distances. Based on support vector machines, a new framework is proposed in [145] for interval-valued data regression and classification. In summary, most existing research related to interval-valued data mainly focuses on clustering analysis [28, 29], regression analysis [56, 145, 149], and feature selection [82].

Limitations: There is no research to provide a complete theoretical analysis of learning from imprecise data problem. To fill this gap, we derive the estimation error bounds of this problem based on Rademacher complexity (See Chapter 3). In addition, the learnability of the underlying problem with perfect observation is also discussed (See Chapter 4).

CHAPTER 3

MULTI-CLASS CLASSIFICATION WITH IMPRECISE OBSERVATIONS

3.1 Introduction

Machine learning methods for the multi-class classification problem have gained great achievements in many areas, including medical imaging [131], natural language processing [158], biology [179] and computer vision [108]. The theoretical analysis of existing well-known multi-class classification machine learning algorithms, such as SVM and neural networks, has been well researched [105]. Recently, many researchers considered using different measures to give the estimation error bounds for classification problems that can guarantee the rationality of these algorithms. These measures include Rademacher complexity [70, 102, 105], VC-dimension [1, 27], stability and *probably approximately correct* (PAC)-Bayesian [57, 103], and local Rademacher Complexity [78, 162].

Rademacher complexity is a crucial tool to derive generalization bounds, which measure how well a given hypothesis set can fit random noise. A Rademacher complexity

based bound was first proposed by Koltchinskii and Panchenko [70]. Subsequently, this bound was improved in [105]. Then, Maximov, Amini and Harchaoui [102] presented a new estimation error bound using Rademacher complexity for multi-class classification issues. In addition, to ensure multi-class PAC learnability, a series of estimation error bounds based on VC-dimension and Natarajan dimension were proposed in [1, 27]. Because of the dependence on dimensions, these VC-dimension based bounds rarely apply to large-scale issues. To conduct theoretical analysis of neural networks for multi-class classification problems, Hardt *et al.* [57] and McAllester [103] introduced the new bounds based on stability and PAC-Bayesian. Further, tighter and sharper bounds were proposed in [78, 162] by using local Rademacher complexity. According to these theoretical analyses, it illustrates that we can always learn a good classifier for multi-class classification problems to predict the test set when the instances are precise in the training and test sets with same distribution and enough instances can be collected in the training set.

However, there is one limitation with multi-class classification that the existing methods can not handle the scenario that only imprecise observations are available. For example, the readings on many measuring devices are not exact numbers but intervals because there are only a limited number of decimals available on most of these measuring devices. Thus, this scenario has inspired us to consider a further realistic problem called *multi-class classification with imprecise observations* (MCIMO). With the MCIMO problem, we aim to train a classifier with high classification performance for multi-class classification problems when the features of all the instances in both training and test sets are imprecise (e.g., fuzzy-valued or interval-valued features).

The main challenge to solving the MCIMO problem is how to handle observations with imprecise features. Existing well-known machine learning methods can not be directly used to address the MCIMO problem. Recently, combining fuzzy techniques

with machine learning methods (especially for transfer learning methods [44, 177]) has drawn increasing attention. In Section 2.2, we give a brief review of these machine learning methods with fuzzy techniques [22, 90, 97, 181]. According to these fuzzy-based methods, it demonstrates that fuzzy techniques are powerful tools to analyze imprecise observations and provide better interpretability to handle the uncertainty of different issues. Therefore, we consider using fuzzy techniques to address the MCIMO problem because they can represent the imprecise features of the instances in both training and test sets and can handle different types of uncertainty issues.

In this chapter, we consider using fuzzy random variable, which was proposed in [119, 157], to represent the imprecise feature of the instances. Then, we give the theoretical analysis and obtain the estimation error bounds for the MCIMO problem. In the MCIMO problem, these bounds are really important as it ensures that we can always train a fuzzy classifier with high classification accuracy when the instances are drawn from the same fuzzy distribution and enough fuzzy-feature instances can be collected.

Subsequently, we construct two fuzzy technique-based algorithms, which combine fuzzy techniques with SVM and neural networks to analyze imprecise data. The proposed algorithms contain two main parts. The first part aims to extract the most significant crisp-valued information from imprecise observations, which is the main difficulty of the proposed algorithms. In this chapter, we compare the performance of different defuzzification methods on synthetic datasets to find the optimal defuzzification function for the proposed algorithms. The second part is to classify the extracted crisp-valued information by two well-known machine learning methods: SVM and neural networks. In addition, interval-valued data is also a common type of imprecise data in real-world scenarios. In this chapter, we give one approach to apply the proposed methods to analyze interval-valued data. Finally, experimental results on both synthetic and real-world datasets reveal the superiority of the proposed algorithms and demonstrate that

the proposed fuzzy-based methods can obtain better performance to analyze fuzzy data or interval-valued data than non-fuzzy methods through comparisons with seven baselines. The main contributions of this paper are as follows.

1. We identify a novel problem called MCIMO, which considers addressing the multi-class classification problem when only imprecise observations are available, and we propose a framework to handle this problem. Based on this framework, two fuzzy technique-based machine learning algorithms called DF-SVM and DF-MLP are constructed, which combine fuzzy techniques with SVM and neural networks. These algorithms significantly improve classification accuracy since they use fuzzy vectors to express the distribution of imprecise data and apply different defuzzification methods to extract crisp-valued information from imprecise observations.

2. We give the theoretical analysis of the MCIMO problem based on the fuzzy Rademacher complexity, which ensures that we can always train a fuzzy classifier with high classification accuracy. This theory provides a theoretical basis for imprecise data analysis.

3. By comparing the performance of different defuzzification methods on synthetic datasets, we find the optimal defuzzification function for the fuzzy technique-based algorithms. Through experimental comparisons with several baselines on both synthetic and real-world datasets, it demonstrates the superiority of the proposed algorithms to analysis imprecise data.

3.2 Theoretical Basis for Imprecise Data Analysis

In our thesis, we apply fuzzy random vectors to represent imprecise data. In this section, we introduce some basic definitions related to fuzzy random variables and then we

introduce the probability density function of fuzzy random variables and propose the notion of the join probability density function of fuzzy random vectors.

3.2.1 Fuzzy Random Variables

Definition 3.1 ([157])**.** Let X be a universal set. Then a fuzzy subset $\widetilde{A}$ of X is defined by its membership function $\mu_{\widetilde{A}} : X \to [0,1]$. We can also write the fuzzy set $\widetilde{A}$ as $\{(x, \mu_{\widetilde{A}}(x)) : x \in X\}$. We denote $\widetilde{A}_\alpha = \{x : \mu_{\widetilde{A}}(x) \geq \alpha\}$ as the α-level set of $\widetilde{A}$, where $\widetilde{A}_0$ is the closure of the set $\{x : \mu_{\widetilde{A}}(x) \neq 0\}$.

Definition 3.2 ([157])**.** Let $\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x)) : x \in X\}$. (i) $\widetilde{A}$ is called normal fuzzy set if there exists x such that $\mu_{\widetilde{A}}(x) = 1$.(ii) $\widetilde{A}$ is called convex fuzzy set if $\mu_{\widetilde{A}}(\lambda x + (1-\lambda)y) \geq \min\{\mu_{\widetilde{A}}(x), \mu_{\widetilde{A}}(y)\}$ for $\lambda \in [0,1]$. Namely, $\mu_{\widetilde{A}}$ is a quasi-concave function.

Definition 3.3 ([157])**.** Suppose that $X = \mathbb{R}$. (i) $\widetilde{a}$ is called a fuzzy number if $\widetilde{a}$ is a normal convex fuzzy set and the α-level set, $\widetilde{a}_\alpha$, is bounded $\forall \alpha \neq 0$. (ii) $\widetilde{a}$ is called a closed fuzzy number if $\widetilde{a}$ is a fuzzy number and its membership function $\mu_{\widetilde{a}}$ is upper semicontinuous. (iii) $\widetilde{a}$ is called a bounded fuzzy number if $\widetilde{a}$ is a fuzzy number and its membership function $\mu_{\widetilde{a}}$ has compact support.

Definition 3.4 ([157])**.** $\widetilde{a}$ is called a canonical fuzzy number if it is a closed and bounded fuzzy number and its membership function is strictly increasing on the interval $[\widetilde{a}_0^L, \widetilde{a}_1^L]$ and strictly decreasing on the interval $[\widetilde{a}_1^U, \widetilde{a}_1^U]$.

Let $\mathscr{F}_\mathbb{R}$ be a set of all fuzzy real numbers induced by the real number system R. We define the relation $\sim$ on $\mathscr{F}_\mathbb{R}$ as $\widetilde{x}^1 \sim \widetilde{x}^2$ if and only if $\widetilde{x}^1$ and $\widetilde{x}^2$ are induced by the same real number x. Then $\sim$ is an equivalence relation, and we have the equivalence classes $[\widetilde{x}] = \{\widetilde{a} | \widetilde{a} \sim \widetilde{x}\}$. The fuzzy real number system is denoted as

$$(\mathscr{F}_\mathbb{R}/\sim) = \{\widetilde{x} | \widetilde{x} \in [\widetilde{x}], \widetilde{x} \text{ is the only element from } [\widetilde{x}]\}.$$

Definition 3.5 ([157])**.** Let $(\Omega, \mathscr{A}, P)$ be a probability space and $(\mathscr{F}_{\mathbb{R}}/\sim)$ be a canonical fuzzy real number system and $\widetilde{X}: \Omega \to (\mathscr{F}_{\mathbb{R}}/\sim)$ be a closed fuzzy-valued function. $\widetilde{X}$ is fuzzy random variable if and only if $\widetilde{X}_\alpha^U$ and $\widetilde{X}_\alpha^L$ are random variables for all $\alpha \in [0,1]$.

3.2.2 Fuzzy Probability

First, we give the definition of fuzzy probability density function.

Definition 3.6 ([157])**.** Let R be the universal set, $\widetilde{X}$ is a fuzzy random variable. Suppose $f_{\widetilde{X}_\alpha}(x)$ is the probability density function of $\widetilde{X}_\alpha^L$ and $\widetilde{X}_\alpha^U$, where $[\widetilde{X}_\alpha^L, \widetilde{X}_\alpha^U]$ is the α-cut of $\widetilde{X}$. We define $\widetilde{f}(\widetilde{x})$ as the fuzzy probability density function of $\widetilde{X}$. Then, the membership function of $\widetilde{f}(\widetilde{x})$ is defined as:

$$\mu_{\widetilde{f}(\widetilde{x})}(r) = \sup_{0 \le \alpha \le 1} \alpha 1_{A_\alpha}(r). \tag{3.1}$$

where

$$\begin{aligned} A_\alpha &= [\min_{x \in [\widetilde{x}_\alpha^L, \widetilde{x}_\alpha^U]} f_{\widetilde{X}_\alpha}(x), \max_{x \in [\widetilde{x}_\alpha^L, \widetilde{x}_\alpha^U]} f_{\widetilde{X}_\alpha}(x)] \\ &= [\min\{\min_{\alpha \le \beta \le 1} f_{\widetilde{X}_\alpha}(\widetilde{x}_\beta^L), \min_{\alpha \le \beta \le 1} f_{\widetilde{X}_\alpha}(\widetilde{x}_\beta^U)\}, \max\{\max_{\alpha \le \beta \le 1} f_{\widetilde{X}_\alpha}(\widetilde{x}_\beta^L), \max_{\alpha \le \beta \le 1} f_{\widetilde{X}_\alpha}(\widetilde{x}_\beta^U)\}]. \end{aligned}$$

Then, the definition of fuzzy probability is shown as follow.

Definition 3.7. We denote $\widetilde{D}$ as the fuzzy probability distribution of $\widetilde{X} \in \mathscr{F}_{\mathbb{R}}$ (denoted as $\widetilde{X} \sim \widetilde{D}$), which contains the value range and fuzzy probability density function of $\widetilde{X}$, where D represents the value range of real-valued variable x which induce all fuzzy real numbers in $\widetilde{D}$.

3.2.3 The Real-valued Representation of Fuzzy Expectation

Let $\widetilde{\mathbf{X}} = (\widetilde{x}_1, \widetilde{x}_2, \cdots, \widetilde{x}_p) \in \mathscr{F}_{\mathbb{R}^p}^p$ be p-fuzzy random vector, where $\widetilde{x}_1, \widetilde{x}_2, \cdots, \widetilde{x}_p \in \mathscr{F}_{\mathbb{R}}$ are i.i.d fuzzy random variables.

Definition 3.8. The probability density function of $\widetilde{x}_j$ is $\widetilde{f}_j(\widetilde{x})$, $j=1,\cdots,p$. We denote the joint probability density function of $\widetilde{\mathbf{X}}$ is $\widetilde{f}_{\widetilde{\mathbf{X}}}(\widetilde{x})=\widetilde{f}_1(\widetilde{x_1})\bigotimes\cdots\bigotimes\widetilde{f}_p(\widetilde{x}_p)$ and its membership function is defined by

$$\xi_{\widetilde{f}_{\widetilde{\mathbf{X}}}(\widetilde{x})}(r)=\sup_{0\le\alpha\le1}1_{[\widetilde{f}_{\widetilde{\mathbf{X}}}(\widetilde{x})]_\alpha}(r), \tag{3.2}$$

where

$$\begin{aligned}[\widetilde{f}_{\widetilde{\mathbf{X}}}(\widetilde{x})]_\alpha &= [\prod_{j=1}^{p}\min_{x_j\in[\widetilde{x}_{j\alpha}^L,\widetilde{x}_{j\alpha}^U]}f_{\widetilde{x}_{j\alpha}}(x_j),\prod_{j=1}^{p}\max_{x_j\in[\widetilde{x}_{j\alpha}^L,\widetilde{x}_{j\alpha}^U]}f_{\widetilde{x}_{j\alpha}}(x_j)]\\ &=[\prod_{j=1}^{p}\min\{\min_{\alpha\le\beta\le1}f_{\widetilde{x}_{j\alpha}}(\widetilde{x}_{j\beta}^L),\min_{\alpha\le\beta\le1}f_{\widetilde{x}_{j\alpha}}(\widetilde{x}_{j\beta}^U)\},\prod_{j=1}^{p}\max\{\max_{\alpha\le\beta\le1}f_{\widetilde{x}_{j\alpha}}(\widetilde{x}_{j\beta}^L),\max_{\alpha\le\beta\le1}f_{\widetilde{x}_{j\alpha}}(\widetilde{x}_{j\beta}^U)\}].\end{aligned}$$

Then, we denote $\widetilde{\mathscr{D}}$ as the fuzzy distribution over $\widetilde{\mathscr{X}}\subset\mathscr{F}_{\mathbb{R}^p}^p$, where $\widetilde{\mathscr{D}}$ contains the value range and the joint probability density function of any fuzzy vector belongs to $\widetilde{\mathscr{X}}$. Next, we define the real-valued representation of fuzzy expectation.

Definition 3.9. Suppose $\widetilde{\mathbf{X}}=(\widetilde{x}_1,\widetilde{x}_2,\cdots,\widetilde{x}_p)\sim\widetilde{\mathscr{D}}$ and $\widetilde{x}_j\sim\widetilde{D}_j, j=1,\cdots,p$. $\forall\alpha\in(0,1]$, let the join distribution of $\widetilde{x_1}_\alpha^{L(U)},\cdots,\widetilde{x_p}_\alpha^{L(U)}$ is $D_\alpha^{L(U)}$ and the joint probability density function is $\prod_{j=1}^{p}f_{\widetilde{x}_{j\alpha}}(\widetilde{x}_{j\alpha}^{L(U)}), x_j\in D_j$. So the real-valued representation of fuzzy expectation of $\widetilde{\mathbf{X}}$ is defined as,

$$\begin{aligned}\mathbb{E}_{\widetilde{\mathbf{X}}\sim\widetilde{\mathscr{D}}}[\ell(h(\widetilde{\mathbf{X}}),y)] &= \frac{1}{2}[\int_{D_1}\cdots\int_{D_p}\int_{\alpha}\ell(h(\widetilde{\mathbf{X}}),y)\prod_{j=1}^{p}f_{\widetilde{x}_{j\alpha}}(\widetilde{x}_{j\alpha}^L)dx_1\cdots dx_p d\alpha\\ &\quad+\int_{D_1}\cdots\int_{D_p}\int_{\alpha}\ell(h(\widetilde{\mathbf{X}}),y)\prod_{j=1}^{p}f_{\widetilde{x}_{j\alpha}}(\widetilde{x}_{j\alpha}^U)dx_1\cdots dx_p d\alpha]\\ &=\frac{1}{2}\int_{D_1}\cdots\int_{D_p}\int_{\alpha\in(0,1]}\ell(h(\widetilde{\mathbf{X}}),y)(f_{\widetilde{\mathbf{X}}_\alpha}(\widetilde{\mathbf{X}}_\alpha^L)+f_{\widetilde{\mathbf{X}}_\alpha}(\widetilde{\mathbf{X}}_\alpha^U))dXd\alpha\\ &=\frac{1}{2}\int_{\alpha\in(0,1]}\{\mathbb{E}_{D_\alpha^L}[\ell(h(\widetilde{\mathbf{X}}),y)]+\mathbb{E}_{D_\alpha^U}[\ell(h(\widetilde{\mathbf{X}}),y)]\}d\alpha,\end{aligned}$$

where $\prod_{j=1}^{p}f_{\widetilde{x}_{j\alpha}}(\widetilde{x}_{j\alpha}^{L(U)})\triangleq f_{\widetilde{\mathbf{X}}_\alpha}(\widetilde{\mathbf{X}}_\alpha^{L(U)})$.

3.3 Problem Setting

In this section, we introduce the MCIMO problem. Let $\widetilde{\mathscr{X}} \subset \mathscr{F}_{\mathbb{R}^p}^p$ be the input space and $\mathscr{Y} = \{1, \cdots, K\} \triangleq [K]$ be the output space, and let $\widetilde{\mathscr{D}}$ be an unknown fuzzy distribution over $\widetilde{\mathscr{X}}$. Suppose $\widetilde{S} = \{(\widetilde{\mathbf{X}}_i, y_i)\}_{i=1}^m$ be a sample drawn from $\widetilde{\mathscr{X}} \times \mathscr{Y}$, where $\widetilde{\mathbf{X}}_i = (\widetilde{x}_{i1}, \widetilde{x}_{i2}, \cdots, \widetilde{x}_{ip}), i = 1, 2, \cdots, m$ drawn i.i.d. from $\widetilde{\mathscr{D}}$ and $y_i = f(\widetilde{\mathbf{X}}_i)$ is the ground truth function denoted as,

$$\begin{aligned} f : &\widetilde{\mathscr{X}} \to \mathscr{Y} \\ &(\widetilde{x}_{i1}, \widetilde{x}_{i2}, \cdots, \widetilde{x}_{ip}) \to k. \end{aligned}$$

We noticed that if $\widetilde{\mathbf{X}}_i \in \mathscr{X}$ belongs to the k-th class, then $f(\widetilde{\mathbf{X}}_i) = k$. Let $\mathscr{H} \subset \{h : \widetilde{\mathscr{X}} \to \mathbb{R}^K\}$ be the hypothesis set of the MCIMO problem and $\forall h \in \mathscr{H}$,

$$\begin{aligned} h : &\widetilde{\mathscr{X}} \to \mathbb{R}^K \\ &(\widetilde{x}_{i1}, \cdots, \widetilde{x}_{ip}) \to (h_1(\widetilde{\mathbf{X}}_i), \cdots, h_K(\widetilde{\mathbf{X}}_i)), \end{aligned}$$

where each $h_k(\widetilde{\mathbf{X}}_i), k = 1, \cdots, K$ represents the probability of the instance $\widetilde{\mathbf{X}}_i$ belongs to the k-th category. Then, we give the definition of the loss function with respect to h,

$$\ell : \mathbb{R}^K \times \mathscr{Y} \to \mathbb{R}_+.$$

Let $\mathscr{L}_{\mathscr{H}} = \{l(h(\widetilde{\mathbf{X}}), y) | \widetilde{\mathbf{X}} \in \widetilde{\mathscr{X}}, h \in \mathscr{H}, y \in \mathscr{Y}\}$ be the class of loss functions associated with $\mathscr{H}$.

The traditional multi-class classification problems aim to use the sample $\widetilde{S}$ to find a hypothesis $h \in \mathscr{H}$ which can cause as small as possible risk $R(h)$ with respect to f. In the MCIMO problem, the purpose is similar to traditional multi-class classification problems. Then, we give the definition of the risk with respect to h,

$$R_{\widetilde{\mathscr{D}}}(h) \triangleq R(\ell(h(\widetilde{\mathbf{X}}), y)) = \mathbb{E}_{\widetilde{\mathbf{X}} \sim \widetilde{\mathscr{D}}}[\ell(h(\widetilde{\mathbf{X}}), y)]. \tag{3.3}$$

Thus, to address the MCIMO problem, we are committed to find the optimal hypothesis function h^* to minimize the risk, i.e., $h^* = \arg\min_{h\in\mathscr{H}} R_{\widetilde{\mathscr{D}}}(h)$.

Hence, we give a formal definition of the MCIMO problem.

Multi-class Classification with Imprecise Observations (MCIMO): Let $\widetilde{\mathbf{X}}$ be a fuzzy random vector defined on a fuzzy space $\widetilde{\mathscr{X}} \subset \mathscr{F}_{\mathbb{R}^p}^p$ with the joint probability density function $\widetilde{f}_{\widetilde{\mathbf{X}}}(\widetilde{x})$, and let $\widetilde{\mathscr{D}}$ be an unknown fuzzy distribution over $\widetilde{\mathscr{X}}$. $\mathscr{Y} = [K]$ is denoted as the label space. Let $\mathscr{L}_{\mathscr{H}} = \{\ell(h(\widetilde{\mathbf{X}}), y) : h \in \mathscr{H}, y \in \mathscr{Y}\}$ be the class of loss functions associated with hypothesis set $\mathscr{H}$, where $\ell : \mathbb{R}^K \times \mathscr{Y} \to \mathbb{R}_+$ and $h : \widetilde{\mathscr{X}} \to \mathbb{R}^K, \forall h \in \mathscr{H}$. In multi-class classification with imprecise observations problem, we aim to train a hypothesis function $h^* \in \mathscr{H}$ to minimize the risk, i.e., $h^* = \arg\min_{h\in\mathscr{H}} R(\ell(h(\widetilde{\mathbf{X}}), y))$.

3.4 Theoretical Analysis of MCIMO

In this section, the theoretical analysis of the MCIMO problem is presented. Firstly, the notion of fuzzy Rademacher complexity is introduced. Then, we obtain the estimation error bounds of the MCIMO problem, which guarantees that we can always obtain a fuzzy classifier with high classification accuracy when infinite fuzzy-feature instances are available.

Definition 3.10. Let $\mathscr{L}_{\mathscr{H}}$ be a family of loss functions and $\widetilde{S} = \{(\widetilde{\mathbf{X}}_i, y_i)\}_{i=1}^m$ a sample drawn from $\mathscr{F}_{\mathbb{R}^p}^p \times \mathscr{Y}$. Then, the empirical fuzzy Rademacher complexity of $\mathscr{L}_{\mathscr{H}}$ and $\mathscr{H}$ with respect to the sample $\widetilde{S}$ and $\widetilde{S}_{\mathbf{X}} = \{\widetilde{\mathbf{X}}_i\}_{i=1}^m$ are defined as:

$$\widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{H}}) = \mathbb{E}_{\boldsymbol{\sigma}}[\sup_{\ell\in\mathscr{L}_{\mathscr{H}}} \frac{1}{m}\sum_{i=1}^m \sigma_i \ell(h(\widetilde{\mathbf{X}}_i), y_i)], \tag{3.4}$$

$$\widehat{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}}(\mathscr{H}) = \mathbb{E}_{\boldsymbol{\sigma}}[\sup_{h\in\mathscr{H}} \frac{1}{m}\sum_{i=1}^m \sum_{k=1}^K \sigma_{ik} h_k(\widetilde{\mathbf{X}}_i)],$$

where $\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_m)^T$, with σ_is independent random variables drawn from the Rademacher distribution, i.e. $\mathbb{P}(\sigma_i = +1) = \mathbb{P}(\sigma_i = -1) = \frac{1}{2}, i = 1, \cdots, m$.

Definition 3.11. Let $\widetilde{\mathscr{D}}' \triangleq \widetilde{\mathscr{D}} \times \mathscr{Y}$ and $\widetilde{\mathscr{D}}$ denote the fuzzy distribution according to $\widetilde{S}$ and $\widetilde{S}_{\mathbf{X}}$. Then, the fuzzy Rademacher complexity of $L_{\mathscr{H}}$ and $\mathscr{H}$ are defined as follow:

$$\begin{aligned} \widetilde{\mathscr{R}}_{\widetilde{S}\sim\widetilde{\mathscr{D}}'}(\mathscr{L}_{\mathscr{H}}) &= \mathbb{E}_{\widetilde{\mathscr{D}}'}[\widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{H}})], \\ \widetilde{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}\sim\widetilde{\mathscr{D}}}(\mathscr{H}) &= \mathbb{E}_{\widetilde{\mathscr{D}}}[\widehat{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}}(\mathscr{H})]. \end{aligned} \tag{3.5}$$

Using related lemmas and theorems (See Appendix 3.9) and the theoretical analysis of traditional multi-class classification algorithms (show in [1, 70, 78, 102, 105]), the estimation error bounds with hypotheses $\mathscr{H}$ are show in the following theorem.

Theorem 3.1. *Let* $\widetilde{S} = \{(\widetilde{\mathbf{X}}_i, y_i)\}_{i=1}^m$ *and* $\widetilde{S}_{\mathbf{X}} = \{\widetilde{\mathbf{X}}_i\}_{i=1}^m, \widetilde{X}_i \sim \widetilde{\mathscr{D}} \in \widetilde{\mathbf{X}}, y_i = f(\widetilde{\mathbf{X}}_i)$ *, and suppose that there are* $C_l, C_h > 0$ *such that* $\sup_{h\in\mathscr{H}} \| h \|_\infty \leq C_h$ *and* $\sup_{\|h\|_\infty \leq C_h} \max_y \ell(t,y) \leq C_l$*, and* $\forall \ell \in \mathscr{L}_{\mathscr{H}}$ *is* L_l*-Lipschitz functions. For any* $\delta > 0$*, with fuzzy probability at least* $1-\delta$*, each of the following holds for all* $\ell \in \mathscr{L}_{\mathscr{H}}$*:*

$$\begin{aligned} |\mathbb{E}_{\widetilde{\mathbf{X}}\sim\widetilde{\mathscr{D}}}[\ell(h(\widetilde{\mathbf{X}}), y)] - \frac{1}{m}\sum_{i=1}^m \ell(h(\widetilde{\mathbf{X}}_i), y_i)| &\leq 2\widetilde{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{H}}) + C_l\sqrt{\frac{2\log(1/\delta)}{m}} \\ |\mathbb{E}_{\widetilde{\mathbf{X}}\sim\widetilde{\mathscr{D}}}[\ell(h(\widetilde{\mathbf{X}}), y)] - \frac{1}{m}\sum_{i=1}^m \ell(h(\widetilde{\mathbf{X}}_i), y_i)| &\leq 2\widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{H}}) + 3C_l\sqrt{\frac{2\log(2/\delta)}{m}}. \end{aligned} \tag{3.6}$$

Because $\forall \ell \in \mathscr{L}_{\mathscr{H}}$ *is* L_l*-Lipschitz functions, we have*

$$\begin{aligned} \widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{H}}) &\leq \sqrt{2}L_l\widehat{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}}(\mathscr{H}) \\ \widetilde{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{H}}) &\leq \sqrt{2}L_l\widetilde{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}}(\mathscr{H}). \end{aligned} \tag{3.7}$$

Then,

$$\begin{aligned} |R_{\widetilde{\mathscr{D}}}(h) - \widehat{R}_{\widetilde{\mathscr{D}}}(h)| &\leq 2\sqrt{2}L_l\widetilde{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}}(\mathscr{H}) + C_l\sqrt{\frac{2\log(1/\delta)}{m}} \\ |R_{\widetilde{\mathscr{D}}}(h) - \widehat{R}_{\widetilde{\mathscr{D}}}(h)| &\leq 2\sqrt{2}L_l\widehat{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}}(\mathscr{H}) + 3C_l\sqrt{\frac{2\log(2/\delta)}{m}}. \end{aligned} \tag{3.8}$$

The detailed proof of theorem 3.1 can be found in Appendix 3.9.

In Section 3.5, we decompose the hypothesis function into defuzzification function and optimization function. We let the loss function $\ell(h(\widetilde{\mathbf{X}}_i), y_i) = \ell(g(M(\widetilde{\mathbf{X}}_i)), y_i)$, where g is a optimization function that maps $\mathbb{R}^p$ into $\mathbb{R}^K$. Let $\mathscr{M} \subset \{M : \widetilde{\mathscr{X}} \to \mathbb{R}^p\}$ be the class of defuzzification functions, $\mathscr{G}_{\mathscr{M}} \subset \{g(M(\widetilde{\mathbf{X}})) : \mathbb{R}^p \to \mathbb{R}^K | M \in \mathscr{M}, y \in \mathscr{Y}\}$ be the class of

optimization functions associated with $\mathscr{M}$, and $\mathscr{L}_{\mathscr{G}} = \{\ell(g(M(\widetilde{\mathbf{X}}_i)), y) | M \in \mathscr{M}, g \in \mathscr{G}, y \in \mathscr{Y}\}$ be the class of loss functions associated with $\mathscr{G}$. Then, we have:

$$
\begin{aligned}
\widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{G}}) &= \mathbb{E}_{\boldsymbol{\sigma}}[\sup_{\ell \in \mathscr{L}_{\mathscr{G}}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i \ell(g(M(\widetilde{\mathbf{X}}_i)), y_i)], \\
\widehat{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}}(\mathscr{G}_{\mathscr{M}}) &= \mathbb{E}_{\boldsymbol{\sigma}}[\sup_{g \in \mathscr{G}} \frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} \sigma_{ik} g_k(M(\widetilde{\mathbf{X}}_i))], \\
\widehat{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}}(\mathscr{M}) &= \mathbb{E}_{\boldsymbol{\sigma}}[\sup_{M \in \mathscr{M}} \frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} \sum_{j=1}^{p} \sigma_{ikj} M(\widetilde{x}_{ij})]
\end{aligned}
\tag{3.9}
$$

Then, we can get the following theorem using theorem 3.1.

Theorem 3.2. *Let* $\widetilde{S} = \{(\widetilde{\mathbf{X}}_i, y_i)\}_{i=1}^m$ *and* $\widetilde{S}_{\mathbf{X}} = \{\widetilde{\mathbf{X}}_i\}_{i=1}^m, \widetilde{\mathbf{X}}_i \sim \widetilde{\mathscr{D}} \in \widetilde{\mathbf{X}}, y_i = f(\widetilde{\mathbf{X}}_i)$ *, and suppose that there are* $C, C_l > 0$ *such that* $\sup_{g \in \mathscr{G}} \| g \|_\infty \leq C$ *and* $\sup_{\|g\|_\infty \leq C} \max_y \ell(t, y) \leq C_l$*, and* $\forall \ell \in \mathscr{L}_{\mathscr{G}}$ *is* L_l*-Lipschitz functions. For any* $\delta > 0$*, with fuzzy probability at least* $1 - \delta$*, each of the following holds for all* $g \in \mathscr{L}_{\mathscr{G}}$*:*

$$
\begin{aligned}
|\mathbb{E}_{\widetilde{\mathbf{X}} \sim \widetilde{\mathscr{D}}}[\ell(g(M(\widetilde{\mathbf{X}})), y)] - \frac{1}{m} \sum_{i=1}^{m} \ell(g(M(\widetilde{\mathbf{X}}_i)), y_i)| &\leq 2\widetilde{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{G}}) + C_l \sqrt{\frac{2\log(1/\delta)}{m}} \\
|\mathbb{E}_{\widetilde{\mathbf{X}} \sim \widetilde{\mathscr{D}}}[\ell(g(M(\widetilde{\mathbf{X}})), y)] - \frac{1}{m} \sum_{i=1}^{m} \ell(g(M(\widetilde{\mathbf{X}}_i)), y_i)| &\leq 2\widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{G}}) + 3C_l \sqrt{\frac{2\log(2/\delta)}{m}}.
\end{aligned}
\tag{3.10}
$$

Because $\forall \ell \in \mathscr{L}_{\mathscr{G}}$ *is* L_l*-Lipschitz functions, we have*

$$
\begin{aligned}
\widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{G}}) &\leq \sqrt{2} L_l \widehat{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}}(\mathscr{G}_{\mathscr{M}}) \\
\widetilde{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{G}}) &\leq \sqrt{2} L_l \widetilde{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}}(\mathscr{G}_{\mathscr{M}}).
\end{aligned}
\tag{3.11}
$$

Then,

$$
\begin{aligned}
|R_{\widetilde{\mathscr{D}}}(h) - \widehat{R}_{\widetilde{\mathscr{D}}}(h)| &\leq 2\sqrt{2} L_l \widetilde{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}}(\mathscr{G}_{\mathscr{M}}) + C_l \sqrt{\frac{2\log(1/\delta)}{m}} \\
|R_{\widetilde{\mathscr{D}}}(h) - \widehat{R}_{\widetilde{\mathscr{D}}}(h)| &\leq 2\sqrt{2} L_l \widehat{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}}}(\mathscr{G}_{\mathscr{M}}) + 3C_l \sqrt{\frac{2\log(2/\delta)}{m}}.
\end{aligned}
\tag{3.12}
$$

The proof of theorem 3.2 is similar to theorem 3.1.

Next, we consider the estimation error bounds for kernel-based optimization functions such as SVM. Let $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ be a PDS kernel function, $\Phi : \mathbb{R}^p \to \mathbb{H}$ be a feature mapping associated to K and $w_1, \cdots, w_K \in \mathbb{H}$ are weight vectors. For any $p \geq 1$, the family of kernel-based hypotheses is denoted as:

$$
\begin{aligned}
\mathscr{G}_{K,p} = \{g : M(\widetilde{\mathbf{X}}) \to (w_1^T \Phi(M(\widetilde{\mathbf{X}})), \cdots, w_K^T \Phi(M(\widetilde{\mathbf{X}}))), \\
W = (w_1^T, \cdots, w_K^T)^T, ||W||_{\mathbb{H},p} \leq \Lambda\},
\end{aligned}
$$

where, $||W||_{\mathbb{H},p} = (\sum_{l=1}^{K} ||w_l||_{\mathbb{H}}^p)^{1/p}$. Hence, the fuzzy Rademacher complexity of $\mathscr{G}_{K,p}$ can be bounded as follow.

Lemma 3.1. *Let $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ be a PDS kernel function and $\Phi : \mathbb{R}^p \to \mathbb{H}$ be a feature mapping associated to K. Assume that there exists $r > 0$ such that $K(M(\widetilde{\mathbf{X}}), M(\widetilde{\mathbf{X}})) \leq r^2$ for all $\widetilde{\mathbf{X}} \in \widetilde{\mathscr{X}}$. Let $\widetilde{S}_{\mathbf{X}} = \{\widetilde{\mathbf{X}}_i\}_{i=1}^m, \widetilde{\mathbf{X}}_i \sim \widetilde{\mathscr{D}} \in \widetilde{\mathscr{X}}$. Then, for any $m \geq 1$,*

$$\widetilde{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}} \sim \widetilde{\mathscr{D}}}(\mathscr{G}_{K,p}) \leq K\sqrt{\frac{r^2\Lambda^2}{m}}. \tag{3.13}$$

Proof. For all $l \in [K]$, $||w_l||_{\mathbb{H}} \leq (\sum_{l=1}^{K} ||w_l||_{\mathbb{H}}^p)^{1/p} = ||W||_{\mathbb{H},p}$ holds. Thus, as $||W||_{\mathbb{H},p} \leq \Lambda$, we have $||w_l||_{\mathbb{H}} \leq \Lambda$ for all $l \in [1,K]$. Then, the fuzzy Rademacher complexity of the hypothesis set $\mathscr{G}_{K,p}$ can be bounded as follows:

$$\begin{aligned}
\widetilde{\mathscr{R}}_{\widetilde{S}_{\mathbf{X}} \sim \widetilde{\mathscr{D}}}(\mathscr{G}_{K,p}) &= \frac{1}{m}\mathbb{E}_{\widetilde{\mathscr{D}},\boldsymbol{\sigma}}[\sup_{||W||\leq\Lambda} \sum_{i=1}^{m}\sum_{k=1}^{K} \sigma_{ik} g_k(M(\widetilde{\mathbf{X}}_i))] \\
&= \frac{1}{m}\mathbb{E}_{\widetilde{\mathscr{D}},\boldsymbol{\sigma}}[\sup_{||W||\leq\Lambda} \sum_{i=1}^{m}\sum_{k=1}^{K} \sigma_{ik} w_k^T \Phi(M(\widetilde{\mathbf{X}}_i))] \\
&\leq \frac{K}{m}\mathbb{E}_{\widetilde{\mathscr{D}},\boldsymbol{\sigma}}[\sup_{k\in[K],||W||\leq\Lambda} \langle w_k, \sum_{i=1}^{m} \sigma_{ik}\Phi(M(\widetilde{\mathbf{X}}_i))\rangle] \text{(using Cauchy-Schwarz inequality)} \\
&\leq \frac{K}{m}\mathbb{E}_{\widetilde{\mathscr{D}},\boldsymbol{\sigma}}[\sup_{k\in[K],||W||\leq\Lambda} ||w_k||_{\mathbb{H}} || \sum_{i=1}^{m} \sigma_{ik}\Phi(M(\widetilde{\mathbf{X}}_i))||_{\mathbb{H}}] \\
&\leq \frac{K\Lambda}{m}\mathbb{E}_{\widetilde{\mathscr{D}},\boldsymbol{\sigma}}[\sup_{k\in[K]} || \sum_{i=1}^{m} \sigma_{ik}\Phi(M(\widetilde{\mathbf{X}}_i))||_{\mathbb{H}}] \text{(using Jensen's inequality)} \\
&\leq \frac{K\Lambda}{m}[\mathbb{E}_{\widetilde{\mathscr{D}},\boldsymbol{\sigma}}[\sup_{k\in[K]} || \sum_{i=1}^{m} \sigma_{ik}\Phi(M(\widetilde{\mathbf{X}}_i))||_{\mathbb{H}}^2]]^{1/2} \quad (i \neq j \Rightarrow \mathbb{E}_{\boldsymbol{\sigma}}[\sigma_{ik}\sigma_{jk}] = 0) \\
&= \frac{K\Lambda}{m}[\mathbb{E}_{\widetilde{\mathscr{D}}}[\sum_{i=1}^{m} ||\Phi(M(\widetilde{\mathbf{X}}_i))||_{\mathbb{H}}^2]]^{1/2} \\
&= \frac{K\Lambda}{m}[\mathbb{E}_{\widetilde{\mathscr{D}}}[\sum_{i=1}^{m} K(M(\widetilde{\mathbf{X}}_i), M(\widetilde{\mathbf{X}}_i))]]^{1/2} \leq K\sqrt{\frac{r^2\Lambda^2}{m}},
\end{aligned}$$

which yields the result. ■

Next, combining theorem 3.2 and lemma 3.1 directly yields the following generalization bound.

Theorem 3.3. *Let $K:\mathbb{R}^p\times\mathbb{R}^p\to\mathbb{R}$ be a PDS kernel function and $\Phi:\mathbb{R}^p\to\mathbb{H}$ be a feature mapping associated to K. Assume that there exists $r>0$ such that $K(M(\widetilde{\mathbf{X}}),M(\widetilde{\mathbf{X}}))\leq r^2$ for all $\widetilde{\mathbf{X}}\in\widetilde{\mathscr{X}}$. Let $\widetilde{S}_{\mathbf{X}}=\{\widetilde{\mathbf{X}}_i\}_{i=1}^m,\widetilde{\mathbf{X}}_i\sim\widetilde{\mathscr{D}}\in\widetilde{\mathscr{X}}$ and suppose that there are $C,C_l>0$ such that $\sup_{g\in\mathscr{G}_{K,p}}\|g\|_\infty\leq C$ and $\sup_{\|g\|_\infty\leq C}\max_y\ell(t,y)\leq C_l$, and $\forall\ell\in\mathscr{L}_{\mathscr{G}_{K,p}}$ is L_l-Lipschitz functions. For any $\delta>0$, with fuzzy probability at least $1-\delta$, each of the following holds for all $h\in\mathscr{G}_{K,p}$:*

$$|R_{\widetilde{\mathscr{D}}}(h)-\widehat{R}_{\widetilde{\mathscr{D}}}(h)|\leq 2KL_l\sqrt{\frac{2r^2\Lambda^2}{m}}+C_l\sqrt{\frac{2\log(1/\delta)}{m}}. \tag{3.14}$$

According to Eqs. (3.8), (3.12), and (3.14), we notice that fix some constants, as $m\to\infty$, $R_{\widetilde{\mathscr{D}}}(h)\to\widehat{R}_{\widetilde{\mathscr{D}}}(h)$. Therefore, these bounds demonstrate that we can always obtain a fuzzy classifier with high classification accuracy when enough fuzzy-feature instances can be collected. These theoretical analyses reveal that fuzzy classifiers can be constructed to effectively and accurately handle the MCIMO problem.

3.5 Model Construction

In this section, two fuzzy classifiers are constructed to handle the MCIMO problem. In the MCIMO problem, we aim to train a fuzzy classifier for fuzzy-feature input prediction. Let $\widetilde{\mathbf{X}}_i=(\widetilde{x}_{i1},\widetilde{x}_{i2},\cdots,\widetilde{x}_{ip}),i=1,\cdots,m$ be a fuzzy-feature input, where $\widetilde{x}_{ij},i=1,\cdots,m,j=1,\cdots,p$ are fuzzy number. Common used fuzzy numbers include Gaussian fuzzy numbers, trapezoidal fuzzy numbers and triangular fuzzy numbers. Firstly, a Gaussian fuzzy number $\widetilde{x}$ can be characterized by (c,δ) and the membership function is given in the following equation:

$$\mu_{\widetilde{x}}(t)=\exp(-(t-c)/2\delta)^2.$$

A trapezoidal fuzzy number $\widetilde{x}$ can be characterized by (a_1, b_1, b_2, a_2) and the membership function of a trapezoidal fuzzy number $\widetilde{x}$ is shown as follow:

$$\mu_{\widetilde{x}}(t) = \begin{cases} 0, & t < a_1 \\ \dfrac{t-a_1}{b_1-a_1}, & a_1 \le t < b_1 \\ 1, & b_1 \le t < b_2 \\ \dfrac{t-a_2}{b_2-a_2}, & b_2 \le t < a_2 \\ 0, & t \ge a_2. \end{cases}$$

Finally, when $b_1 = b_2$, a trapezoidal fuzzy number is become a triangular fuzzy number. Thus, a triangular fuzzy number $\widetilde{x}$ can be characterized by (a_1, b_1, a_2).

To address the MCIMO problem, we need to construct a hypothesis function $h \in \mathscr{H}$ which mapping the input space $\widetilde{\mathscr{X}} \subset \mathscr{F}_{\mathbb{R}^p}^p$ into $\mathbb{R}^K$. A hypothesis function h can be decomposed into a composition of two functions. The first function M, called defuzzification function, is defined as follow:

$$\begin{aligned} M: \quad & \widetilde{\mathscr{X}} \to \mathbb{R}^p \\ & (\widetilde{x}_{i1}, \widetilde{x}_{i2}, \cdots, \widetilde{x}_{ip}) \to (M(\widetilde{x}_{i1}), \cdots, M(\widetilde{x}_{ip})). \end{aligned}$$

Next, four different defuzzification methods are introduced:

1. The first method is called *Mean / Middle of Maxima* (MOM) [127] which is widely-used due to its calculation simplicity. MOM is defined as:

$$\text{MOM}(\widetilde{x}) = \text{Mean}(t = \arg\max_t \mu_{\widetilde{x}}(t)). \tag{3.15}$$

2. *The Centre of Gravity* (COG) [146] is another widely-used defuzzification method. The definitions of COG for discrete and continuous situations are show as follow:

$$\text{COG}(\widetilde{x}) = \frac{\sum t\mu_{\widetilde{x}}(t)}{\sum \mu_{\widetilde{x}}(t)}(\text{discrete}) = \frac{\int t\mu_{\widetilde{x}}(t)dt}{\int \mu_{\widetilde{x}}(t)dt}(\text{continuous}). \tag{3.16}$$

3. The third approach, called *averaging level cuts* (ALC) [109], is defined as the flat averaging of all midpoints of the α-cuts. ALC is defined as :

$$\mathrm{ALC}(\widetilde{x}) = \frac{1}{2}\int_0^1 (\widetilde{x}_\alpha^L + \widetilde{x}_\alpha^U) d\alpha. \tag{3.17}$$

4. The final method is called *value of a fuzzy number* (VAL) [30] which uses α-levels as weighting factors in averaging the α-cut midpoints. VAL is defined as :

$$\mathrm{VAL}(\widetilde{x}) = \int_0^1 \alpha(\widetilde{x}_\alpha^L + \widetilde{x}_\alpha^U) d\alpha. \tag{3.18}$$

In Section 3.6, we compare the performance of different defuzzification methods on synthetic datasets. The experimental results illustrate that VAL outperforms than other three defuzzification methods. Therefore, Eq. (3.18) is used as the defuzzification function in all subsequent experiments.

Through the first progress, the initial issue becomes a traditional multi-class classification problem with crisp data. Therefore, the second function, called the optimization function, is a hypothesis function that maps $\mathbb{R}^p$ into $\mathbb{R}^K$ to solve the traditional multi-class classification problem. Since support vector machine and neural networks have gained great achievements on multi-classification problems, we decide to apply both algorithms as the optimization method. Next, we will introduce both algorithms for multi-classification problems.

3.5.1 Defuzzified Support Vector Machine

Firstly, support vector machine (one-vs-rest SVM [154]) with PDS kernel function is used as the optimization function to solve the MCIMO problem.

Suppose $D_{tr} = ((\widetilde{\mathbf{X}}_1, y_1), \cdots, (\widetilde{\mathbf{X}}_N, y_N))$ is the training data, where $\widetilde{\mathbf{X}}_i \in \widetilde{\mathscr{X}} \subset \mathscr{F}^p_{\mathbb{R}^p}, y_i \in \{-l, +l\}, l = 1, 2, \cdots, K, i = 1, 2, \cdots, N$. The $-l$ indicates that $\widetilde{\mathbf{X}}_i$ does not belong to category l, and the $+l$ represents that $\widetilde{\mathbf{X}}_i$ belongs to category l. In the first step, defuzzification

Algorithm 1 DF-SVM

Input: the training data D_{tr}, selected appropriate regularization parameter C and kernel function ;
Initial: Preprocessing the training data D_{tr};
Defuzzification: Using equation (3.18) to transform $\widetilde{D}_x = (\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_N)$ into $D_x = (\mathbf{X}_1, \cdots, \mathbf{X}_N)$;
Optimization: Solving K optimization problems in (3.19);
Output: $\boldsymbol{\alpha}_l^* = (\alpha_{1l}^*, \cdots, \alpha_{Nl}^*)^T, l = 1, 2, \cdots, K$ and the decision function in (3.21).

function (3.18) is used to transform fuzzy input $\widetilde{D}_x = (\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_N)$ to crisp input denoted as $D_x = (\mathbf{X}_1, \cdots, \mathbf{X}_N)$. Let $K : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ be a PDS kernel function. Hence, we need to solve K optimization problems separately, and the lth problem is shown as follows:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{il} \alpha_{jl} y_i y_j K(\mathbf{X}_i, \mathbf{X}_j) - \sum_{i=1}^{N} \alpha_{il} \\ s.t \quad & \sum_{i=1}^{N} \alpha_{il} y_i = 0 \\ & 0 \le \alpha_{il} \le C, i = 1, 2, \cdots, N. \end{aligned} \tag{3.19}$$

The optimal solution is $\boldsymbol{\alpha}_l^* = (\alpha_{1l}^*, \cdots, \alpha_{Nl}^*)^T, l = 1, 2, \cdots, K$. Then, choose a positive component $0 \le \alpha_{jl}^* \le C$ of $\boldsymbol{\alpha}_l^*$, and calculate

$$b_l^* = y_j - \sum_{i=1}^{N} \alpha_{il}^* y_i K(\mathbf{X}_i, \mathbf{X}_j). \tag{3.20}$$

Finally, the decision function is given as follow:

$$h(\mathbf{X}) = \arg\max_{l \in [K]} \left(\sum_{i=1}^{N} \alpha_{il}^* y_i K(\mathbf{X}, \mathbf{X}_i) + b_l^* \right). \tag{3.21}$$

The following algorithm called *defuzzified support vector machine* (DF-SVM) is shown in Algorithm 1.

3.5.2 Defuzzified Multilayer Perception

Secondly, a multilayer perception model, which contains two hidden layers and an output layer (softmax), is used as the optimization function to complete the second progress. We denote the parameters of the two hidden layers are W_1, b_1 and W_2, b_2 respectively, and

Algorithm 2 DF-MLP

Input: training data D_{tr}, learning rate η, fixed epoch T_{max}, loss function (cross-entropy loss function is selected) and optimization algorithm (Adam algorithm [67] is selected);
Initial: W_0^0, W_1^0, W_2^0, b_0^0, b_1^0, b_2^0;

1: Fetch mini-batches from $\mathscr{D}_s$ and $\mathscr{D}_t$.
2: **for** $T = 1, 2, \ldots, T_{max}$ **do**
3: Fetch mini-batch $\check{D}_{tr}$ from D_{tr};
4: Calculate $L = loss(h(\widetilde{X}; W_0^{T-1}, W_1^{T-1}, W_2^{T-1}, b_0^{T-1}, b_1^{T-1}, b_2^{T-1}), \widehat{y})$ according to Eqs. (3.18) and (3.22);
5: Update W_0^T, W_1^T, W_2^T, b_0^T, b_1^T, $b_2^T = \text{Adam}(L)$;
6: **end for**

Output: $W_0^{T_{max}}$, $W_1^{T_{max}}$, $W_2^{T_{max}}$, $b_0^{T_{max}}$, $b_1^{T_{max}}$, $b_2^{T_{max}}$.

the parameters of the output layer are W_0, b_0 respectively, and the activation function is ϕ. Then, the outcome of the constructed multilayer perception model can be expressed as when we get a fuzzy-feature input $\widetilde{\mathbf{X}}$:

$$\begin{aligned} O(\widetilde{\mathbf{X}}) &= \phi(\phi(M(\widetilde{\mathbf{X}})W_1 + b_1)W_2 + b_2)W_0 + b_0, \\ \widehat{y} &= \arg\max_{k \in \{1,2,\cdots,K\}}(h_k(\widetilde{\mathbf{X}})), \end{aligned} \tag{3.22}$$

where

$$h(\widetilde{\mathbf{X}}) = (h_1(\widetilde{\mathbf{X}}), \cdots, h_K(\widetilde{\mathbf{X}})) = \text{softmax}(O(\widetilde{\mathbf{X}})).$$

The following algorithm called *defuzzified multilayer perception* (DF-MLP) is shown in Algorithm 2.

3.6 Experiments on Synthetic Datasets

In this section, we first compare the performance of different defuzzification methods on synthetic datasets to select the optimal defuzzification function for the proposed algorithms. Then, we verify the efficacy of the proposed algorithms for solving the MCIMO problem by comparing seven baselines in terms of classification accuracy on synthetic datasets.

3.6.1 Dataset Generation

In this section, we introduce how to construct the synthetic dataset (Balanced data) which contains N fuzzy instances distributed in five categories. Each instance has 20 fuzzy features. Firstly, we generate the real-valued vectors $\mathbf{X}_i = (x_{i1}, \cdots, x_{i20}), i = 1, \cdots, N$ in five categories by a random number generator as the true value of the instance. Then, we use the generated real-valued vectors to construct the observation datasets $\{\widetilde{\mathbf{X}}_i = (\widetilde{x}_{i1}, \cdots, \widetilde{x}_{i20})\}_{i=1}^{N}$. Each $\widetilde{x}_{ij}$ is a triangular fuzzy number characterized by $(x_{ij} - a_{ij}, x_{ij} + b_{ij}, x_{ij} + c_{ij})$ where $a_{ij} \sim U[1.5, 3], b_{ij} \sim U[-0.5, 0.5], c_{ij} \sim U[2, 4]$ and $U[a, b]$ denotes the uniform distribution over $[a, b]$.

3.6.2 Experimental Setup

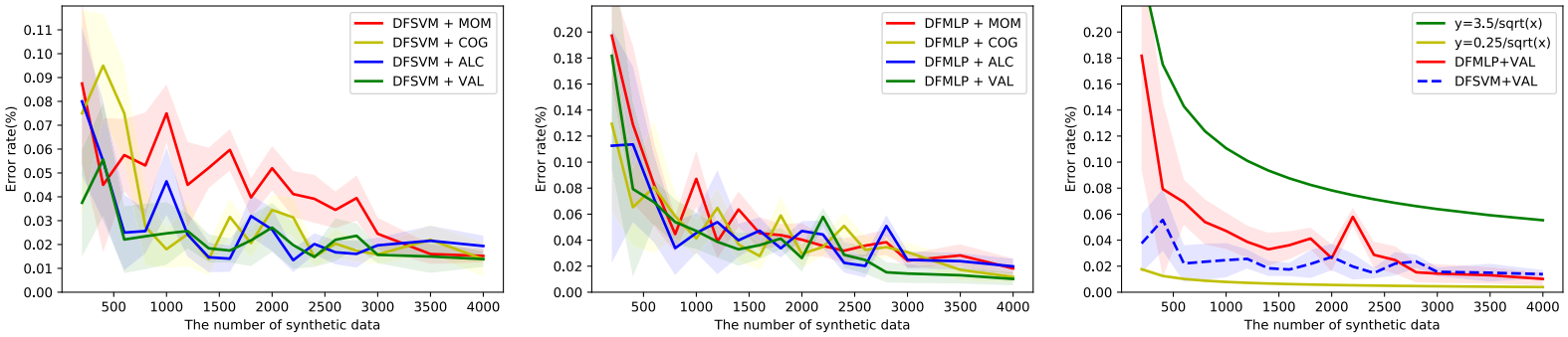
In this section, baselines and experimental details of all baselines, DF-SVM and DF-MLP are introduced.

3.6.2.1 Baselines

Firstly, we introduce the first five baselines which called Meanlogistic, MeanSVM, MeanDecisiontree, MeanRandomForest and MeanMLP. For fuzzy-feature dataset, a fuzzy feature is denoted as $\widetilde{x} = (\inf P_0, \sup P_0, \inf P_1, \sup P_1)$. We use $M_1(\widetilde{x}) = (\inf P_0 + \sup P_0 + \inf P_1 + \sup P_1)/4$ to transfer fuzzy features to crisp features. For interval-valued datasets, $x = [A, B]$ is denoted as a interval-valued feature. Similarly, $M_2(x) = (A + B)/2$ is used to transfer interval-valued features to crisp features. Then, those baselines apply five well-known machine learning methods (logistic regression, SVM, decision trees, random forests and neural networks) to classify crisp-valued data obtained with the above-mentioned methods. Secondly, the last two baselines called DCCF and BCCF are presented in [22].

Table 3.1: Hyperparameters for the proposed algorithms and seven baselines

Algorithm	Hyperparameters	Ranges
Meanlogistic	regularization parameter C	$\{0.1, 0.2, \cdots, 0.9, 1, 2, \cdots, 100\}$
MeanSVM	regularization parameter C, kernel type	$\{0.1, 0.2, \cdots, 0.9, 1, 2, \cdots, 100\}$, {'linear', 'poly', 'rbf'}
MeanDecisiontree	min samples leaf	$\{1, 2, \cdots, 10\}$
MeanRandomForest	min samples leaf, the number of trees	$\{1, 2, \cdots, 10\}$, $\{5, 10, \cdots, 100\}$
MeanMLP	learning rate, hidden layer units, epochs	$\{0.0001, 0.001, 0.01, 0.1\}, \{20, 30, \cdots, 200\}, \{100, 200, 500, 1000, 1500\}$
DCCF[22]	bandwidth h_g	$\{1, 2, \cdots, 10, 20, \cdots, 50\}$
BCCF[22]	distance parameter δ	$\{0.1, 0.5, 1, 2, \cdots, 10\}$
DF-SVM	regularization parameter C, kernel type	$\{0.1, 0.2, \cdots, 0.9, 1, 2, \cdots, 100\}$, {'linear', 'poly', 'rbf'}
DF-MLP	learning rate, hidden layer units, epochs	$\{0.0001, 0.001, 0.01, 0.1\}, \{20, 30, \cdots, 200\}, \{100, 200, 500, 1000, 1500\}$



(a) DF-SVM with 4 defuzzification functions. (b) DF-MLP with 4 defuzzification functions. (c) DF-SVM and DF-MLP with VAL.

Figure 3.1: Classification error rate on the test set varies with the number of synthetic data.

3.6.2.2 Experimental Details

For DF-MLP, we let momentum = 0.9 and weight decay = 0.0001. Finally, for the DCCF and BCCF algorithms, φ is selected to be the Lebesgue measure on $[0,1]$ and $\theta = 1/3$, $K(u) = \frac{15}{8}(1-u^2)^2 I_{(u\in[0,1])}$ is used as the kernel function. All these settings of DCCF and BCCF algorithms can obtain the best performance from [22]. However, DCCF and BCCF algorithms can only process the fuzzy data with one fuzzy feature, whereas the generated synthetic datasets contain multiple fuzzy features. Therefore, we consider using the average distance between each fuzzy feature to represent the distance between the fuzzy feature vectors in the DCCF and BCCF algorithms.

For each algorithm on each dataset, we randomly divide each dataset into the training set, the validation set and the test set, which contain 60%, 20% and 20% of the data, respectively. First, we select the hyperparameters that can obtain the highest average classification accuracy on the validation set. The average classification accuracy on the validation set is the average of the results of 10 repeated experiments on the validation set. The hyperparameters that need to be selected are shown in Table 3.1. Then, the selected optimal hyperparameters are used to test the performance of each algorithm on the test set. We repeat the entire experiment process 20 times. Thus, the final results are shown in the form of "mean± standard deviation." To avoid random errors, we randomly scramble the data before each experiment. Classification accuracy is used to evaluate the performance of the proposed model. The definition of classification accuracy is shown as follows:

$$\text{Accuracy} = \frac{|\widetilde{\mathbf{X}} \in \widetilde{\mathscr{X}} : f(\widetilde{\mathbf{X}}) = h(\widetilde{\mathbf{X}})|}{|\widetilde{\mathbf{X}} \in \widetilde{\mathscr{X}}|},$$

where $f(\widetilde{\mathbf{X}})$ is the ground truth label of $\widetilde{\mathbf{X}}$, while $h(\widetilde{\mathbf{X}})$ is the label predicted by the presented algorithms and the baselines.

In the first experiment, we compare the performance of the proposed two algorithms with different defuzzification functions on the test set when the number of synthetic data

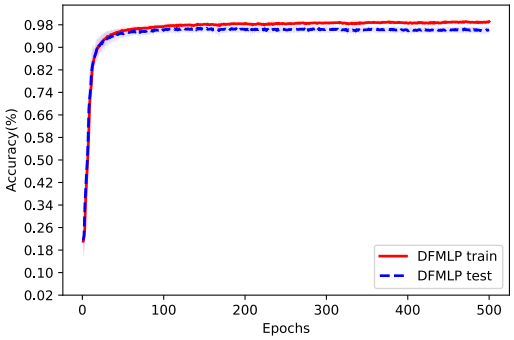


Figure 3.2: Accuracy curve on the synthetic datasets vs. the number of epochs.

increases. The number of synthetic data N is selected from $\{200, 400, \cdots, 3000, 3500, 4000\}$. In the second experiment, we generated 2000 synthetic data and analyzed them using the proposed methods and baselines, respectively. In addition, the Wilcoxon rank-sum test results of the method, which obtains the best performance, with other methods are given.

3.6.3 Experimental Results Analysis

The results of the first experiment are shown in Figure 3.1. From Figures 3.1(a) and 3.1(b), we find that COG and VAL have better performance than another two methods in terms of convergence speed and classification error and VAL is more stable than the other three methods. The reason why VAL can achieve better performance than other methods is that VAL uses all information from fuzzy sets so that some key information is not discarded. In addition, VAL gives less importance to the lower levels of fuzzy sets, which is reasonable from the perspective of the concept of membership function. Therefore, we use VAL as the defuzzification method in the following experiments. Moreover, from Figure 3.1(c), it illustrates that the convergence rate of the two proposed algorithms with VAL defuzzification method is $O(1/\sqrt{m})$. Therefore, we confirmed the theoretical analysis results in Section 3.4 that we can always obtain a fuzzy classifier with high classification accuracy when sufficient fuzzy-feature observations are available.

Table 3.2: Experiment Result of Synthetic Dataset.

Algorithms	Test accuracy	p	Time (sec)
Meanlogistic	96.86% ±0.87%	2.2×10^{-6}*	119.97
MeanSVM	97.72% ±0.71%	0.0337*	127.35
MeanDecisiontree	78.20% ±2.70%	6.3×10^{-8}*	2.23
MeanRandomForest	95.82% ±0.85%	9.8×10^{-8}*	1088.57
MeanMLP	96.16% ±0.80%	3.7×10^{-7}*	6607.89
DCCF[22]	92.58% ±1.02%	6.3×10^{-8}*	1122687
BCCF[22]	92.51% ± 1.03%	6.3×10^{-8}*	1123543
DF-SVM	**98.24%** ± **0.52%**	—-	119.98
DF-MLP	96.90% ± 0.95%	2.2×10^{-5}*	6593.64

The bold value represents the highest accuracy in each column.
p: The p-value of the Wilcoxon rank-sum test between the best performance and other baselines' outcomes.
* $p < 0.05$

The results of the second experiment are illustrated in Table 3.2, and Figure 3.2 shows the classification accuracy curve of Algorithm 2 on the synthetic datasets vs. the number of epochs. From the results, DF-SVM and DF-MLP obtain better performance than the most other baselines on the synthetic dataset. Further, the results of the statistic test show that DF-SVM outperforms other methods significantly at the 0.05 significance level, which demonstrates the superiority of the proposed algorithms. In addition, we present the experimental running times for the proposed algorithms and all baselines.

3.7 Experiments on Real-World Datasets

In this section, five real-world datasets are used to verify the efficacy of proposed algorithms for solving the MCIMO problem by comparing with seven baselines in terms

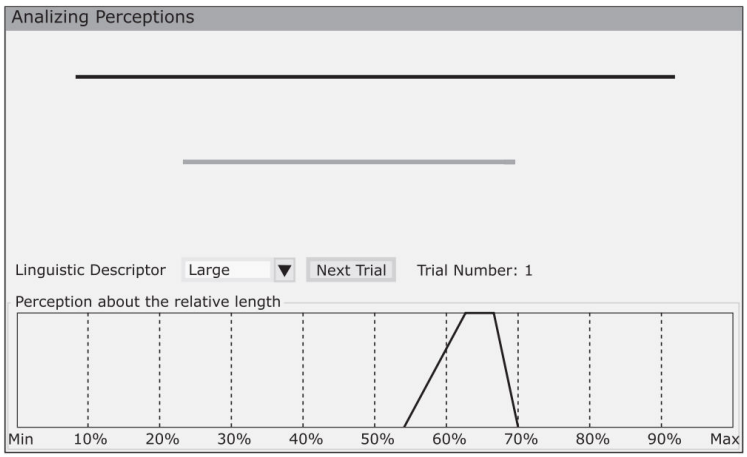


Figure 3.3: Software to evaluate the visual perception of a line segment. This experiment regards your perception about the relative length of different lines. At each trial of the experiment we will show you a black line and you will be asked about its relative length (in comparison with the length of the reference bold line).

of classification accuracy. Besides, we show how to apply the proposed algorithms to analyze interval-valued datasets.

3.7.1 Real-world Datasets

In this section, we briefly introduce the five real-world datasets used in the experiments.

3.7.1.1 Perceptions Experiment Dataset

The 1st dataset, called the perceptions experiment dataset [1], contains 551 observations with one fuzzy feature. The fuzzy feature is a trapezoidal fuzzy number characterized by $(\inf P_0, \sup P_0, \inf P_1, \sup P_1)$. Each observation is the perceptions experiment result for one person.

In the perceptions experiment, the one black line that people will see is shown in Figure 3.3. Once participants see a black line, they will be asked to give a trapezoidal fuzzy number characterized by $(\inf P_0, \sup P_0, \inf P_1, \sup P_1)$ to describe it.

[1] See http://bellman.ciencias.uniovi.es/SMIRE/Perceptions.html for more details.

For the first dataset, we consider using the fuzzy feature (i.e., the trapezoidal fuzzy number) to predict the category (very small; small; medium; large or very large), which will be selected by the participants according to their perception of the black line.

3.7.1.2 Mushroom Dataset

The 2nd dataset is the California mushroom dataset [2] that contains 245 instances in 17 fungi species categories. There are five interval-valued variables: the pileus cap width (X_1), the stipe length (X_2), the stipe thickness (X_3), the spores major axis length (X_4), and the spores minor axis length (X_5). Some instances of the mushroom dataset are shown in Table 3.3. The goal of our experiment on this dataset is to predict the species category of the California mushroom using five interval-valued features.

3.7.1.3 Letter Recognition Dataset

The 3rd dataset is the letter recognition dataset, selected from UCI Machine Learning Repository [3], which contains 20000 instances in 26 categories. This dataset contains 16 integer features extracted from raster scan images of the letters. We use the same methods described in Section 3.6 to transfer integer features into fuzzy features. Then, we obtain one real-world dataset with fuzzy-valued features. The goal of our experiment on this dataset is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet.

3.7.1.4 London Weather Dataset

The 4th dataset is the meteorological data of London (from March 1, 2016 to December 31, 2021), provided by the 'Reliable Prognosis' site [4], which contains 2131 instances. Each instance is meteorological data of one day in London, which described by five

[2] See https://www.mykoweb.com/CAF/ for more details.
[3] See https://archive-beta.ics.uci.edu/ for more details.
[4] See https://rp5.ru/ for more details.

Table 3.3: Some Instances of the Mushroom Dataset

Species	X_1(cm)	X_2(cm)	X_3(cm)	X_4(cm)	X_5(μm)
Agaricus	[6,12]	[2,7]	[1.5,3]	[6,7.5]	[4,5]
Boletus	[7,14]	[5,9]	[3,6]	[11.5,13.5]	[3.5,4.5]
Amanita	[6,12]	[9,17]	[1,2]	[9.5,11.5]	[8.5,10]
Clitocybe	[2,9]	[2,6]	[0.5,1.2]	[5,6]	[2.5,3.5]
Cortinarius	[4,8.5]	[5.5,11]	[0.8,1.5]	[8.8,10.1]	[5.9,6.6]
Entoloma	[5,13]	[5,11]	[1.5,3]	[7,8.5]	[6,7.5]
Gyromitra	[5,10]	[2,8]	[3,7]	[25,35]	[12,16]
Hygrocybe	[2.5,5]	[2.5,5.5]	[0.5,1]	[7,9.5]	[4,5]
Inocybe	[4,8]	[4,8]	[1,2]	[9,12]	[6,7.5]
Lactarius	[4.5,9.8]	[3,5.7]	[1.5,2.5]	[8.5,9.9]	[6.6,7.7]
Marasmius	[1,3]	[1.5,4]	[0.2,0.5]	[9,10]	[3.5,4.5]
Mycena	[0.5,1.5]	[3,7]	[0.1,0.3]	[8,9.5]	[4,5]

interval-valued variables (air temperature T, atmospheric pressure at weather station level $P0$, atmospheric pressure reduced to main sea level P, humidity U and dew-point temperature Td) and one category variable (Precipitation or not: $0 \equiv$ No Precipitation, $1 \equiv$ Precipitation). Some instances of this dataset are shown in Table 3.4. We aim to use the five interval-valued features for precipitation prediction.

3.7.1.5 Washington Weather Dataset

The 5th dataset is the meteorological data of Washington (from January 1, 2016 to December 31, 2021) in the 'Reliable Prognosis' site as well, which contains 2191 instances. Each instance is meteorological data of one day in Washington, which described by five interval-valued variables (same as the 4th dataset) and one category variable (same as the 4th dataset). We aim to use the five interval-valued features for precipitation prediction.

Table 3.4: Some Instances of the London Weather Data

Times	T	P0	P	U	Td	Y
31/12/2021	[0.8,6.1]	[730.2,733.4]	[755.5,759]	[76,99]	[0,3.3]	1
30/12/2021	[-1.4,1.5]	[734.2,735.8]	[759.8,762]	[77,93]	[-2.4,-0.6]	0
29/12/2021	[-1.2,2.1]	[730.5,735.4]	[756,761]	[93,97]	[-2.4,1.7]	1
28/12/2021	[-1.2,1.4]	[730.5,734.2]	[756.1,760]	[72,96]	[-4.2,0.1]	1

3.7.2 Preprocessing of Interval-valued Data

We notice that the features of the 2nd, 4th and 5th datasets are interval-valued. Therefore, in this section, we present an approach to transform interval-valued features into fuzzy-valued features. Suppose $[A,B]$ is denoted as a feature of one interval-valued instance. Thus, we use one approach that maps $[A,B]$ to a triangular fuzzy number $\widetilde{x}$ characterized by $(A,\beta A+(1-\beta)B,B)$, where $\beta\in[0,1]$ is a hyperparameter to control the shape of the membership function of $\widetilde{x}$.

Through the above preprocessing, the DF-SVM and DF-MLP algorithms can be used to classify dataset with interval-valued instances. In addition, we realize that the second dataset is an imbalanced dataset which means that each category contains a different number of instances. Therefore, a random oversampling technique (KMeansSMOTE [74]) is used to improve the performance of the proposed algorithms. After the process of the random oversampling technique, the data of each category in the second dataset is expanded to 30.

3.7.3 Experimental Setup

We use the same baselines in Section 3.6, and the experimental details of all methods are basically the same as in Section 3.6. The only difference is that one more hyperparameter β needs to be selected when analyzing the second dataset. We select the shape parameter β from $\{0,0.05,0.1,\cdots,1\}$. Further, we complete the Wilcoxon rank-sum tests of the

method, which obtains the best performance, with other methods on real-world datasets. Since DCCF and BCCF can not well handle the dataset with a large number of instances, we only compare the proposed algorithms with the first five baselines on the last three datasets in our experiments.

In addition, since the second dataset is an imbalanced dataset, we use balanced accuracy [12] and AUC instead of classification accuracy to compare model performance on the second dataset. The definition of balanced accuracy is

$$\text{Balanced Accuracy} = \frac{1}{K}\sum_{k=1}^{K}(\text{Recall of } k\text{-th class}),$$
$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN}),$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative. AUC is equal to the compute area under the receiver operating characteristic curve.

3.7.4 Experimental Results Analysis

All the experiment results on the five real-world datasets are illustrated in Tables 3.5, 3.6 and how the evaluation metrics varies with the number of epochs for Algorithm 2 are shown in Figure 3.4. From these results, the proposed two algorithms achieve better performance than other baselines on all five real-world datasets, which illustrates the efficacy of the proposed algorithms in addressing real-world datasets with fuzzy-valued or interval-valued features. Moreover, the results of the statistic test show that the proposed two algorithms outperform most other methods significantly at the 0.05 significance level, which demonstrates the superiority of the proposed algorithms. Further, for the 1st, 2nd and 5th datasets, DF-MLP obtains the highest average performance on the test set. While, for the letter recognition dataset and London weather dataset, DF-SVM is more prioritized than other methods, which means that the proposed algorithms are applicable to different types of datasets.

Table 3.5: Experiment Result on Real-world Datasets.

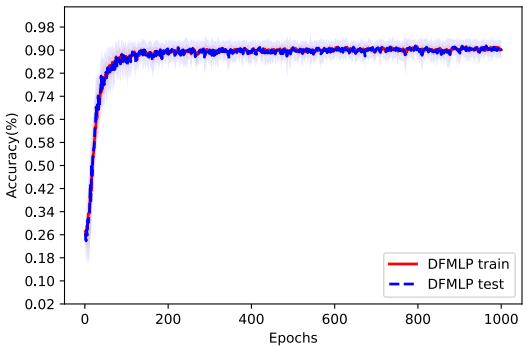
	Perceptions Experiment Dataset		Mushroom Dataset			
Algorithms	Test accuracy	p	Balanced accuracy	p	AUC	p
Meanlogistic	90.04% ±2.20%	0.0080^*	71.36% ±3.86%	$6.3 \times 10^{-8*}$	0.9645 ± 0.0079	0.0012^*
MeanSVM	90.36% ±2.98%	0.5075	79.08% ±3.08%	$3.5 \times 10^{-5*}$	0.9728 ± 0.0071	0.4171
MeanDecisiontree	89.32% ±3.30%	0.0231^*	70.68% ±4.16%	$6.3 \times 10^{-8*}$	0.9069 ± 0.0203	$6.3 \times 10^{-8*}$
MeanRandomForest	90.27% ±3.10%	0.3169	79.04% ±3.83%	0.0002^*	0.9750 ± 0.0077	0.0935
MeanMLP	90.45% ±2.91%	0.3793	80.49% ±3.40%	0.0041^*	0.9721 ± 0.0071	0.6849
DCCF[22]	87.82% ±2.15%	0.0001^*	65.14% ±5.31%	$6.3 \times 10^{-8*}$	0.9584 ± 0.0078	$6.3 \times 10^{-5*}$
BCCF[22]	88.23% ± 2.01%	0.0001^*	64.16% ±4.53%	$6.3 \times 10^{-8*}$	0.9554 ± 0.0083	$6.3 \times 10^{-5*}$
DF-SVM	91.00% ±2.52%	0.7251	81.71% ±4.44%	0.1762	0.9758 ± 0.0103	0.3438
DF-MLP	**91.50%** ± **2.51%**	—-	**83.57%** ± **2.04%**	—-	**0.9784** ± **0.0025**	—

The bold value represents the highest accuracy in each column.
p: The p-value of the Wilcoxon rank-sum test between the best performance and other baselines' outcomes.
$^*p < 0.05$

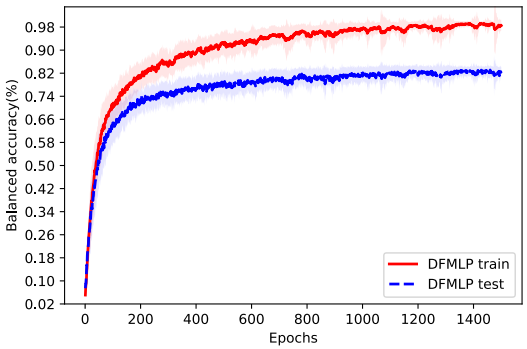
Table 3.6: Experiment Result on Real-world Datasets.

	Letter Recognition Dataset		London Weather Dataset		Washington Weather Dataset	
Algorithms	Test accuracy	p	Test accuracy	p	Test accuracy	p
Meanlogistic	73.50% ±0.70%	6.3×10^{-8}*	71.58% ±1.94%	0.0038*	97.60% ±0.60%	0.045*
MeanSVM	94.60% ±0.36%	0.0011*	72.26% ±2.15%	0.049*	97.76% ±0.66%	0.30
MeanDecisiontree	78.09% ±0.69%	6.3×10^{-8}*	69.11% ±1.99%	1.5×10^{-5}*	97.26% ±0.74%	0.0026*
MeanRandomForest	93.50% ±0.41%	6.3×10^{-8}*	72.76% ±1.84%	0.24	97.34% ±0.74%	0.0043*
MeanMLP	91.79% ±0.47%	6.3×10^{-8}*	71.53% ±2.10%	0.00059*	97.65% ±0.52%	0.049*
DF-SVM	**95.01%** ± **0.32%**	—-	**73.55%** ± **1.73%**	—-	97.95% ±0.66%	0.90
DF-MLP	93.61% ±0.43%	6.3×10^{-8}*	73.06% ±1.91%	0.33	**98.01%** ± **0.62%**	—-

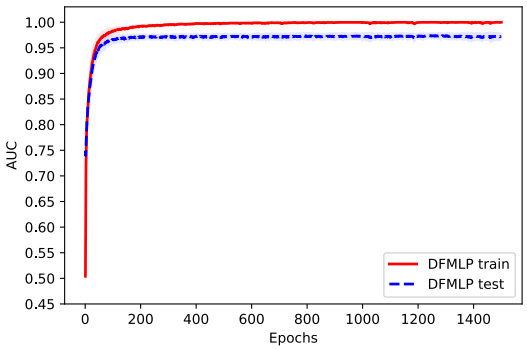
The bold value represents the highest accuracy in each column.
p: The p-value of the Wilcoxon rank-sum test between the best performance and other baselines' outcomes.
*$p < 0.05$



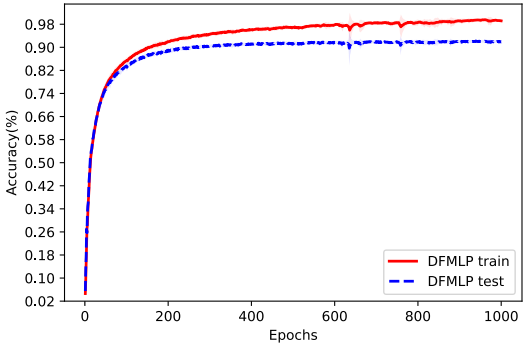
(a) DF-MLP on the perceptions experiment dataset.



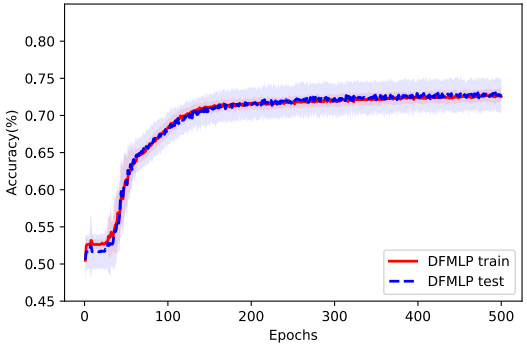
(b) DF-MLP on the mushroom dataset.



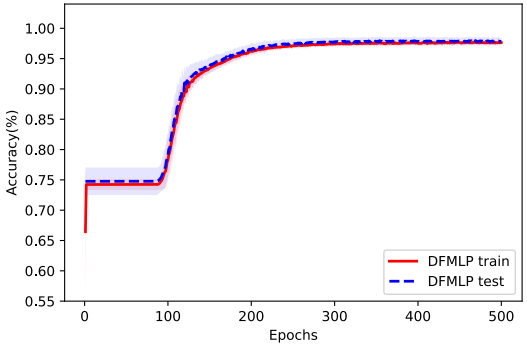
(c) DF-MLP on the mushroom dataset.



(d) DF-MLP on the letter recognition dataset.

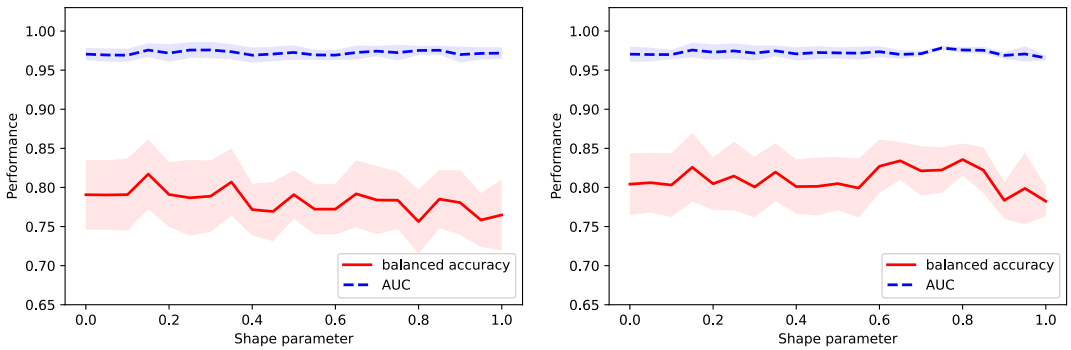


(e) DF-MLP on the London weather dataset.



(f) DF-MLP on the Washington weather dataset.

Figure 3.4: Evaluation metrics varies with the number of epochs.



(a) DF-SVM. (b) DF-MLP.

Figure 3.5: Evaluation metrics of the test sets varies with the value of β.

3.7.5 Parameters Sensitivity Analysis

In this section, we analyze whether the value of the shape parameter β in DF-SVM and DF-MLP affects the balanced accuracy and AUC on the mushroom dataset.

We conduct the same preprocessing for the mushroom dataset. We select the shape parameter β from $\{0, 0.05, 0.1, \cdots, 1\}$. Then, for each value of β, the results are obtained using the same experimental operation in Section 3.6. Figures 3.5(a) and 3.5(b) show the mean and standard deviation of the balanced accuracy and AUC of the test sets on the mushroom dataset when the shape parameter β of both algorithms changes from 0 to 1. These figures illustrate that a different value for the shape parameter β will affect the classification performance since the value of β determines the shape of the triangular fuzzy number. A value of β that can achieve high performance means that the proposed algorithms with this value of β can extract more significant information from the datasets with fuzzy-valued or interval-valued features. Therefore, we can improve the performance of DF-SVM and DF-MLP by finding a suitable value of β. In our experiments, we find the optimal value of β in the validation set.

3.8 Summary

In this chapter, we identify a new problem called MCIMO. In the MCIMO problem, we need to train a fuzzy classifier when only imprecise observations are available.

Firstly, we identify the MCIMO problem in Section 3.3. Since there are no existing papers for theoretical analysis of fuzzy classifiers, we give the estimation error bounds for the MCIMO problem. These bounds illustrate that we can always train a fuzzy classifier with high classification accuracy to solve the MCIMO problem as long as sufficient fuzzy-feature instances can be collected.

Hence, two algorithms are constructed to handle the MCIMO problem. In addition, the optimal defuzzification function for the proposed fuzzy technique-based algorithms is found by comparing the performance of different defuzzification methods on synthetic datasets. Finally, experimental results on synthetic datasets and three real-world datasets show the superiority of the proposed algorithms. Moreover, through comparisons with several non-fuzzy baselines, the experimental results demonstrate that the proposed fuzzy-based methods can obtain better performance in analyzing fuzzy data or interval-valued data than non-fuzzy methods. Since they use fuzzy vectors to express the distribution of imprecise data and apply different defuzzification methods to extract crisp-valued information from imprecise observations.

3.9 Appendix

3.9.1 Preparation for Proving Theorem 3.1

First, we introduce some related definitions.

Definition 3.12. The fuzzy probability of fuzzy real-valued function ℓ mapping from

$\mathscr{F}^p_{\mathbb{R}^p} \times \mathscr{Y}$ to $\mathbb{R}_+$ is defined as:

$$\widetilde{\mathbb{P}}(\ell(\widetilde{\mathbf{X}}) \geq \varepsilon) = \frac{1}{2} \int_{\mathbf{X} \in B} \int_{\alpha \in (0,1]} (f_{\widetilde{\mathbf{X}}_\alpha}(\widetilde{\mathbf{X}}^L_\alpha) + f_{\widetilde{\mathbf{X}}_\alpha}(\widetilde{\mathbf{X}}^U_\alpha)) d\mathbf{X} d\alpha,$$

where $B = \{\mathbf{X} \in \mathbb{R}^p \,|\, \ell(\widetilde{\mathbf{X}}) \geq \varepsilon\}$.

According to definition 3.8, the fuzzy Markov's inequality is denoted as:

$$\widetilde{\mathbb{P}}(\ell(\widetilde{\mathbf{X}}) \geq \varepsilon) = \frac{1}{2} \int_{\mathbf{X} \in B} \int_{\alpha \in (0,1]} e^{-t\ell(\widetilde{\mathbf{X}})} e^{t\ell(\widetilde{\mathbf{X}})} (f_{\widetilde{\mathbf{X}}_\alpha}(\widetilde{\mathbf{X}}^L_\alpha) + f_{\widetilde{\mathbf{X}}_\alpha}(\widetilde{\mathbf{X}}^U_\alpha)) d\mathbf{X} d\alpha \leq e^{-t\varepsilon} \mathbb{E}_{\widetilde{\mathbf{X}} \sim \widetilde{\mathscr{D}}}[e^{t\ell(\widetilde{\mathbf{X}})}].$$

Definition 3.13. A sequence of $V_1, V_2, \cdots$ is a martingale difference sequence with respect to fuzzy random vectors $\widetilde{\mathbf{X}}_1, \widetilde{\mathbf{X}}_2, \cdots$ if for all $i > 0$, V_i is a real-value function of $\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_i$ and $\mathbb{E}_{\widetilde{\mathbf{X}}_i \sim \widetilde{\mathscr{D}}}[V_{i+1} | \widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_i] = 0$, where

$$\begin{aligned} \mathbb{E}_{\widetilde{\mathbf{X}}_i \sim \widetilde{D}}[V_{i+1} | \widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_i] = \frac{1}{2} \int_{\alpha \in (0,1]} \{ & \mathbb{E}_{\widetilde{\mathbf{X}^L_{i+1\alpha}} \sim \mathscr{D}^L_\alpha}[V_{i+1}(\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_{i+1})] \\ & + \mathbb{E}_{\widetilde{\mathbf{X}^U_{i+1\alpha}} \sim \mathscr{D}^U_\alpha}[V_{i+1}(\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_{i+1})] \} d\alpha. \end{aligned}$$

Next, we introduce one lemma and two theorems to prove Theorem 3.1.

Lemma 3.2. *Let $V_1, V_2, \cdots$ be a martingale difference sequence with respect to the fuzzy random variables $\widetilde{\mathbf{X}}_1, \widetilde{\mathbf{X}}_2, \cdots$ and assume that for all $i > 0$ there is a constant $c_i \geq 0$ and fuzzy random variable Z_i, which is a real-value function of $\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_{i-1}$, that satisfy*

$$Z_i \leq V_i \leq Z_i + c_i.$$

Then, for all $t > 0$, the following upper bound holds:

$$\mathbb{E}_{\widetilde{\mathbf{X}}_i \sim \widetilde{\mathscr{D}}}[e^{tV_i} | \widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_{i-1}] \leq e^{t^2 c_i^2/8}. \tag{3.23}$$

Proof. By the convexity of $x \to e^x$, for all $x \in [a, b]$, the following holds:

$$e^{tx} \leq \tfrac{b-x}{b-a} e^{ta} + \tfrac{x-a}{b-a} e^{tb}.$$

Thus, using $\mathbb{E}_{\widetilde{\mathbf{X}}_i\sim\widetilde{\mathscr{D}}}[V_{i+1}|\widetilde{\mathbf{X}}_1,\cdots,\widetilde{\mathbf{X}}_i]=0$, then

$$\begin{aligned}\mathbb{E}_{\widetilde{\mathbf{X}}_i\sim\widetilde{\mathscr{D}}}[e^{tV_{i+1}}|\widetilde{\mathbf{X}}_1,\cdots,\widetilde{\mathbf{X}}_i] &\leq \mathbb{E}_{\widetilde{\mathbf{X}}_i\sim\widetilde{\mathscr{D}}}[\frac{Z_{i+1}+c_{i+1}-V_{i+1}}{c_{i+1}}e^{tZ_{i+1}}+\frac{V_{i+1}-Z_{i+1}}{c_{i+1}}e^{t(Z_{i+1}+c_{i+1})}|\widetilde{\mathbf{X}}_1,\cdots,\widetilde{\mathbf{X}}_i]\\ &= \frac{Z_{i+1}+c_{i+1}}{c_{i+1}}e^{tZ_{i+1}}+\frac{-Z_{i+1}}{c_{i+1}}e^{t(Z_{i+1}+c_{i+1})}\\ &\leq e^{t^2c_{i+1}^2/8},\end{aligned}$$

which completes the proof. ■

Theorem 3.4 (Fuzzy Azuma's Inequality)**.** *Let $V_1,V_2,\cdots$ be a martingale difference sequence with respect to the fuzzy random variables $\widetilde{\mathbf{X}}_1,\widetilde{\mathbf{X}}_2,\cdots$ and assume that for all $i>0$ there is a constant $c_i\geq 0$ and fuzzy random variable Z_i, which is a real-value function of $\widetilde{\mathbf{X}}_1,\cdots,\widetilde{\mathbf{X}}_{i-1}$, that satisfies*

$$Z_i\leq V_i\leq Z_i+c_i.$$

Then for all $\varepsilon>0$ and $m\in N^+$, the following inequalities hold:

$$\begin{aligned}\widetilde{\mathbb{P}}[\sum_{i=1}^{m}V_i\geq\varepsilon]&\leq\exp\frac{-2\varepsilon^2}{\sum_{i=1}^{m}c_i^2},\\ \widetilde{\mathbb{P}}[\sum_{i=1}^{m}V_i\leq-\varepsilon]&\leq\exp\frac{-2\varepsilon^2}{\sum_{i=1}^{m}c_i^2}.\end{aligned}\tag{3.24}$$

Proof. Let $S_k=\sum_{i=1}^{k}V_i$. Then, using fuzzy Markov's inequality, for any $t>0$, we can write

$$\begin{aligned}\widetilde{\mathbb{P}}[S_m\geq\varepsilon] &\leq e^{-t\varepsilon}\mathbb{E}_{\widetilde{\mathbf{X}}_i\sim\widetilde{\mathscr{D}}}[e^{tS_m}]\\ &= e^{-t\varepsilon}\mathbb{E}_{\widetilde{\mathbf{X}}_i\sim\widetilde{\mathscr{D}}}[e^{tS_{m-1}}\mathbb{E}_{\widetilde{\mathbf{X}}_i\sim\widetilde{\mathscr{D}}}[e^{tV_m}|\widetilde{\mathbf{X}}_1,\widetilde{\mathbf{X}}_2,\cdots,\widetilde{\mathbf{X}}_{m-1}]]\\ &\leq e^{-t\varepsilon}\mathbb{E}_{\widetilde{\mathbf{X}}_i\sim\widetilde{\mathscr{D}}}[e^{tS_{m-1}}]e^{t^2c_m^2/8}\ \text{(iterating previous argument)}\\ &\leq e^{-t\varepsilon}e^{t^2\sum_{i=1}^{m}c_i^2/8}\ \ (\text{let } t=4\varepsilon/\sum_{i=1}^{m}c_i^2)\\ &= e^{\frac{-2\varepsilon^2}{\sum_{i=1}^{m}c_i^2}},\end{aligned}$$

the second statement is shown in a similar way. ■

Theorem 3.5 (Fuzzy McDiarmid's Inequality)**.** *Let* $\widetilde{\mathbf{X}}_1, \widetilde{\mathbf{X}}_2, \cdots, \widetilde{\mathbf{X}}_m \in \widetilde{\mathscr{X}} \subset \mathscr{F}^p_{\mathbb{R}^p} \sim \mathscr{D}$ *be a set of* $m \geq 1$ *independent fuzzy random vectors and assume that there exist* $c_1, c_2, \cdots, c_m > 0$ *such that* $f : \widetilde{\mathscr{X}}^m \triangleq \widetilde{\mathscr{X}} \times \cdots \times \widetilde{\mathscr{X}} \to \mathbb{R}$ *satisfies the following conditions:*

$$|f(\widetilde{\mathbf{x}}_1, \cdots, \widetilde{\mathbf{x}}_i, \cdots, \widetilde{\mathbf{x}}_m) - f(\widetilde{\mathbf{x}}_1, \cdots, \widetilde{\mathbf{x}}_i', \cdots, \widetilde{\mathbf{x}}_m)| \leq c_i,$$

for all $i \in [1, m]$ *and any points* $\widetilde{\mathbf{x}}_1, \cdots, \widetilde{\mathbf{x}}_i, \cdots, \widetilde{\mathbf{x}}_m, \widetilde{\mathbf{x}}_i' \in \widetilde{\mathscr{X}}$*. Let* $f(\widetilde{S})$ *denote* $f(\widetilde{\mathbf{X}}_1, \widetilde{\mathbf{X}}_2, \cdots, \widetilde{\mathbf{X}}_m)$*, then, for all* $\varepsilon > 0$*, the following inequalities hold:*

$$\begin{aligned} \widetilde{\mathbb{P}}[f(\widetilde{S}) - \mathbb{E}_{\widetilde{S} \sim \widetilde{\mathscr{D}}^m}[f(\widetilde{S})] \geq \varepsilon] &\leq \exp \frac{-2\varepsilon^2}{\sum_{i=1}^{m} c_i^2}, \\ \widetilde{\mathbb{P}}[f(\widetilde{S}) - \mathbb{E}_{\widetilde{S} \sim \widetilde{\mathscr{D}}^m}[f(\widetilde{S})] \leq -\varepsilon] &\leq \exp \frac{-2\varepsilon^2}{\sum_{i=1}^{m} c_i^2}. \end{aligned} \tag{3.25}$$

Proof. Define a sequence of random variables $V_k, k \in [1, m]$, as follows:

$$\begin{aligned} V &= f(\widetilde{S}) - \mathbb{E}_{\widetilde{S} \sim \widetilde{\mathscr{D}}^m}[f(\widetilde{S})], \\ V_1 &= \mathbb{E}_{\widetilde{S}}[V|\widetilde{\mathbf{X}}_1] - \boldsymbol{E}_{\widetilde{S}}[V], \\ V_k &= \mathbb{E}_{\widetilde{S}}[V|\widetilde{\mathbf{X}}_1, \widetilde{\mathbf{X}}_2, \cdots, \widetilde{\mathbf{X}}_k] - \mathbb{E}_{\widetilde{S}}[V|\widetilde{\mathbf{X}}_1, \widetilde{\mathbf{X}}_2, \cdots, \widetilde{\mathbf{X}}_{k-1}]. \end{aligned}$$

Note that $V = \sum_{i=1}^{m} V_i$. Furthermore, the fuzzy random vector $\mathbb{E}_{\widetilde{S}}[V|\widetilde{\mathbf{X}}_1, \widetilde{\mathbf{X}}_2, \cdots, \widetilde{\mathbf{X}}_k]$ is a function of $\widetilde{\mathbf{X}}_1, \widetilde{\mathbf{X}}_2, \cdots, \widetilde{\mathbf{X}}_k$, therefore:

$$\mathbb{E}_{\widetilde{S}}[\mathbb{E}_{\widetilde{S}}[V|\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_k]|\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_{k-1}] = \mathbb{E}_{\widetilde{S}}[V|\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_{k-1}],$$

which implies $\mathbb{E}_{\widetilde{S}}[V_k|\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_{k-1}] = 0$. Thus, the sequence $(V_k), k \in [1, m]$ is a martingale difference sequence. Next, observe that, since $\mathbb{E}_{\widetilde{S}}[f(\widetilde{S})]$ is a scalar, V_k can be expressed as follows:

$$V_k = \mathbb{E}_{\widetilde{S}}[f(\widetilde{S})|\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_k] - \mathbb{E}_{\widetilde{S}}[f(\widetilde{S})|\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_{k-1}].$$

Thus, we can define an upper bound W_k and lower bound U_k for V_k by:

$$\begin{aligned} W_k &= \sup_{\widetilde{\mathbf{x}}} \mathbb{E}_{\widetilde{S}}[f(\widetilde{S})|\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_k, \widetilde{\mathbf{x}}] - \mathbb{E}_{\widetilde{S}}[f(\widetilde{S})|\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_k], \\ U_k &= \inf_{\widetilde{\mathbf{x}}'} \mathbb{E}_{\widetilde{S}}[f(\widetilde{S})|\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_k, \widetilde{\mathbf{x}}'] - \mathbb{E}_{\widetilde{S}}[f(\widetilde{S})|\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_k], \end{aligned}$$

$$
\begin{aligned}
W_k - U_k &= \sup_{\widetilde{\mathbf{x}},\widetilde{\mathbf{x}}'}\{\mathbb{E}_{\widetilde{S}}[f(\widetilde{S})|\widetilde{\mathbf{X}}_1,\cdots,\widetilde{\mathbf{X}}_k,\widetilde{\mathbf{x}}] - \mathbb{E}_{\widetilde{S}}[f(\widetilde{S})|\widetilde{\mathbf{X}}_1,\cdots,\widetilde{\mathbf{X}}_k,\widetilde{\mathbf{x}}']\} \\
&\leq \frac{1}{2}\sup_{\widetilde{\mathbf{x}},\widetilde{\mathbf{x}}',\alpha}\{\mathbb{E}_{(\widetilde{\mathbf{X}_{k+1\alpha}}^{L},\cdots,\widetilde{\mathbf{X}_{m\alpha}}^{L})\sim(\mathscr{D}_\alpha^L)^{m-k}}[|f(\widetilde{S}_1) - f(\widetilde{S}_2)|] + \mathbb{E}_{(\widetilde{\mathbf{X}_{k+1\alpha}}^{U},\cdots,\widetilde{\mathbf{X}_{m\alpha}}^{U})\sim(\mathscr{D}_\alpha^U)^{m-k}}[|f(\widetilde{S}_1) - f(\widetilde{S}_2)|]\} \\
&\leq c_k,
\end{aligned}
$$

where $\widetilde{S}_1 = (\widetilde{\mathbf{X}}_1,\cdots,\widetilde{\mathbf{X}}_k,\widetilde{\mathbf{x}},\widetilde{\mathbf{X}}_{k+1},\cdots,\widetilde{\mathbf{X}}_m), \widetilde{S}_2 = (\widetilde{\mathbf{X}}_1,\cdots,\widetilde{\mathbf{X}}_k,\widetilde{\mathbf{x}}',\widetilde{\mathbf{X}}_{k+1},\cdots,\widetilde{\mathbf{X}}_m)$.

Thus, $U_k \leq V_k \leq W_k \leq U_k + c_k$. In the view of these inequalities, we can apply Theorem 3.4 to $V = \sum_{i=1}^{m} V_i$, which yields the result. ■

3.9.2 Proof of Theorem 3.1

We are now ready to proof Theorem 3.1.

Proof. Let $\widetilde{z}_i = (\widetilde{\mathbf{X}}_i, y_i)$. For any sample $\widetilde{S} = (\widetilde{z}_1,\widetilde{z}_2,\cdots,\widetilde{z}_m) \sim \widetilde{\mathscr{D}}^m$ and any $\ell \in \mathscr{L}_{\mathscr{H}}$, we denote

$$
\Phi(\widetilde{S}) = \sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})] - \frac{1}{m}\sum_{i=1}^{m}\ell(\widetilde{z}_i)\} = \sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})] - \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]\}.
$$

Let $\widetilde{S}$ and $\widetilde{S}'$ be two samples differing by exactly one point, say $\widetilde{z}_m$ in $\widetilde{S}$ and $\widetilde{z}_m'$ in $\widetilde{S}'$. Then, since the difference of suprema does not exceed the supremum of the difference, we have

$$
\Phi(\widetilde{S}') - \Phi(\widetilde{S}) \leq \sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})] - \widehat{\mathbb{E}}_{\widetilde{S}'}[\ell(\widetilde{z})]\} \leq \sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\frac{\ell(\widetilde{z}_m) - \ell(\widetilde{z}_m')}{m} \leq \frac{2C_l}{m}.
$$

Similarly, we can obtain $\Phi(\widetilde{S}) - \Phi(\widetilde{S}') \leq \frac{2C_l}{m}$, thus $|\Phi(\widetilde{S}') - \Phi(\widetilde{S})| \leq \frac{2C_l}{m}$. Then, by fuzzy McDiarmid's inequality, for any $\delta > 0$, with fuzzy probability at least $1 - \delta/2$, the following holds:

$$
\Phi(\widetilde{S}) \leq \mathbb{E}_{\widetilde{S}}[\Phi(\widetilde{S})] + C_l\sqrt{\frac{2\log(2/\delta)}{m}},
$$

$$
\mathbb{E}_{\widetilde{S}}[\Phi(\widetilde{S})] = \mathbb{E}_{\widetilde{S}}[\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})] - \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]\}] = \mathbb{E}_{\widetilde{S}}[\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\mathbb{E}_{\widetilde{S}'}\{\widehat{\mathbb{E}}_{\widetilde{S}'}[\ell(\widetilde{z}')] - \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]\}].
$$

Using the fact that points in $\widetilde{S}'$ are sampled in an i.i.d. fashion and thus

$$\begin{aligned}\mathbb{E}_{\widetilde{S}'}\{\widehat{\mathbb{E}}_{\widetilde{S}'}[\ell(\widetilde{z}')]\} &= \frac{1}{2}\int_{\alpha\in(0,1]}\{\mathbb{E}_{(\mathscr{D}_\alpha^L)^m}[\frac{1}{m}\sum_{i=1}^m \ell(\widetilde{z}_i')] + \mathbb{E}_{(\mathscr{D}_\alpha^U)^m}[\frac{1}{m}\sum_{i=1}^m \ell(\widetilde{z}_i')]\}d\alpha \\ &= \frac{1}{2}\int_{\alpha\in(0,1]}\{\frac{1}{m}\sum_{i=1}^m \mathbb{E}_{\mathscr{D}_\alpha^L}[\ell(\widetilde{z}_i')] + \frac{1}{m}\sum_{i=1}^m \mathbb{E}_{\mathscr{D}_\alpha^U}[\ell(\widetilde{z}_i')]\}d\alpha \\ &= \frac{1}{2}\int_{\alpha\in(0,1]}\{\mathbb{E}_{\mathscr{D}_\alpha^L}[\ell(\widetilde{z})] + \mathbb{E}_{\mathscr{D}_\alpha^U}[\ell(\widetilde{z})]\}d\alpha = \mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})].\end{aligned}$$

Because

$$\begin{aligned}&\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\mathbb{E}_{\widetilde{S}'}\{\widehat{\mathbb{E}}_{\widetilde{S}'}[\ell(\widetilde{z}')] - \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]\} \\ =\ &\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\tfrac{1}{2}\int_{\alpha\in(0,1]}\{\mathbb{E}_{(\widetilde{S}')_\alpha^L\sim(\mathscr{D}_\alpha^L)^m}[\widehat{\mathbb{E}}_{\widetilde{S}'}[\ell(\widetilde{z}')] - \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]] + \mathbb{E}_{(\widetilde{S}')_\alpha^U\sim(\mathscr{D}_\alpha^U)^m}[\widehat{\mathbb{E}}_{\widetilde{S}'}[\ell(\widetilde{z}')] - \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]]\}d\alpha\} \\ \le\ &\tfrac{1}{2}\int_{\alpha\in(0,1]}\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\mathbb{E}_{(\widetilde{S}')_\alpha^L\sim(\mathscr{D}_\alpha^L)^m}[\widehat{\mathbb{E}}_{\widetilde{S}'}[\ell(\widetilde{z}')] - \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]] + \mathbb{E}_{(\widetilde{S}')_\alpha^U\sim(\mathscr{D}_\alpha^U)^m}[\widehat{\mathbb{E}}_{\widetilde{S}'}[\ell(\widetilde{z}')] - \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]]\}d\alpha \\ \le\ &\tfrac{1}{2}\int_{\alpha\in(0,1]}\{\mathbb{E}_{(\widetilde{S}')_\alpha^L\sim(\mathscr{D}_\alpha^L)^m}\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}[\widehat{\mathbb{E}}_{\widetilde{S}'}[\ell(\widetilde{z}')] - \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]] + \mathbb{E}_{(\widetilde{S}')_\alpha^U\sim(\mathscr{D}_\alpha^U)^m}\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}[\widehat{\mathbb{E}}_{\widetilde{S}'}[\ell(\widetilde{z}')] - \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]]\}d\alpha \\ =\ &\mathbb{E}_{\widetilde{S}'}\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\widehat{\mathbb{E}}_{\widetilde{S}'}[\ell(\widetilde{z}')] - \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]\}.\end{aligned}$$

Then, we have

$$\mathbb{E}_{\widetilde{S}}[\Phi(\widetilde{S})] \le \mathbb{E}_{\widetilde{S},\widetilde{S}'}\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\widehat{\mathbb{E}}_{\widetilde{S}'}[\ell(\widetilde{z}')] - \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]\} = \mathbb{E}_{\widetilde{S},\widetilde{S}'}\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\frac{1}{m}\sum_{i=1}^m[\ell(\widetilde{z}_i') - \ell(\widetilde{z}_i)]\}.$$

We introduce Rademacher variables σ_is, that are uniformly distributed independent random variables taking values in $\{-1,+1\}$,

$$\begin{aligned}\mathbb{E}_{\widetilde{S}}[\Phi(\widetilde{S})] &\le \mathbb{E}_{\widetilde{S},\widetilde{S}'}\mathbb{E}_{\boldsymbol{\sigma}}\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\frac{1}{m}\sum_{i=1}^m[\sigma_i\ell(\widetilde{z}_i') - \ell(\widetilde{z}_i)]\}\ \ (\sup(U+V)\le\sup U+\sup V) \\ &\le \mathbb{E}_{\widetilde{S}'}\mathbb{E}_{\boldsymbol{\sigma}}\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\frac{1}{m}\sum_{i=1}^m\sigma_i\ell(\widetilde{z}_i')\} + \mathbb{E}_{\widetilde{S}}\mathbb{E}_{\boldsymbol{\sigma}}\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\frac{1}{m}\sum_{i=1}^m-\sigma_i\ell(\widetilde{z}_i)\}.\end{aligned}$$

Because the definition of fuzzy Rademacher complexity and the fact that the variables σ_i and $-\sigma_i$ are distributed in the same way. Then

$$\mathbb{E}_{\widetilde{S}}[\Phi(\widetilde{S})] \le 2\mathbb{E}_{\widetilde{S}}\mathbb{E}_{\boldsymbol{\sigma}}\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\tfrac{1}{m}\sum_{i=1}^m\sigma_i\ell(\widetilde{z}_i)\} = 2\widetilde{\mathscr{R}}_{\widetilde{S}\sim\widetilde{\mathscr{D}}^m}(\mathscr{L}_{\mathscr{H}}).$$

Then using δ instead of $\delta/2$, with fuzzy probability $1-\delta$ the following holds :

$$\begin{aligned}\Phi(\widetilde{S}) &\leq 2\widetilde{\mathscr{R}}_{\widetilde{S}\sim\widetilde{\mathscr{D}}^m}(\mathscr{L}_{\mathscr{H}}) + C_l\sqrt{\frac{2\log(1/\delta)}{m}} \\ \mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})] - \frac{1}{m}\sum_{i=1}^{m}\ell(\widetilde{z}_i) &\leq 2\widetilde{\mathscr{R}}_{\widetilde{S}\sim\widetilde{\mathscr{D}}^m}(\mathscr{L}_{\mathscr{H}}) + C_l\sqrt{\frac{2\log(1/\delta)}{m}}.\end{aligned} \tag{3.26}$$

We observe that changing one point in $\widetilde{S}$ changes $\widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{H}})$ by at most $2C_l/m$. Then, again using fuzzy McDiarmid's inequality, with fuzzy probability $1-\delta/2$ the following holds:

$$\widetilde{\mathscr{R}}_{\widetilde{S}\sim\widetilde{\mathscr{D}}^m}(\mathscr{L}_{\mathscr{H}}) \leq \widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{H}}) + C_l\sqrt{\frac{2\log(2/\delta)}{m}}.$$

Then with probability at least $1-\delta$:

$$\begin{aligned}\Phi(\widetilde{S}) &\leq 2\widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{H}}) + 3C_l\sqrt{\frac{2\log(2/\delta)}{m}} \\ \mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})] - \frac{1}{m}\sum_{i=1}^{m}\ell(\widetilde{z}_i) &\leq 2\widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{H}}) + 3C_l\sqrt{\frac{2\log(2/\delta)}{m}}.\end{aligned} \tag{3.27}$$

Since l is Lipschitz continuous, according to [101], we have

$$\begin{aligned}\widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{H}}) &\leq \sqrt{2}L_l\widehat{\mathscr{R}}_{\widetilde{S}_X}(\mathscr{H}) \\ \widetilde{\mathscr{R}}_{\widetilde{S}\sim\widetilde{\mathscr{D}}^m}(\mathscr{L}_{\mathscr{H}}) &\leq \sqrt{2}L_l\widetilde{\mathscr{R}}_{\widetilde{S}_X\sim\widetilde{\mathscr{D}}^m}(\mathscr{H}).\end{aligned} \tag{3.28}$$

Next we let,

$$\Psi(\widetilde{S}) = \inf_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})] - \frac{1}{m}\sum_{i=1}^{m}l(\widetilde{z}_i)\} = -\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{-\mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})] + \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]\}.$$

In the same way, we can obtain $|\Psi(\widetilde{S}') - \Psi(\widetilde{S})| \leq \frac{2C_l}{m}$ Then, by fuzzy McDiarmid's inequality, for any $\delta > 0$, with fuzzy probability at least $1-\delta/2$, the following holds:

$$\Psi(\widetilde{S}) \geq \mathbb{E}_{\widetilde{S}}[\Psi(\widetilde{S})] - C_l\sqrt{\frac{2\log(2/\delta)}{m}}$$

$$\mathbb{E}_{\widetilde{S}}[\Psi(\widetilde{S})] = -\mathbb{E}_{\widetilde{S}}[\sup_{\ell\in\mathscr{L}_{\mathscr{H}}}\{-\mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})] + \widehat{\mathbb{E}}_{\widetilde{S}}[\ell(\widetilde{z})]\}] \geq -2\widetilde{\mathscr{R}}_{\widetilde{S}\sim\widetilde{\mathscr{D}}^m}(\mathscr{L}_{\mathscr{H}})$$

$$\mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})] - \frac{1}{m}\sum_{i=1}^{m}\ell(\widetilde{z}_i) \geq \inf_{\ell\in\mathscr{L}_{\mathscr{H}}}\{\mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})] - \frac{1}{m}\sum_{i=1}^{m}\ell(\widetilde{z}_i)\} \geq -2\widetilde{\mathscr{R}}_{\widetilde{S}\sim\widetilde{\mathscr{D}}^m}(\mathscr{L}_{\mathscr{H}}) - C_l\sqrt{\frac{2\log(2/\delta)}{m}}.$$

Then, with fuzzy probability at least $1-\delta$:

$$\begin{aligned}\mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})] - \frac{1}{m}\sum_{i=1}^{m}\ell(\widetilde{z}_i) &\geq -2\widetilde{\mathscr{R}}_{\widetilde{S}\sim\widetilde{\mathscr{D}}^m}(\mathscr{L}_{\mathscr{H}}) - C_l\sqrt{\frac{2\log(1/\delta)}{m}} \\ \mathbb{E}_{\widetilde{z}\sim\widetilde{\mathscr{D}}}[\ell(\widetilde{z})] - \frac{1}{m}\sum_{i=1}^{m}\ell(\widetilde{z}_i) &\geq -2\widehat{\mathscr{R}}_{\widetilde{S}}(\mathscr{L}_{\mathscr{H}}) - 3C_l\sqrt{\frac{2\log(2/\delta)}{m}}.\end{aligned} \tag{3.29}$$

Following from Inequalities (3.26), (3.27), (3.29) and for any $\delta > 0$, with fuzzy probability at least $1-\delta$, each of the following holds for all $\ell \in \mathscr{L}_{\mathscr{H}}$:

$$
\begin{aligned}
|\mathbb{E}_{\tilde{z}\sim\tilde{\mathscr{D}}}[\ell(\tilde{z})] - \tfrac{1}{m}\sum_{i=1}^{m}\ell(\tilde{z}_i)| &\leq 2\widetilde{\mathscr{R}}_{\tilde{S}\sim\tilde{\mathscr{D}}^m}(\mathscr{L}_{\mathscr{H}}) + C_l\sqrt{\tfrac{2\log(1/\delta)}{m}} \\
|\mathbb{E}_{\tilde{z}\sim\tilde{\mathscr{D}}}[\ell(\tilde{z})] - \tfrac{1}{m}\sum_{i=1}^{m}\ell(\tilde{z}_i)| &\leq 2\widehat{\mathscr{R}}_{\tilde{S}}(\mathscr{L}_{\mathscr{H}}) + 3C_l\sqrt{\tfrac{2\log(2/\delta)}{m}}.
\end{aligned}
\tag{3.30}
$$

Using

$$R(\ell(h(\tilde{\mathbf{X}}), y)) = \mathbb{E}_{\tilde{X}\sim\tilde{\mathscr{D}}}[\ell(h(\tilde{\mathbf{X}}), y)],$$

and Inequalities (3.28) and (3.30), we have for any $\delta > 0$, with fuzzy probability at least $1-\delta$, each of the following holds for all $\ell \in \mathscr{L}_{\mathscr{H}}$:

$$
\begin{aligned}
|R_{\tilde{\mathscr{D}}}(h) - \widehat{R}_{\tilde{\mathscr{D}}}(h)| &\leq 2\sqrt{2}L_l\widetilde{\mathscr{R}}_{\tilde{S}_{\mathbf{X}}}(\mathscr{H}) + C_l\sqrt{\tfrac{2\log(1/\delta)}{m}} \\
|R_{\tilde{\mathscr{D}}}(h) - \widehat{R}_{\tilde{\mathscr{D}}}(h)| &\leq 2\sqrt{2}L_l\widehat{\mathscr{R}}_{\tilde{S}_{\mathbf{X}}}(\mathscr{H}) + 3C_l\sqrt{\tfrac{2\log(2/\delta)}{m}}.
\end{aligned}
$$

■

CHAPTER 4

MULTI-VIEW CLASSIFICATION THROUGH LEARNING FROM INTERVAL-VALUED DATA

4.1 Introduction

Interval-valued data [8] is a common type of data where all of the observations' features are described by intervals, not crisp-valued numbers. For example, Fig. 4.1 shows the visualization of some interval-valued data, where each rectangle represents a two-dimensional interval-valued data. Moreover, the data extracted by many measuring devices are not exact numbers but intervals because there are only a limited number of decimals available on most of these measuring devices. Recently, some researchers have begun exploring imprecise data from different perspectives, such as superset label learning and data disambiguation [85]. Unfortunately, the existing research related to interval-valued data mainly focuses on clustering analysis [28], regression analysis [151], and feature selection [111], yet less on classification tasks [145]. Besides, limited research on interval-valued classification only gives some simple framework and no relevant experimental analysis on real-world interval-valued datasets.

In this chapter, we consider a more specificial situation of the MCIMO problem called *learning from interval-valued data* (LIND), where we aim to learn a classifier that can obtain high classification accuracy on interval-valued observations. Throughout existing research involving interval-valued data, no research discusses a theory regarding the interval-valued data classification problem. To fill this gap, we first present theoretical analysis to obtain the estimation error bound of the LIND problem based on Rademacher complexity (Theorem 4.1). This Rademacher complexity-based bound demonstrates that we can always train a well-performed classifier to address LIND problems when enough interval-valued instances can be collected. Next, we discuss the learnability of the underlying problem with perfect observations (Theorem 4.2). Finally, we provide two theorems to show the strengths of multi-view learning in addressing classification problems (Theorems 4.3 and 4.4). These theorems inspire us to propose a new algorithm called the *multi-view interval information extraction* (Mv-IIE) approach using multi-view learning [9, 172].

The proposed algorithm, which comprises two main parts (Figure 4.2), applies multi-view learning to classify multi-view information extracted from the interval-valued observations. The *first part* is used to extract crisp-valued information from the interval-valued observations. The most commonly used method is taking the intervals' midpoint to extract crisp-valued information. However, using this method will result in losing a lot of critical information from the intervals. For example, suppose we have two intervals $\bar{x}_1 = [1,5]$ and $\bar{x}_2 = [2,4]$, we will obtain the same crisp-valued information $x = 3$ from different intervals by taking the midpoint of the intervals. However, $\bar{x}_1$ clearly has a larger interval than $\bar{x}_2$ has, thus it is improper to consider them as the same instance in the view of the midpoint. Therefore, in this chapter, we propose a membership function-based method [30, 109] to extract multi-view information (more details and motivation are discussed in Section 4.4). The *second part* is a multi-view classifier to handle the

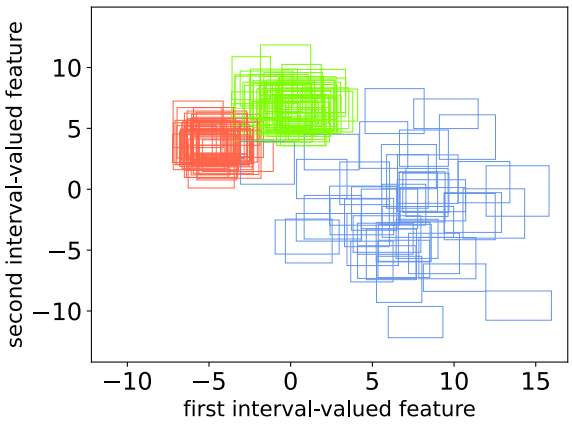


Figure 4.1: Visualization of some interval-valued data.

extracted multi-view information. In this chapter, SVM, random forests and neural networks are used as the basic structures of the multi-view classifier. This multi-view classifier guided by the proposed theorems is trained on the view-fusion representation vectors constructed by integrating an appropriate number of candidate views.

Finally, we compare the performance of the Mv-IIE algorithm with several baselines on both synthetic and real-world datasets. The experiment results illustrate the superiority of the proposed model in handling interval-valued data. Moreover, we detail an application of Mv-IIE that we present a novel framework for protecting data privacy called *interval privacy-preserving* (INPP). Through experiments on one real-world dataset, it demonstrates that applying INPP can prevent raw (crisp-valued) data leakage while ensuring high performance.

The main contributions of this chapter are as follows.

1. A new challenging problem with interval-valued observations called LIND has been identified that is different from most existing classification problems with crisp-valued observations. The estimation error bounds of the LIND problem based on Rademacher complexity is provided, which ensures that we can always train a classifier with high classification accuracy. The learnability of the underlying

problem with perfect observation is also discussed. This is the first work to give these theoretical analysis of interval-valued data.

2. To solve the LINO problem, a new algorithm called Mv-IIE is developed by using multi-view learning. The theoretical analysis to demonstrate the motivation for our algorithm construction is also provided. Experimental comparisons with several baselines on both synthetic and real-world datasets demonstrate the superiority of Mv-IIE for interval-valued data classification.

3. A novel framework for protecting data privacy called INPP is presented to show an application of Mv-IIE. This is the first paper to consider utilizing the property of interval-valued data to realize data privacy protection. Experimental results on one real-world dataset show that applying INPP can prevent crisp-valued data leakage while ensuring high classification accuracy.

4.2 Problem Setting

In this section, we introduce the problem of learning from interval-valued data.

Let $\bar{\mathbf{x}} = (\bar{x}_1, \cdots, \bar{x}_p)^\top$ be a p-dimension interval-valued vector, where $\bar{x}_j = [x_j^{\mathrm{l}}, x_j^{\mathrm{r}}], j \in [p]$. Here, we denote $[p] = \{1, \cdots, p\}$. $\bar{\mathbb{R}}$ is denoted as the set of all real-valued intervals (closed) and $\bar{\mathbb{R}}^p$ is denoted as the set of all p-dimension interval-valued vector, i.e., $\bar{\mathbb{R}} = \{[x^{\mathrm{l}}, x^{\mathrm{r}}] : x^{\mathrm{l}}, x^{\mathrm{r}} \in \mathbb{R}, x^{\mathrm{l}} \leq x^{\mathrm{r}}\}$ and $\bar{\mathbb{R}}^p = \{([x_1^{\mathrm{l}}, x_1^{\mathrm{r}}], \cdots, [x_p^{\mathrm{l}}, x_p^{\mathrm{r}}])^\top : x_j^{\mathrm{l}}, x_j^{\mathrm{r}} \in \mathbb{R}, x_j^{\mathrm{l}} \leq x_j^{\mathrm{r}}, j \in [p]\}$.

Key Definitions. In this part, we introduce some basic definitions to identify the LIND problem. We first show the definition of the interval-valued random variable.

Definition 4.1 (Interval-valued Random Variable)**.** Suppose $X^{\mathrm{l}}, \delta = X^{\mathrm{r}} - X^{\mathrm{l}} \in \mathbb{R}$ are two real-valued random variables [60] defined in $\mathbb{R}$ and δ is a nonnegative random variable. Then, X^{r} is also a real-valued random variables satisfied $X^{\mathrm{l}} \leq X^{\mathrm{r}}$. We define

$\bar{X} = [X^{\mathrm{l}}, X^{\mathrm{r}}] \in \bar{\mathbb{R}}$ as an interval-valued random variable. Then, a p-dimension interval-valued random vector $\bar{\mathbf{X}} = (\bar{X}_1, \cdots, \bar{X}_p)^\top \in \bar{\mathbb{R}}^p$ is a k-tuple of the interval-valued random variables, where $\bar{X}_j$ ($j \in [p]$) is an interval-valued random variable.

The interval-valued random variable is a natural extension of the ordinary real-valued random variable. Then, we define $\bar{\mathscr{D}}$ as the interval probability distribution of $\bar{\mathbf{X}} = (\bar{X}_1, \cdots, \bar{X}_p)^\top$ (denoted as $\bar{\mathbf{X}} \sim \bar{\mathscr{D}}$).

Definition 4.2 (Interval Probability Density Function)**.** Suppose $X^{\mathrm{l}}, X^{\mathrm{r}}$ are two real-valued random variables and have the same continuous pdf $p_X(x)$. We define $\bar{p}_{\bar{X}}(x)$ as the interval pdf of interval-valued random variable $\bar{X}$, where

$$\bar{p}_{\bar{X}}(x) = \left[\min_{x \in [X^{\mathrm{l}}, X^{\mathrm{r}}]} p_X(x), \max_{x \in [X^{\mathrm{l}}, X^{\mathrm{r}}]} p_X(x) \right].$$

Let $\bar{\mathbf{X}} = (\bar{X}_1, \cdots, \bar{X}_p)^\top$ be a p-interval-valued random vector and the interval pdf of $\bar{X}_j$ is $\bar{p}_{\bar{X}_j}(x), j \in [p]$. Then, we denote the joint interval pdf of $\bar{\mathbf{X}}$ as

$$\bar{p}_{\bar{\mathbf{X}}}(\mathbf{x}) = \left[\prod_{j=1}^{p} \min_{x_j \in [X_j^{\mathrm{l}}, X_j^{\mathrm{r}}]} p_{X_j}(x_j), \prod_{j=1}^{p} \max_{x_j \in [X_j^{\mathrm{l}}, X_j^{\mathrm{r}}]} p_{X_j}(x_j) \right],$$
$$\mathbf{x} = (x_1, \cdots, x_p)^\top.$$

Definition 4.3 (Interval Probability Distribution)**.** Let $\bar{\mathbf{X}} = (\bar{X}_1, \cdots, \bar{X}_p)^\top$ be a p-interval-valued random vector with the joint interval pdf $\bar{p}_{\bar{\mathbf{X}}}(\mathbf{x})$. Let $\mathbf{X}^{\mathrm{l}} = (X_1^{\mathrm{l}}, \cdots, X_p^{\mathrm{l}})^\top, \mathbf{X}^{\mathrm{r}} = (X_1^{\mathrm{r}}, \cdots, X_p^{\mathrm{r}})^\top$ be two real-valued random vectors following probability distribution $\mathscr{D}^{\mathrm{l}}, \mathscr{D}^{\mathrm{r}}$. We define $\bar{\mathscr{D}}$ as the interval probability distribution of $\bar{\mathbf{X}}$ (denoted as $\bar{\mathbf{X}} \sim \bar{\mathscr{D}}$), if

$$\bar{\mathscr{D}}(\bar{\mathbb{R}}^p) = \bar{\int} \bar{p}_{\bar{\mathbf{X}}}(\mathbf{x}) d\mathbf{x} = 1,$$

where $\bar{\int} \bar{p}_{\bar{\mathbf{X}}}(\mathbf{x}) d\mathbf{x} = \frac{1}{2} \int d\mathscr{D}^{\mathrm{l}}(\mathbf{x}) + \frac{1}{2} \int d\mathscr{D}^{\mathrm{r}}(\mathbf{x})$. Therefore, $\bar{\mathbf{X}} \sim \bar{\mathscr{D}}$ if and only if $\mathbf{X}^{\mathrm{l}} \sim \mathscr{D}^{\mathrm{l}}$ and $\mathbf{X}^{\mathrm{r}} \sim \mathscr{D}^{\mathrm{r}}$. Then, we denote $\mathbb{P}(\bar{\mathbf{X}} \in \bar{B}) = \bar{\mathscr{D}}(\bar{B})$ as the probability of the event $\{\bar{\mathbf{X}} \in \bar{B}\}$, where $\bar{B} \in \bar{\mathscr{B}}$ and $\bar{\mathscr{B}}$ is the Borel σ-algebra in $\bar{\mathbb{R}}^p$.

Definition 4.4. Let $\bar{\mathbf{X}} = (\bar{X}_1, \cdots, \bar{X}_p)^\top$ be a p-interval-valued random vector with the joint interval pdf $\bar{p}_{\bar{\mathbf{X}}}(\mathbf{x})$ and $\mathbf{X}^{\mathrm{l}} = (X_1^{\mathrm{l}}, \cdots, X_p^{\mathrm{l}})^\top \sim \mathscr{D}^{\mathrm{l}}, \mathbf{X}^{\mathrm{r}} = (X_1^{\mathrm{r}}, \cdots, X_p^{\mathrm{r}})^\top \sim \mathscr{D}^{\mathrm{r}}$ are two real-valued random vectors. Then, the probability with respect to the function $q : \bar{\mathscr{X}} \to \mathbb{R}_+$ is defined as:

$$\mathbb{P}(q(\bar{\mathbf{X}}) \geq \varepsilon) = \frac{1}{2}\int_{\mathrm{A}} d\mathscr{D}^{\mathrm{l}}(\mathbf{x}) + \frac{1}{2}\int_{\mathrm{B}} d\mathscr{D}^{\mathrm{r}}(\mathbf{x}),$$

where $\mathrm{A} = \{\mathbf{X}^{\mathrm{l}} \in \mathbb{R}^p : q(\bar{\mathbf{X}}) \geq \varepsilon\}, \mathrm{B} = \{\mathbf{X}^{\mathrm{r}} \in \mathbb{R}^p : q(\bar{\mathbf{X}}) \geq \varepsilon\}$.

Definition 4.5 (Independence)**.** The interval-valued random vectors $\bar{\mathbf{X}}_1, \cdots, \bar{\mathbf{X}}_n$ are said to be (mutually) independent if and only if the real-valued random vectors $\mathbf{X}_1^{\mathrm{l}}, \cdots, \mathbf{X}_n^{\mathrm{l}}, \mathbf{X}_1^{\mathrm{r}}, \cdots, \mathbf{X}_n^{\mathrm{r}}$ are (mutually) independent. Then, we denote $\bar{\mathbf{X}}_1, \cdots, \bar{\mathbf{X}}_n$ as i.i.d. interval-valued random vectors if and only if $\bar{\mathbf{X}}_1, \cdots, \bar{\mathbf{X}}_n$ are independent and have the same interval probability distribution.

Next, we define the interval expectation for an interval-valued random vector.

Definition 4.6 (Interval Expectation)**.** Suppose $\bar{\mathbf{X}} \sim \bar{\mathscr{D}}$ is an interval-valued random vector. We denote $\mathbf{X}^{\mathrm{l}} = (X_1^{\mathrm{l}}, \cdots, X_p^{\mathrm{l}})^\top$ and $\mathbf{X}^{\mathrm{r}} = (X_1^{\mathrm{r}}, \cdots, X_p^{\mathrm{r}})^\top$, which are two real-valued random vectors following probability distribution $\mathscr{D}^{\mathrm{l}}$ and $\mathscr{D}^{\mathrm{r}}$. Then, the interval expectation of an interval-valued random vector $\bar{\mathbf{X}}$ is defined as,

$$\begin{aligned}\mathbb{E}_{\bar{\mathscr{D}}}[\bar{\mathbf{X}}] &= \frac{1}{2}\int \mathbf{x} d\mathscr{D}^{\mathrm{l}}(\mathbf{x}) + \frac{1}{2}\int \mathbf{x} d\mathscr{D}^{\mathrm{r}}(\mathbf{x}) \\ &= \frac{1}{2}\mathbb{E}[\mathbf{X}^{\mathrm{l}}] + \frac{1}{2}\mathbb{E}[\mathbf{X}^{\mathrm{r}}].\end{aligned}$$

Remark: Most previous works considering that the expectation of an interval is itself an interval [3] were primarily focused on the operation of interval-valued data. However, this paper focuses on learning this type of data (interval) from defineive. Therefore, we give a different definition of the expectation of an interval.

Definition 4.7 (Interval Probability)**.** Suppose $\bar{\mathbf{X}} = (\bar{X}_1, \cdots, \bar{X}_p)^\top$ is an interval-valued random vector with the joint interval pdf $\bar{p}_{\bar{\mathbf{X}}}(\mathbf{x})$, and $\mathbf{X}^{\mathrm{l}} = (X_1^{\mathrm{l}}, \cdots, X_p^{\mathrm{l}})^\top \sim \mathscr{D}^{\mathrm{l}}$ and $\mathbf{X}^{\mathrm{r}} =$

$(X_1^{\mathrm{r}},\cdots,X_p^{\mathrm{r}})^\top \sim \mathscr{D}^{\mathrm{r}}$ are two real-valued random vectors. Then, the probability with respect to the function $g:\bar{\mathscr{X}}\to\mathbb{R}_+$ is defined as:

$$\mathbb{P}(g(\bar{\mathbf{X}})\geq\varepsilon)=\frac{1}{2}\int_{\mathrm{A}}d\mathscr{D}^{\mathrm{l}}(\mathbf{x})+\frac{1}{2}\int_{\mathrm{B}}d\mathscr{D}^{\mathrm{r}}(\mathbf{x}), \tag{4.1}$$

where

$$\mathrm{A}=\{\mathbf{X}^{\mathrm{l}}\in\mathbb{R}^p : g(\bar{\mathbf{X}})\geq\varepsilon\},\mathrm{B}=\{\mathbf{X}^{\mathrm{r}}\in\mathbb{R}^p : g(\bar{\mathbf{X}})\geq\varepsilon\}.$$

Based on the above definitions and the introduction of ordinary classification problems with crisp-valued observations [105], we can identify the LIND problem.

Learning from Interval-valued Data: Let $\bar{\mathscr{X}}\subset\bar{\mathbb{R}}^p$ be the input space of interval-valued observations and $\mathscr{Y}=[K]$ be the output space. Suppose $\bar{S}=\{(\bar{\mathbf{x}}_i,y_i)\}_{i=1}^m$ is a sample drawn i.i.d. from $\bar{\mathscr{D}}$, where $\bar{\mathbf{x}}_i=(\bar{x}_{i1},\cdots,\bar{x}_{ip})^\top\in\bar{\mathscr{X}}$ and $y_i=f(\bar{\mathbf{x}}_i)\in\mathscr{Y}$ be the ground-truth function. Let $\bar{\mathscr{H}}\subset\{\bar{\boldsymbol{h}}:\bar{\mathscr{X}}\to\mathbb{R}^K\}$ be the hypothesis space of the LIND problem and for any $\bar{\boldsymbol{h}}\in\bar{\mathscr{H}}$,

$$\begin{aligned}\bar{\boldsymbol{h}}:&\bar{\mathscr{X}}\to\ \mathbb{R}^K\\&\bar{\mathbf{x}}_i\to(\bar{h}_1(\bar{\mathbf{x}}_i),\cdots,\bar{h}_K(\bar{\mathbf{x}}_i))^\top.\end{aligned}$$

Without loss of generality, we suppose that $\sum_{k=1}^K\bar{h}_k(\bar{\mathbf{x}}_i)=1$ and each $\bar{h}_k(\bar{\mathbf{x}}_i)$ represents the probability of instance $\bar{\mathbf{x}}_i$ belonging to the k-th category. Therefore, we have $\sup_{\bar{\boldsymbol{h}}\in\bar{\mathscr{H}}}\|\bar{\boldsymbol{h}}\|_\infty\leq 1$. Let $\mathscr{L}_{\bar{\mathscr{H}}}=\{\ell(\bar{\boldsymbol{h}}(\bar{\mathbf{x}}),y):\bar{\mathbf{x}}\in\bar{\mathscr{X}},\bar{\boldsymbol{h}}\in\bar{\mathscr{H}},y\in\mathscr{Y}\}$ be the class of functions with respect to the loss ℓ and $\bar{\mathscr{H}}$, where $\ell:\mathbb{R}^K\times\mathscr{Y}\to\mathbb{R}_+$. Based on the ordinary classification problem, we denote $R_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})=\mathbb{E}_{\bar{\mathscr{D}}}[\ell(\bar{\boldsymbol{h}}(\bar{\mathbf{x}}),y)]$ as the risk of the LIND problem. Therefore, the aim of the LIND problem is to find the optimal classifier $\bar{\boldsymbol{h}}^*\in\bar{\mathscr{H}}$ such that $\bar{\boldsymbol{h}}^*=\arg\min_{\bar{\boldsymbol{h}}\in\bar{\mathscr{H}}}R_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})$.

Let $\mathscr{X}\subset\mathbb{R}^p$ be the input space of the corresponding true value of $\bar{\mathscr{X}}$ that can not be observed. Correspondingly, $\mathbf{x}_i=(x_1,\cdots,x_p)^\top\sim\mathscr{D}$ is the true value of $\bar{\mathbf{x}}_i$. Without loss of generality, we suppose that for all $j\in[p],x_{ij}\in\bar{x}_{ij}$. Because of the difficulty to directly

construct $\bar{\boldsymbol{h}}$, we divide $\bar{\boldsymbol{h}}$ into two components, i.e., let $\bar{\boldsymbol{h}} = \boldsymbol{h} \circ \boldsymbol{h}$ where $\boldsymbol{h} : \bar{\mathscr{X}} \to \mathscr{X}$ be a transformation function and $\boldsymbol{h} : \mathscr{X} \to \mathbb{R}^K$ be a hypothesis function. Let $\mathscr{G} \subset \{\boldsymbol{h} : \bar{\mathscr{X}} \to \mathscr{X}\}$ be a set of transformation function and $\mathscr{H} \subset \{\boldsymbol{h} : \mathscr{X} \to \mathbb{R}^K\}$ be the hypothesis space of $\mathscr{X}$. Let $\boldsymbol{h}(\mathbf{x}) = (h_1(\mathbf{x}), \cdots, h_K(\mathbf{x}))^\top, \mathbf{x} \in \mathscr{X}$. Without loss of generality, we suppose that $\sum_{k=1}^K h_k(\mathbf{x}) = 1$ and each $h_k(\mathbf{x})$ represents the probability of instance $\mathbf{x}$ belonging to the k-th category. Then, we have $\sup_{\boldsymbol{h} \in \mathscr{H}} \| \boldsymbol{h} \|_\infty \leq 1$.

4.3 Theoretical Analysis

In this section, the estimation error bound of the LIND problem is presented. Then, we discuss under which conditions we can expect the performances of the method to converge to the optimal accuracy one would have obtained with perfect observations (Similar discussions are shown in [24, 91]). In addition, we give theoretical analysis to show the strengths of multi-view learning, which inspires us to construct the Mv-IIE framework to address the LIND problem (all proofs are shown in the Appendix 4.8).

4.3.1 Theoretical Analysis of LIND Problem

Let $\bar{S}_{\bar{X}} = \{\bar{\mathbf{x}}_i\}_{i=1}^m$ be a sample drawn i.i.d. from $\bar{\mathscr{D}}$. We first introduce the definition of Rademacher complexity of $\bar{\mathscr{H}}$ with respect to $\bar{S}_{\bar{X}}$.

Based on Theorem 3.1, we can obtain the following theorem (proof is similar to Theorem 3.1).

Theorem 4.1. *Suppose that* $\sup_{\|\bar{\boldsymbol{h}}\|_\infty \leq 1} \max_y \ell(\bar{\boldsymbol{h}}, y) \leq C_\ell$*, and all functions in* $\mathscr{L}_{\bar{\mathscr{H}}}$ *are* L_ℓ*-Lipschitz functions. For any* $\delta > 0$*, with probability at least* $1 - \delta$*, each of the following holds for all* $\bar{\boldsymbol{h}} \in \bar{\mathscr{H}}$*:*

$$|R_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}}) - \widehat{R}_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})| \leq 2\sqrt{2} L_\ell \widehat{\mathscr{R}}_{\bar{S}_{\bar{X}}}(\bar{\mathscr{H}}) + 3C_\ell \sqrt{\frac{\log(2/\delta)}{2m}}. \tag{4.2}$$

This theorem presents a generalization bound of the discrepancy between the risk and empirical risk of $\bar{\boldsymbol{h}}$ based on empirical Rademacher complexity. $\widehat{\mathscr{R}}_{\bar{S}_{\bar{X}}}(\bar{\mathscr{H}})$ is in the order of $O(1/\sqrt{m})$ under some certain restrictions of $\bar{\mathscr{H}}$ [4, 23, 69], for example, $\bar{\mathscr{H}}$ has limited-VC dimension or $\bar{\mathscr{H}}$ is a kernel class with bounded trace. Then, if $\widehat{\mathscr{R}}_{\bar{S}_{\bar{X}}}(\bar{\mathscr{H}}) = O(1/\sqrt{m})$, we notice that as $m \to \infty$, $R_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}}) \to \widehat{R}_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})$. Therefore, this bound demonstrates that we can always train a well-performed classifier to address LIND problems when enough interval-valued instances can be collected. Next, we discuss the learnability of the underlying problem with perfect observations in the following theorem. Let $S_X = \{\mathbf{x}_i\}_{i=1}^m$ be the corresponding true value of $\bar{S}_{\bar{X}}$ drawn i.i.d. from $\mathscr{D}$.

Theorem 4.2. *Suppose the conditions of Theorem 4.1 are hold, and* $\max_{\bar{\mathbf{x}}_i \in \bar{\mathscr{X}}, \boldsymbol{g} \in \mathscr{G}} \| \mathbf{x}_i - \boldsymbol{g}(\bar{\mathbf{x}}_i) \|_2 = O(1/m^\gamma), \gamma > 0$*, and all functions in* $\mathscr{H}$ *are* L_h*-Lipschitz functions. For any* $\delta > 0$*, with probability at least* $1-\delta$*, each of the following holds for all* $\boldsymbol{h} \in \mathscr{H}$ *and* $\bar{\boldsymbol{h}} \in \bar{\mathscr{H}}$*:*

$$|R_{\mathscr{D}}(\boldsymbol{h}) - R_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})| \leq 2\sqrt{2} L_\ell (\widehat{\mathscr{R}}_{\bar{S}_{\bar{X}}}(\bar{\mathscr{H}}) + \widehat{\mathscr{R}}_{S_X}(\mathscr{H})) + 6C_\ell \sqrt{\frac{\log(4/\delta)}{2m}} + O(1/m^\gamma). \tag{4.3}$$

Same as $\widehat{\mathscr{R}}_{\bar{S}_{\bar{X}}}(\bar{\mathscr{H}})$, $\widehat{\mathscr{R}}_{S_X}(\mathscr{H})$ is in the order of $O(1/\sqrt{m})$ under some restrictions of $\mathscr{H}$. According to Eq. (4.3) and if $\widehat{\mathscr{R}}_{\bar{S}_{\bar{X}}}(\bar{\mathscr{H}}) = O(1/\sqrt{m}), \widehat{\mathscr{R}}_{S_X}(\mathscr{H}) = O(1/\sqrt{m})$, we notice that as $m \to \infty$, $R_{\mathscr{D}}(\boldsymbol{h}) \to R_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})$. Therefore, Theorem 4.2 reveals the learnability of the LIND problem under certain restrictions.

Remark: The condition

$$\max_{\bar{\mathbf{x}}_i \in \bar{\mathscr{X}}, \boldsymbol{h} \in \mathscr{G}} \| \mathbf{x}_i - \boldsymbol{g}(\bar{\mathbf{x}}_i) \|_2 = O(1/m^\gamma), \gamma > 0,$$

means that any $\boldsymbol{g} \in \mathscr{G}$ can precisely extract information from interval-valued observations, which is difficult to construct this kind of transformation functions. But if the interval size of all interval-valued features is small enough, this condition becomes trivial.

4.3.2 Why Multi-view Methodology Is Used

In this section, we consider why using multi-view learning to address the LIND problem in terms of error rate and estimation error bound.

Let $\mathscr{X}_v$ ($v \in [c]$) be the single-view input space and $\mathscr{X} = \mathscr{X}_1 \times \cdots \times \mathscr{X}_c$ be the multi-view input space. Let $S_X = \{\mathbf{X}_i = (\mathbf{x}_i^1, \cdots, \mathbf{x}_i^c)\}_{i=1}^m \subset \mathscr{X}$ be the multi-view sample drawn i.i.d. form $\mathscr{D}$, where $\mathbf{x}_i^v$ is the single-view observation, $\mathbf{X}_i \in \mathscr{X}$ and $y_i = f(\mathbf{X}_i) \in \mathscr{Y}$ is the ground-truth function. Let $\mathscr{H}_v$ be the hypothesis space of v-th view, where for any $\boldsymbol{h}_v \in \mathscr{H}_v$, $\boldsymbol{h}_v : \mathscr{X}_v \to \mathbb{R}^K$. Then, $f_v : \mathbb{R}^K \to \mathscr{Y}$ is a predict function induced by $\boldsymbol{h}_v$. Lastly, we set $\mathscr{H}_{\text{co}}$ to be the multi-view hypothesis space, where for any $\boldsymbol{h}_{\text{co}} \in \mathscr{H}_{\text{co}}$, $\boldsymbol{h}_{\text{co}} : \mathscr{X} \to \mathbb{R}^K$. Then, we can induce a predict function $f_{\text{co}} : \mathbb{R}^K \to \mathscr{Y}$ by $\boldsymbol{h}_{\text{co}}$.

Error Rate First, we propose a notion called discrepancy set to measure the predict functions difference across different view. Then, we denote $\mathbf{D}_{\mathscr{F}}(f_1, \cdots, f_c)$ as the discrepancy set between the predict functions $f_1, \cdots, f_c$ over $\mathscr{X}$, which is shown as follow:

$$\mathbf{D}_{\mathscr{F}}(f_1, \cdots, f_c) = \left\{ \mathbf{X} = (\mathbf{x}^1, \cdots, \mathbf{x}^c) \in \mathscr{X} : \bigvee_{1 \le v_1 < v_2 \le c} f_{v_1}(\mathbf{x}^{v_1}) \neq f_{v_2}(\mathbf{x}^{v_2}) \right\},$$

here $\bigvee$ represents the logical relation "or". Next, we give the following assumption:

$$\begin{aligned} &\text{For } \mathbf{X} = (\mathbf{x}^1, \cdots, \mathbf{x}^c), \text{ if } f_1(\mathbf{x}^1) = \cdots = f_c(\mathbf{x}^c), \\ &\text{we have } f_{\text{co}}(\mathbf{X}) = f_1(\mathbf{x}^1). \end{aligned} \tag{4.4}$$

This assumption means that if all single-view predictions are same, the multi-view predict function also has the same outcome, which is a trivial assumption. Then, we obtain the following theorem.

Theorem 4.3. *We assert that there exists a uniform constant $M \in (0,1)$ such that for any predict function f_{co} satisfies assumption (4.4), if*

$$\mathbb{P}_{\mathscr{D}}(f_{\text{co}}(\mathbf{X}) \neq y | \mathbf{X} \in \mathbf{D}_{\mathscr{F}}(f_1, \cdots, f_c)) \le M,$$

where y is the ground-truth label. We assert $\mathrm{err}(f_{\mathrm{co}}) \leq \min_{v\in[c]} \mathrm{err}(f_v)$*, where* $\mathrm{err}(f_{\mathrm{co}}) = \mathbb{P}_{\mathscr{D}}(f_{\mathrm{co}}(\mathbf{X}) \neq y)$.

We can easily find $M < 1$ that satisfies the condition in Theorem 4.3. According to Theorem 4.3, we always have the error rate of a multi-view prediction function f_{co} is lower than that of any single-view prediction function $f_v, v \in [c]$ when $\mathbb{P}_{\mathscr{D}}(f_{\mathrm{co}}(\mathbf{X}) \neq y | \mathbf{X} \in \mathbf{D}_{\mathscr{F}}(f_1, \cdots, f_c)) \to 0$, which means that using multi-view methodology can reduce the error rate of the predict function for the classification tasks. We can achieve $\mathbb{P}_{\mathscr{D}}(f_{\mathrm{co}}(\mathbf{X}) \neq y | \mathbf{X} \in \mathbf{D}_{\mathscr{F}}(f_1, \cdots, f_c)) \to 0$ by reducing the size of the discrepancy set $\mathbf{D}_{\mathscr{F}}(f_1, \cdots, f_c)$. Based on the above theoretical analysis, we decide to find appropriate multi-view features that can achieve well and similar performance on all single-view classifiers to reduce the size of the discrepancy set $\mathbf{D}_{\mathscr{F}}(f_1, \cdots, f_c)$.

Estimation Error Bound $\mathscr{L}_{\mathscr{H}_{\mathrm{co}}} = \{\ell(\boldsymbol{h}_{\mathrm{co}}(\mathbf{X}), y) : \mathbf{X} \in \mathscr{X}, \boldsymbol{h}_{\mathrm{co}} \in \mathscr{H}_{\mathrm{co}}, y \in \mathscr{Y}\}$ be the class of functions with respect to the loss ℓ and $\mathscr{H}_{\mathrm{co}}$, where $\ell : \mathbb{R}^K \times \mathscr{Y} \to \mathbb{R}_+$. The risk of $\boldsymbol{h}_{\mathrm{co}}$ is denoted as $R_{\mathscr{D}}(\boldsymbol{h}_{\mathrm{co}}) = \mathbb{E}_{\mathscr{D}}[\ell(\boldsymbol{h}_{\mathrm{co}}(\mathbf{X}), y)]$. Next, we give the following theorem to bound $\mathscr{R}_{S_X}(\mathscr{H}_{\mathrm{co}})$.

Theorem 4.4. *For any $m \geq 1$, we have $\mathscr{R}_{S_X}(\mathscr{H}_{\mathrm{co}}) \leq \max_{v\in[c]} \mathscr{R}_{S_{X^v}}(\mathscr{H}_v)$, where $S_{X^v} = \{\mathbf{x}_i^v\}_{i=1}^m$.*

According to Theorem 4.4, if

$$\max_{v\in[c]} \mathscr{R}_{S_{X^v}}(\mathscr{H}_v) - \min_{v\in[c]} \mathscr{R}_{S_{X^v}}(\mathscr{H}_v) \to 0, \tag{4.5}$$

we have $\mathscr{R}_{S_X}(\mathscr{H}_{\mathrm{co}}) \leq \min_{v\in[c]} \mathscr{R}_{S_{X^v}}(\mathscr{H}_v)$, which demonstrates that we can obtain tighter estimation error bound by applying the multi-view methodology. Inspired by the above theoretical analysis, we achieve Eq. (4.5) by finding appropriate multi-view features that can achieve similar performance on all single-view classifiers.

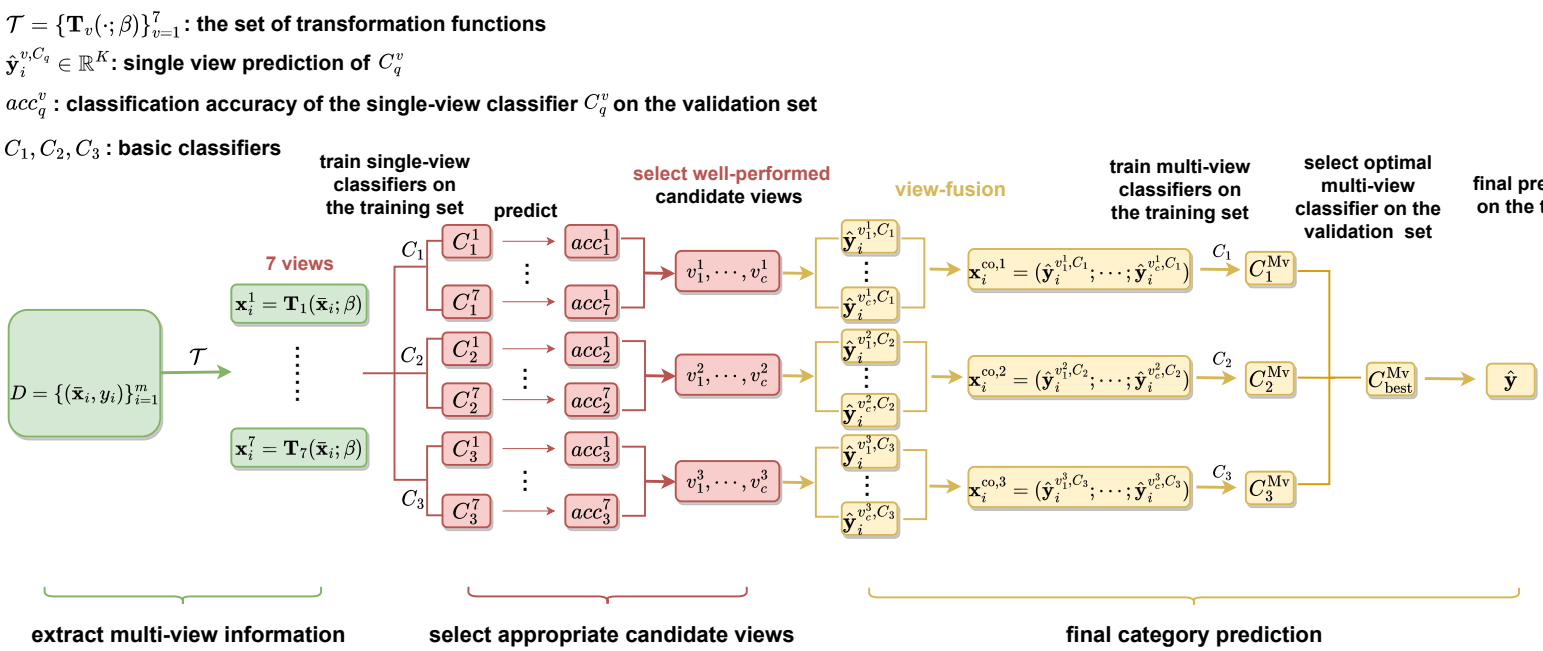


Figure 4.2: **Mv-IIE** structure. The first part (denoted in **green**) is to extract the multi-view information from the interval-valued dataset D. Then, the multi-view classifier with two structures is used to handle the extracted multi-view information. The first structure (denoted in **red**) is used to select well-performed candidate views. The second structure (denoted in **yellow**) aims to train the final multi-view classifiers by using the view-fusion representation vectors.

4.4 Model Construction

In this section, a new algorithm called *multi-view interval information extraction* (Mv-IIE) approach is presented to address the LIND problem. The structure of Mv-IIE is shown in Figure 4.2. We describe this proposed framework in detail in the following paragraph.

First, we introduce two types of fuzzy number and four different defuzzification methods used to construct the membership function-based method. The first type of fuzzy number called triangular fuzzy number. A triangular fuzzy number $\widetilde{x}$ can be characterized by $\mathrm{Tr}(a_1, b_1, a_2)$. and the membership function is shown as follows:

$$\mu_{\widetilde{x}}(t) = \begin{cases} 0, & t < a_1 \\ \dfrac{t-a_1}{b_1-a_1}, & a_1 \le t < b_1 \\ \dfrac{t-a_2}{b_1-a_2}, & b_1 \le t < a_2 \\ 0, & t \ge a_2. \end{cases}$$

A Gaussian fuzzy number is the second type of fuzzy number. A Gaussian fuzzy number $\widetilde{x}$ can be characterized by $\mathrm{Ga}(c, \delta_1, \delta_2)$ and the membership function is shown as follows:

$$\mu_{\widetilde{x}}(t) = \begin{cases} \exp(-(t-c)/2\delta_1)^2, & t < c \\ \exp(-(t-c)/2\delta_2)^2, & t \ge c. \end{cases}$$

We denote $D = \{(\bar{\mathbf{x}}_i, y_i)\}_{i=1}^m$ as the interval-valued dataset, where $\bar{\mathbf{x}}_i = (\bar{x}_{i1}, \cdots, \bar{x}_{ip})^\top \in \bar{\mathbb{R}}^p, y_i \in [K]$. Then, the construction process of the membership function-based method is introduced. We divide this method into two parts. In the first part, we use two functions $F_1(\cdot;\beta), F_2(\cdot;\beta)$ to transfer an interval-valued feature to a triangular fuzzy number and a Gaussian fuzzy number respectively. $F_1(\cdot;\beta), F_2(\cdot;\beta)$ are defined as:

$$F_1(\bar{x}_{ij};\beta) = \mathrm{Tr}(x_{ij}^{\mathrm{l}}, \beta x_{ij}^{\mathrm{l}} + (1-\beta)x_{ij}^{\mathrm{r}}, x_{ij}^{\mathrm{r}}),$$
$$F_2(\bar{x}_{ij};\beta) = \mathrm{Ga}(\beta x_{ij}^{\mathrm{l}} + (1-\beta)x_{ij}^{\mathrm{r}}, S_{1j}, S_{2j})$$

$$S_{1j} = \sqrt{\mathrm{Var}(\mathrm{A_j}), S_{2j}} = \sqrt{\mathrm{Var}(B_j)},$$
$$A_j = \{x_{ij}^{\mathrm{l}} : i \in [m], (\bar{\mathbf{x}}_i, y_i) \in D\},$$
$$B_j = \{x_{ij}^{\mathrm{r}} : i \in [m], (\bar{\mathbf{x}}_i, y_i) \in D\}, j \in [p],$$

where $\beta \in [0,1]$ is a hyperparameter to control the shape of the membership function, $\mathrm{Var}(\cdot)$ is used to find the variance of the set. Using the above process, one interval-valued feature $\bar{\mathbf{x}}_i$ can be transferred into two fuzzy-valued features $\widetilde{\mathbf{x}}_i^1 = (\widetilde{x}_{i1}^1, \cdots, \widetilde{x}_{ip}^1)^\top$ and $\widetilde{\mathbf{x}}_i^2 = (\widetilde{x}_{i1}^2, \cdots, \widetilde{x}_{ip}^2)^\top$, where

$$\widetilde{\mathbf{x}}_i^\tau = \mathbf{F}_\tau(\bar{\mathbf{x}}_i; \beta) = (F_\tau(\bar{x}_{i1}; \beta), \cdots, F_\tau(\bar{x}_{ip}; \beta))^\top, \tau = 1, 2.$$

In the second part, we use the four defuzzification methods to transfer the two fuzzy-valued features $\widetilde{\mathbf{x}}_i^1, \widetilde{\mathbf{x}}_i^2$ into eight crisp-valued features

$$\mathrm{MOM} \circ \mathbf{F}_\tau(\bar{\mathbf{x}}_i; \beta), \mathrm{COG} \circ \mathbf{F}_\tau(\bar{\mathbf{x}}_i; \beta), \mathrm{ALC} \circ \mathbf{F}_\tau(\bar{\mathbf{x}}_i; \beta), \mathrm{VAL} \circ \mathbf{F}_\tau(\bar{\mathbf{x}}_i; \beta), \tau = 1, 2.$$

According to Eq. (3.15), we find that $\mathrm{MOM} \circ \mathbf{F}_\tau(\bar{\mathbf{x}}_i; \beta) = \mathrm{MOM} \circ \mathbf{F}_2(\bar{\mathbf{x}}_i; \beta)$. Therefore, we can use the aforementioned membership function-based method to extract multi-view information, which contains seven parts: $\mathrm{MOM} \circ \mathbf{F}_1(\bar{\mathbf{x}}_i; \beta)$ and $\mathrm{COG} \circ \mathbf{F}_\tau(\bar{\mathbf{x}}_i; \beta), \mathrm{ALC} \circ \mathbf{F}_\tau(\bar{\mathbf{x}}_i; \beta), \mathrm{VAL} \circ \mathbf{F}_\tau(\bar{\mathbf{x}}_i; \beta), \tau = 1, 2$. We denote $\mathscr{T} = \{\mathbf{T}_v(\cdot; \beta)\}_{v=1}^7$ as a set of transfer functions constructed by using the membership function-based method, where

$$\mathbf{T}_1 = \mathrm{MOM} \circ \mathbf{F}_1, \mathbf{T}_2 = \mathrm{COG} \circ \mathbf{F}_1, \mathbf{T}_3 = \mathrm{COG} \circ \mathbf{F}_2,$$
$$\mathbf{T}_4 = \mathrm{ALC} \circ \mathbf{F}_1, \mathbf{T}_5 = \mathrm{ALC} \circ \mathbf{F}_2, \mathbf{T}_6 = \mathrm{VAL} \circ \mathbf{F}_1, \mathbf{T}_7 = \mathrm{VAL} \circ \mathbf{F}_2.$$

By applying the aforementioned transfer functions to extract crisp-valued information from the interval-valued data, one interval-valued feature $\bar{\mathbf{x}}_i$ can be transferred into seven different parts $\mathbf{X}_i^{\mathrm{Mv}} = (\mathbf{x}_i^1, \cdots, \mathbf{x}_i^7)$, where for any $i \in [m], v \in [7], \mathbf{x}_i^v = \mathbf{T}_v(\bar{\mathbf{x}}_i; \beta), \mathbf{T}_v \in \mathscr{T}$.

Through the above process, one interval-valued feature $\bar{\mathbf{x}}_i$ can be transferred into seven different parts $\mathbf{X}_i^{\mathrm{Mv}} = (\mathbf{x}_i^1, \cdots, \mathbf{x}_i^7)$, where for any $i \in [m], v \in [7]$,

$$\mathbf{x}_i^v = \mathbf{T}_v(\bar{\mathbf{x}}_i; \beta), \mathbf{T}_v \in \mathscr{T}.$$

Algorithm 3 Mv-IIE

Input: data $D = \{(\bar{\mathbf{x}}_i, y_i)\}_{i=1}^{m}$, the basic classifiers $\mathrm{C}_1, \mathrm{C}_1$ and C_3;
Initial: network parameters of C_3 and split D into a training set D^{tr} with size m_1, a validation set D^{va} with size m_2 and a test set D^{te} with size m_3;
Compute: extract multi-view information : $\mathbf{x}_i^v = \mathbf{T}_v(\bar{\mathbf{x}}_i; \beta), \mathbf{T}_v \in \mathscr{T}, i \in [m], v \in [7]$;
Train: single-view classifiers $\mathrm{C}_q^v, v \in [7], q \in [3]$ on the training set $\{(\mathbf{x}_i^v, y_i) | (\bar{\mathbf{x}}_i, y_i) \in D^{\mathrm{tr}}\}_{i=1}^{m_1}$;
Compute: classification accuracy of the single-view classifiers $\mathrm{C}_q^v, v \in [7], q \in [3]$ on the validation set $\{(\mathbf{x}_i^v, y_i) | (\bar{\mathbf{x}}_i, y_i) \in D^{\mathrm{va}}\}_{i=1}^{m_2}$;
Select: c candidate views for each $q \in [3]$, denoted as $\mathscr{V}^q = \{v_1^q, \cdots, v_c^q\}$, that achieve higher classification accuracy than the rest of the views;
Compute: view-fusion representation vector :

$$\mathbf{x}_i^{\mathrm{co},q} = (\hat{\mathbf{y}}_i^{v_1^q, \mathrm{C}_q}; \cdots; \hat{\mathbf{y}}_i^{v_c^q, \mathrm{C}_q}), i \in [m], q \in [3],$$

where $\hat{\mathbf{y}}_i^{v, \mathrm{C}_q} \in \mathbb{R}^K$ is the category prediction for the v-th view of the i-th data by applying C_q;
Train: multi-view classifiers $\mathrm{C}_q^{\mathrm{Mv}}, q \in [3]$ on the training set $\{(\mathbf{x}_i^{\mathrm{co},q}, y_i) | (\bar{\mathbf{x}}_i, y_i) \in D^{\mathrm{tr}}\}_{i=1}^{m_1}$;
Select: the optimal multi-view classifier $\mathrm{C}_{\mathrm{best}}^{\mathrm{Mv}}$ with optimal hyperparameters that can obtain the highest classification accuracy on the validation set $\{(\mathbf{x}_i^{\mathrm{co},q}, y_i) | (\bar{\mathbf{x}}_i, y_i) \in D^{\mathrm{va}}\}_{i=1}^{m_2}$;
Output: $\mathrm{C}_{\mathrm{best}}^{\mathrm{Mv}}$ with optimal hyperparameters and use $\mathrm{C}_{\mathrm{best}}^{\mathrm{Mv}}$ to test the performance on the test set $\{(\mathbf{x}_i^{\mathrm{co},q}, y_i) | (\bar{\mathbf{x}}_i, y_i) \in D^{\mathrm{te}}\}_{i=1}^{m_3}$.

Then, we obtain the multi-view information $D_{\mathrm{Mv}} = \{(\mathbf{x}_i^1, y_i, 1), \cdots, (\mathbf{x}_i^7, y_i, 7)\}_{i=1}^{m}$ by using the above mentioned method. For any $(\mathbf{x}_i^v, y_i, v) \in D_{\mathrm{Mv}}, y_i \in [K]$ is the category label, and $v \in [7]$ is the view label.

Motivation of transformation functions construction: The interval-valued features contain similar structures and properties with fuzzy numbers [30], which both exist a considerable amount of uncertainty. Further, the α-cut of a fuzzy number $\widetilde{x}$ is defined as $\{t \in \mathbb{R} | \mu_{\widetilde{x}}(t) \leq \alpha\}$ ($\mu_{\widetilde{x}}(t)$ is the membership function of $\widetilde{x}$), which is a closed and bounded interval. Therefore, we design two fuzzilization methods to transfer the interval-valued features into two well-defined fuzzy numbers. Moreover, the four membership function-based methods can extract different crucial discriminant information from fuzzy numbers. For example, MOM finds the maximum membership level but ignores the changing trend of the membership function, while COG takes into account the trend

and finds the centroid of the area bounded by the membership function. Through the above analysis, it inspired us to construct a set of transformation functions by fusing the two fuzzilization methods and the four membership function-based methods to extract multi-view discriminant information. Experimental results shown in Sections 4.5.2 and 4.5.3 verify the rationality and efficacy of the fuzzy transformation functions.

Next, we propose a multi-view classifier with two parts to train the multi-view information, which aims to minimize the empirical risk $\widehat{R}_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})$ in Section 4.3.1. The first part (denoted in red in Figure 4.2) is used to select appropriate multi-view information. We apply support vector machines, random forests and neural networks as three basic classifiers, which denoted as $\mathrm{C}_1, \mathrm{C}_2$ and C_3. Then, we apply the three basic classifiers to train single-view classifiers $\mathrm{C}_q^v, v \in [7], q \in [3]$ on the training set, and we select several well-performed views with the number of c as the candidate views for each basic classifier on the validation set. This selected approach is inspired by the theoretical analysis of Theorem 4.3 and 4.4. Let $\hat{\mathbf{y}}_i^{v,\mathrm{C}_q} \in \mathbb{R}^K, i \in [m], v \in [7], q \in [3]$ denoted as the category prediction for the v-th view of the i-th instance by applying the basic classifier C_q, and $\mathscr{V}^q = \{v_1^q, \cdots, v_c^q\}, q \in [3]$ denoted as the selected candidate views for basic classifier C_q. We do not combine all views, as doing so would not only substantially increase the complexity of the algorithm, but also our experiments show that combining just two views can yield sufficiently good classification performance. Consequently, we design a technique to select the candidate views and let $c = 2$ in this paper.

The second part (denoted in yellow in Figure 4.2) aims to train the final multi-view classifiers by using the selected candidate views. For each basic classifier $\mathrm{C}_q, q \in [3]$, the category predictions of the selected candidate views are integrated to obtain $\mathbf{x}_i^{\mathrm{co},q} = (\hat{\mathbf{y}}_i^{v_1^q,\mathrm{C}_q}; \cdots; \hat{\mathbf{y}}_i^{v_c^q,\mathrm{C}_q}), i \in [m], q \in [3]$ as view-fusion representation vector, and we use $\mathbf{x}_i^{\mathrm{co},q}$ as input and C_q as a classifier to train the multi-view classifier $\mathrm{C}_q^{\mathrm{Mv}}$ on the training set and select the multi-view classifier $\mathrm{C}_{\mathrm{best}}^{\mathrm{Mv}} \in \{\mathrm{C}_1^{\mathrm{Mv}}, \mathrm{C}_2^{\mathrm{Mv}}, \mathrm{C}_3^{\mathrm{Mv}}\}$ with optimal

hyperparameters on the validation set. Finally, the trained multi-view classifier $C_{\text{best}}^{\text{Mv}}$ with optimal hyperparameters are used to get the final category prediction $\hat{\mathbf{y}}$ of $\bar{\mathbf{x}} \in D^{\text{te}}$ on the test set. More detail of Mv-IIE is shown in **Algorithm** 3.

4.5 Experiments

In this section, we compare the proposed model with several baselines on both synthetic and real-world datasets and introduce an application of our method.

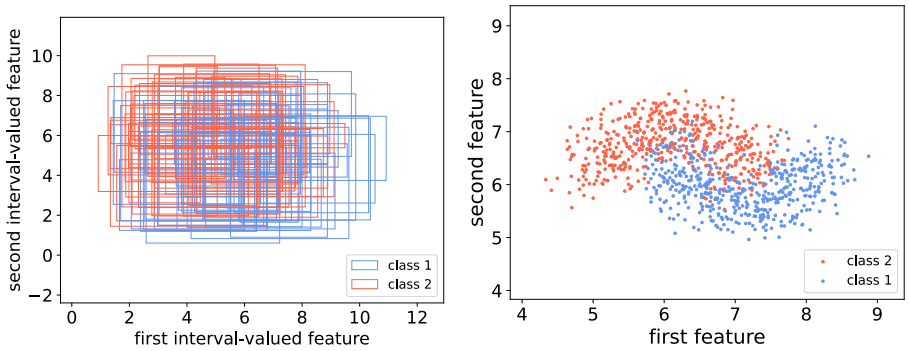
4.5.1 Baselines

This section gives a brief introduction of all baselines. We use the state-of-the-art methods, **D-LDA** [122], **URF** [120], **GURF** [121], **AGURF** [121], **DF-SVM** [99], **DF-MLP** [99], as the first six baselines. Next three baselines called **L-IIE**, **U-IIE** and **M-IIE** that take the lower bound, upper bound and midpoint values from intervals to train the three basic classifiers. The last three baseline called **Mv-2-LU, Mv-2-LM, Mv-2-UM** are constructed based on **Mv-IIE** algorithm. Instead of employing a membership function-based approach to extract multi-view information, **Mv-2-LU, Mv-2-LM, Mv-2-UM** use distinct combinations of views: the upper and lower bounds, the lower bound and the midpoint, and the upper bound and the midpoint of the intervals, respectively.

4.5.2 Experiments on Synthetic Datasets

In this section, we verify the efficacy of the proposed framework on two synthetic datasets. First, we introduce the process of the synthetic datasets generation.

Interval-valued Dataset Generation. We use two different mechanisms to construct synthetic interval-valued datasets. In the first data-generation mechanism, we generate the crisp-valued dataset $\{(\mathbf{x}_i = (x_{i1}, x_{i2})^\top, y_i)\}_{i=1}^n$ in two categories by the double



(a) 1st synthetic dataset (b) 1st synthetic dataset

Figure 4.3: Synthetic datasets. From (a), each rectangle represents one interval-valued instance. (b) plot the the center of the interval-valued data (rectangle) to show the separability of the synthetic dataset.

Number	Feature 1	Feature 2	Feature 3	Label
1	8	3	7	0
2	6	5	2	0
3	12	4	12	1
4	5	6	8	1
5	4	9	10	0
6	2	1	4	1

select Feature1 $T = 2$

Feature1	Label	Number
[2, 7]	0	2, 5
	1	4, 6
[7, 12]	0	1
	1	3

transfer into interval-valued data

Feature 1	Feature 2	Feature 3	Label
[2, 7]	[5, 9]	[2, 10]	0
[2, 7]	[1, 6]	[4, 8]	1
[7, 12]	[3, 3]	[7, 7]	0
[7, 12]	[4, 4]	[12, 12]	1

Figure 4.4: The intervalization approach.

moon data generator. Then, we use the generated crisp-valued dataset to construct the first interval-valued dataset $\{\bar{\mathbf{x}}_i = (\bar{x}_{i1}, \bar{x}_{i2})^\top, y_i\}_{i=1}^n$, where each $\bar{x}_{ij}$ is an interval characterized by $[x_{ij} - a_{ij}, x_{ij} + b_{ij}]$. We let $a_{ij} \sim U[0.5, 1], b_{ij} \sim U[2, 4]$ and $n = 2000$ to generate the first synthetic dataset ($U[a, b]$ denotes the uniform distribution over $[a, b]$). Visualizations of the first two synthetic datasets are shown in Figure 4.3.

In the second data-generation mechanism, we first select one dataset (Letter Recognition dataset selected from the UCI Machine Learning Repository https://archive-beta.ics.uci.edu/) denoted as $D_{\mathrm{R}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})^\top \in \mathbb{R}^p$, and $y_i \in [K]$. Then, we present one intervalization approach to generate the second synthetic interval-valued dataset (see Figure 4.4). We select the first L features in D_{R} and find the maximum value $x_l^{\max}$ and minimum value $x_l^{\min}$ of each feature l, so for any

Table 4.1: Experiment results on the three synthetic datasets. The bold value represents the highest accuracy in each column. p is the p-value of the Wilcoxon rank-sum test between the best performance and other algorithms. * represents $p < 0.05$, meaning that Mv-IIE outperforms other baselines significantly at the 0.05 significance level [125].

	1st synthetic			2nd synthetic	
Algorithms	Test accuracy	p	Time(s)	Test accuracy	p
D-LDA	81.79% ±1.50%	0.00016*	48.9	91.64% ±2.42%	0.0042*
URF	82.43% ±1.48%	0.00016*	546.2	91.86% ±2.27%	0.0072*
GURF	90.56% ±1.15%	0.00016*	564.7	92.67% ±2.01%	0.032*
AGURF	94.01% ±1.02%	0.0022*	608.4	93.96% ±1.98%	0.046*
DF-SVM	97.82% ±0.61%	0.046*	16.1	94.26% ±2.10%	0.39
DF-MLP	97.13% ±1.04%	0.024*	308.4	92.21% ±2.15%	0.030*
L-IIE	97.90% ±0.76%	0.048*	374.6	90.49% ±1.88%	0.0012*
U-IIE	76.95% ±1.43%	0.00016*	374.6	89.31% ±2.94%	0.00067*
M-IIE	89.85% ±0.92%	0.00016*	374.6	94.02% ±2.07%	0.048*
Mv-2-LU	98.18% ±0.81%	0.74	579.2	93.14% ±1.95%	0.038*
Mv-2-LM	98.20% ±0.76%	0.76	579.2	93.48% ±1.85%	0.041*
Mv-2-UM	91.00% ±2.26%	0.00016*	579.2	93.38% ±2.26%	0.040*
Mv-IIE	**98.25%** ± **0.69%**	—-	594.6	**94.66%** ± **1.81%**	—-

$l \in [L], i \in [n], x_{ip} \in [x_l^{\min}, x_l^{\max}]$. We bisect the interval $[x_l^{\min}, x_l^{\max}]$ into T intervals $[x_l^0, x_l^1], [x_l^2, x_l^3], \cdots, [x_l^{T-1}, x_l^T]$. We denote for any $l \in [L]$, $t \in [L]$, and $k \in [K]$,

$$I_{lk}^t = \{(\mathbf{x}_i, y_i) \in D_R : x_{il} \in [x_l^{t-1}, x_l^t], y_i = k\}.$$

Finally, we transfer set I_{lk}^t into an interval-valued data $(([x_1^{\mathrm{l}}, x_1^{\mathrm{r}}], \cdots, [x_p^{\mathrm{l}}, x_p^{\mathrm{r}}])^\top, k)$, where

$$x_j^{\mathrm{l}} = \min_{(\mathbf{x}_i, k) \in I_{lk}^t} x_{ij},\ x_j^{\mathrm{r}} = \max_{(\mathbf{x}_i, k) \in I_{lk}^t} x_{ij}, j \in [p]$$

Then, let $L = 4, T = 12$, we generate the second synthetic interval-valued dataset by using the aforementioned data-generation mechanism.

Implementation. For **URF**, **AGURF**, **DF-MLP**, **L-IIE**, **U-IIE**, **M-IIE**, **Mv-2-LU, Mv-2-LM, Mv-2-UM** and **Mv-IIE** with basic classifier C_3, Adam [67] is used as the optimization algorithm with momentum = 0.9, weight decay = 0.0001, and cross-entropy loss is used as the category label prediction loss. We set epochs equal to 200 and the mini-batch size equal to 200 for all datasets. The network structure of the basic classifier C_3 is a two-layer network with ReLU and Dropout in all the layers ($100 \times 100 \times$ #classes). For each algorithm on each dataset, we randomly divide each dataset into a training set (60%), a validation set (20%) and a test set (20%). First, we select the hyperparameters that can obtain the highest classification accuracy on the validation set. The hyperparameters that need to be selected are shown in Table 4.2. Then, the selected optimal hyperparameters are used to test the performance of each algorithm on the test set. In addition, the validation set is also used to select the candidate views of our proposed framework. We repeat the entire experiment process 10 times. Thus, the final results are shown in the form of "mean± standard deviation". Classification accuracy is used to evaluate the performance of the proposed model. We implement the model with PyTorch 1.9.0. All experiments are conducted on a NVIDIA Quadro GV100 GPU with 32 GB memory.

Experiment Results Analysis. In our experiments, we compare the performance of the Mv-IIE framework with the six baselines on the two generated synthetic datasets. The experimental results are shown in Table 4.1. From these results, it can be seen that the proposed model achieves the best classification accuracy on the two synthetic datasets. Further, results of the Wilcoxon rank-sum test [155] show that our approach outperforms **D-LDA**, **URF**, **GURF**,**AGURF**, **DF-MLP**, **L-IIE**, **U-IIE**, **M-IIE** and **Mv-IIE-2** significantly at the 0.05 significance level in most cases. Further, our method outperforms **Mv-2**, which verifies the rationality of the theoretical analysis of Theorems 4.3 and 4.4 (see Section 4.3.1). In addition, we present the experimental running times for the proposed algorithms and all baselines in Table 4.1.

Table 4.2: Hyperparameters for the proposed method and four baselines

Algorithm	Basic classifier	Hyperparameters	Ranges
DF-SVM		regularization parameter, kernel type, β	$\{0.1, 0.2, \cdots, 1, 2, \cdots, 10\}$, {'linear', 'poly', 'rbf'}, $\{0, 0.1, \cdots, 1\}$
DF-MLP		learning rate, β	$\{0.001, 0.01, 0.1\}$, $\{0, 0.1, \cdots, 1\}$
L-IIE, U-IIE, Mv-2	SVM	regularization parameter, kernel type	$\{0.1, 0.2, \cdots, 1, 2, \cdots, 10\}$, {'linear', 'poly', 'rbf'}
	RF	min samples leaf, the number of trees	$\{1, \cdots, 10\}$, $\{5, 10, \cdots, 100\}$
	Net	learning rate	$\{0.001, 0.01, 0.1\}$
Mv-IIE	same above	same above, β	same above, $\{0, 0.1, \cdots, 1\}$

All baselines only utilize a subset of the information inherent to interval-valued data, potentially leading to the omission of crucial discriminative information and consequently diminishing model performance. In contrast, our proposed method harnesses the power of fuzzy-based transformation functions and multi-view learning to comprehensively extract vital discriminative information from interval-valued data. As a result, it consistently outperforms the other baseline methods in terms of classification performance. All these results verify the superiority of the proposed model in addressing the LINO problem.

4.5.3 Experiments on Real-world Datasets

This section illustrates the experimental results on two real-world datasets which are used to verify the efficacy of the proposed framework. Next, we briefly introduce the two real-world datasets used in the experiments.

Mushroom Dataset : The first dataset is extracted from [1], which contains 248 instances in 17 fungi species categories. There are five interval-valued variables: the pileus cap width Pw, the stipe length Sl, the stipe thickness St, the spores major axis length Sma, and the spores minor axis length Smi. The goal of our experiment on this dataset is to predict the species category of the California mushroom using five interval-valued features.

Weather Dataset : The second dataset is the meteorological data of Washington (from January 1, 2016 to December 31, 2021), provided by the 'Reliable Prognosis' site [2], which contains 2191 instances. Each instance in this dataset is the meteorological data for one day in Washington, which is described by five interval-valued variables (air temperature T, atmospheric pressure at weather station level $P0$, atmospheric pressure reduced to main sea level P, humidity U and dew-point temperature Td) and one category variable (Precipitation or not: 0 ≡ No Precipitation, 1 ≡ Precipitation). We

[1] See https://www.mykoweb.com/CAF/

[2] See https://rp5.ru/

aim to use the five interval-valued features for precipitation prediction.

Implementation. The experiment details of the proposed method and the four baselines are basically the same as the synthetic datasets. We note that the mushroom dataset is an imbalanced dataset which means that each category contains a different number of instances. Therefore, we preprocess this dataset using a random oversampling technique (KMeansSMOTE [74]) and use balanced accuracy [12] instead of ordinary classification accuracy to compare model performance on the mushroom dataset. After the process of the random oversampling technique, the data of each category in the mushroom dataset is expanded to 30. In addition, the Wilcoxon rank-sum test results of the method, which obtains the best performance, compared to the other methods are given on real-world datasets.

Experiment Results Analysis. The experiment results on the two real-world datasets are shown in Table 4.3. From the results of classification accuracy and the Wilcoxon rank-sum test, it can be seen that the proposed model outperforms all baselines significantly at the 0.05 significance level nearly in all cases. **DF-SVM** and **DF-MLP** perform much worse than our methods on the mushroom dataset because they ignore some crucial discriminant information from this dataset. **AGURF** obtained similar results to our method on the mushroom dataset. This similarity arises because **AGURF** aims to address imbalanced interval-valued data, and the mushroom dataset is imbalanced. As a result, **AGURF** can achieve favorable outcomes. However, our method significantly outperforms **AGURF** on other datasets. In these comparison, our methods via multi-view learning and fuzzy transformation functions can extract more discriminant information. These results again demonstrate the superiority of our method in addressing classification problems with interval-valued data.

Ablation Study. To verify the advantage of using multi-view methodology, we apply all single-view classifiers ($C_q^v, v \in [7], q \in [3]$, see Section 4.4) to test classification per-

Table 4.3: Experiment results on the two real-world datasets. The bold value represents the highest accuracy in each column. p is the p-value of the Wilcoxon rank-sum test between the best performance and other algorithms. * represents $p < 0.05$, meaning that Mv-IIE outperforms other baselines significantly at the 0.05 significance level [125].

	Mushroom dataset		Weather dataset	
Algorithms	Test accuracy	p	Test accuracy	p
D-LDA	81.25% ±2.58%	0.0021*	94.74% ±1.34%	0.0018*
URF	81.45% ±2.41%	0.026*	95.55% ±1.17%	0.0035*
GURF	82.26% ±2.56%	0.047*	96.23% ±1.09%	0.031*
AGURF	83.01% ±2.59%	0.077	96.85% ±1.17%	0.038*
DF-SVM	76.67% ±3.86%	0.00067*	97.12% ±0.98%	0.39
DF-MLP	79.39% ±3.32%	0.019*	96.83% ±0.98%	0.038*
L-IIE	76.36% ±6.62%	0.00067*	93.56% ±0.96%	0.00016*
U-IIE	79.14% ±3.58%	0.015*	94.06% ±0.90%	0.00016*
M-IIE	79.34% ±4.75%	0.019*	97.08% ±0.73%	0.050*
Mv-2-LU	81.74% ±5.13%	0.036*	96.76% ±0.91%	0.042*
Mv-2-LM	82.17% ±3.36%	0.042*	97.06% ±0.93%	0.046*
Mv-2-UM	81.68% ±4.64%	0.031*	97.14% ±0.98%	0.41
Mv-IIE	**83.69%** ± **3.39%**	—-	**97.26%** ± **0.81%**	—-

formance on both synthetic and real-world datasets. In Table 4.4, we have included all experimental results of these single-view classifiers, both on synthetic and real-world datasets. These results unequivocally demonstrate that our method outperforms all single-view classifiers, which verifies theoretical analysis in Section 4.3.1 that using multi-view learning can improve model performance in addressing LIND problems. While it's true that some single-view classifiers may achieve results similar to our method on specific datasets, they tend to struggle on others. For instance, view 3 produced similar results (94.38% ±2.05%) to our method on one dataset but performed inadequately on the remaining three datasets. Therefore, our method employs multi-view learning to

Table 4.4: Experiment results of each signal view on the synthetic and real-world datasets.

Algorithms	1^{st} synthetic	2^{nd} synthetic	Mushroom	Weather
view 1	98.12% ±0.66%	94.22% ±2.05%	82.29% ±5.26%	97.03% ±0.68%
view 2	96.50% ±0.56%	94.26% ±1.99%	82.85% ±5.06%	97.12% ±0.74%
view 3	95.20% ±0.56%	94.38% ±2.05%	82.44% ±4.65%	97.01% ±0.94%
view 4	97.82% ±0.61%	94.26% ±2.10%	82.45% ±5.26%	97.12% ±0.98%
view 5	98.12% ±0.66%	94.17% ±1.87%	82.70% ±4.88%	96.96% ±0.89%
view 6	98.12% ±0.66%	94.17% ±2.13%	82.78% ±5.08%	97.01% ±0.87%
view 7	98.12% ±0.66%	94.36% ±2.02%	82.89% ±5.13%	97.05% ±0.75%
Mv-IIE	98.25% ±0.96%	94.66% ±1.81%	83.69% ±3.39%	97.26% ±0.81%

Table 4.5: Experiment results of the ablation study on the mushroom and weather datasets.

Dataset	view 1 + view 2	view 2 + view 6	view 3 + view 5	Ours
Mushroom	82.86% ±5.02%	82.79% ±5.02%	82.73% ±5.12%	83.69% ±3.39%
Weather	97.03% ±0.70%	97.06% ±0.84%	96.98% ±0.91%	97.26% ±0.81%

consistently enhance classification accuracy across all datasets.

Moreover, we randomly choose two views to compare with the proposed view selection strategy to show its superiority. **view i + view j** means that we choose view i and view j as the selected candidate views. The mushroom and weather datasets are used for validation. Comparison results are report in Table 4.5, which verifies the rationality of our proposed view selection strategy.

4.6 Real-world Application of Mv-IIE

In this section, we describe an application of Mv-IIE, where a novel framework for protecting data privacy called *interval privacy-preserving* (INPP) is presented. The

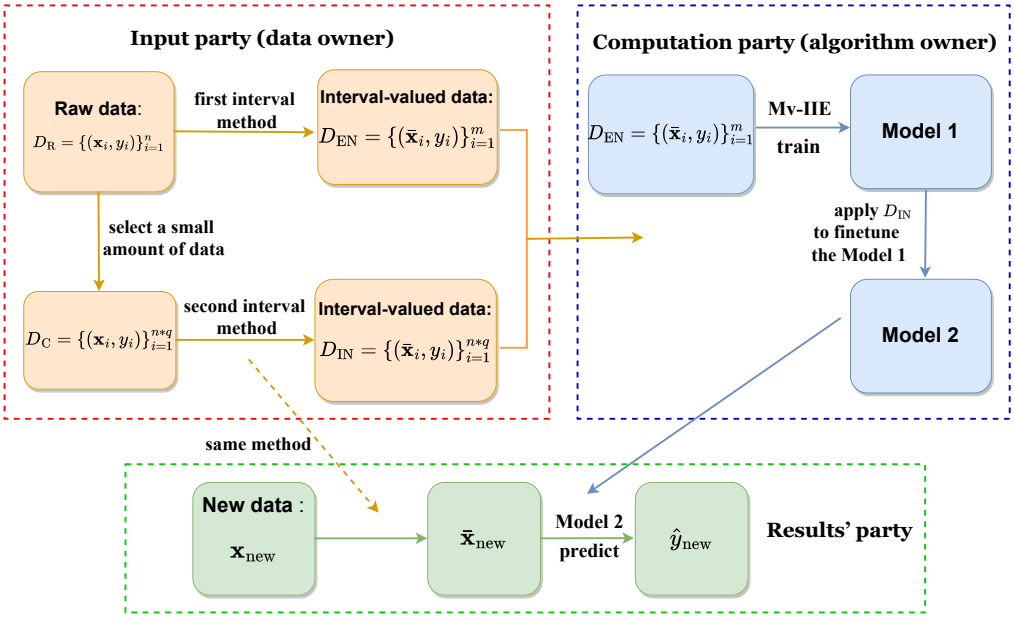


Figure 4.5: **INPP** framework: The input party (denoted in **orange**) applies two interval methods to transfer the raw data into two interval-valued datasets. The computation party (denoted in **blue**) uses D_{EN} to train Model 1 by applying Mv-IIE framework and D_{IN} is used to fine-tune Model 1 to obtain Model 2. The results' party (denoted in **green**) uses Model 2 for new data prediction.

structure of the INPP framework is shown in Figure 4.5.

There are three roles involved in each machine learning task: the input party (data owners), the computation party and the results' party. In such systems, the data owner(s) send their data to the computation party. Then, the computation party trains a model using these data and sends this model to the results' party. Finally, the results' party uses this model to predict new data. If all three roles are from the same entity, then privacy is naturally preserved. However, when these roles are from two or more entities, privacy-preserving is necessary. For example, an online clothing retailer wants to know different customers' preferences to adjust the quantity of each garment. In this situation, different customers play the first role and online clothing retailers play the second and third roles.

In the proposed framework, we denote $D_{\mathrm{R}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ as the raw data from the

data owner(s), where $\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})^\top \in \mathbb{R}^p, y_i \in [K]$. First, the data owner(s) use the intervalization approach (see Figure 4.4) to transfer D_{R} into the interval-valued data $D_{\mathrm{EN}} = \{(\bar{\mathbf{x}}_i, y_i)\}_{i=1}^{m}$ and an interval method the same as the first data-generation mechanism described in Section 4.5.2 to transfer D_{C}, which contains $n * q$ instances randomly selected from D_{R}, into the interval-valued data $D_{\mathrm{IN}} = \{(\bar{\mathbf{x}}_i, y_i)\}_{i=1}^{n*q}$. Then, the data owner(s) send these two interval-valued datasets $D_{\mathrm{EN}}, D_{\mathrm{IN}}$ to the computation party. Secondly, the computation party uses the interval-valued data D_{EN} to train Model 1 by applying Mv-IIE and the interval-valued data D_{IN} is used to fine-tune Model 1 to obtain Model 2. Then, the computation party sends Model 2 to the results' party. Finally, the results' party uses the same interval method described in Section 4.5.2 to transfer $\mathbf{x}_{\mathrm{new}}$ into $\bar{\mathbf{x}}_{\mathrm{new}}$ and uses Model 2 to predict $\bar{\mathbf{x}}_{\mathrm{new}}$ for new data prediction. According to the above methods, the internalization process of our proposed framework is irreversible and the raw data is largely compressed. Therefore, the computation party and other parties cannot obtain the raw data from D_{EN} and D_{IN}, so this process achieve the purpose of preventing data leakage. We define $\mathrm{EN} = 1 - (m + n * q)/n$, where $(m + n * q)$ is the amount of data that the computation party can receive from the data owner(s) and n is the amount of raw data. A smaller EN means the computation party will receive more data from the data owner(s), so the computation party may receive more information about the raw data. Therefore, EN can be used to measure the degree of privacy-preserving to some extent when applying INPP. Greater EN means greater privacy protection by applying INPP.

Differential privacy (DP) [43, 115] and homomorphic encryption [52, 96, 173] are common used schemes to achieve privacy-preserving. DP and homomorphic encryption can be applied to the raw data or the algorithm, but our method only applies to the raw data. DP applied to the raw data is based on data-perturbation, and homomorphic encryption is based on data-encryption, but the amount of data is not changed. Moreover,

Table 4.6: Experiment results of INPP framework on letter recognition dataset. R is equal to the ratio of the outcomes of INPP framework to the best outcome on the original dataset.

Method	L	T	q	Test accuracy	R	EN
original dataset	—-	—-	—-	95.86% ±0.19%	—-	—-
INPP	6	15	0.20	88.85% ±0.71%	92.69%	66.82%
	6	15	0.30	91.19% ±0.75%	95.13%	56.84%
	6	15	0.50	93.24% ±0.50%	97.27%	37.19%

if the keys of the encryption schemes are compromised, the information of the raw data will also be compromised. While our method compresses the raw data into interval-valued data with fewer instances through an irreversible process to protect data privacy. Further, DP and our approach can not be easily applied to image data, which is a meaningful problem worth considering in the future.

Experiments on one real-world dataset are conducted to verify the efficacy and feasibility of the INPP framework. We use four well-known machine learning methods (logistic regression, support vector machines, random forests and neural networks) to classify the original dataset and compare the best outcome of these four methods on the original dataset with the outcomes of the INPP framework. We randomly divide the original dataset (letter recognition dataset selected from the UCI Machine Learning Repository) into a raw dataset from the data owner(s) (70%) and a new dataset (30%) from the results' party. We choose $L = 6, T = 15$ and set $q = 0.20, 0.30, 0.50$. From previous results, Mv-IIE with SVM-rbf (SVM with radial basis kernel function) achieve best outcomes on the second synthetic dataset. Therefore, we use SVM-rbf as the basic classifier of Mv-IIE in this experiment. The experimental details of Mv-IIE are the same as the aforementioned. The experiment details of the four well-known machine learning methods on the original dataset are the same as the experiment details of the four baselines on the synthetic datasets.

All the experiment results are shown in Table 4.6. We note that the proposed framework can achieve 93.24% classification accuracy on the new data with R = 97.27% when $L = 6, T = 15, q = 0.5$, which demonstrates that applying the proposed framework can prevent crisp-valued data leakage while ensuring high classification accuracy of the model that has been trained by the computation party.

4.7 Summary

In this chapter, we focus on a highly challenging problem called LIND, where we aim to learn a classifier with high performance on interval-valued observations. We obtain the estimation error bound of the LINO problem based on Rademacher complexity and discuss the learnability of the underlying problem with perfect observation. Moreover, we construct a new algorithm called Mv-IIE by applying multi-view learning for interval-valued data classification. Experimental comparisons with several baselines on both synthetic and real-world datasets demonstrate the superiority of the proposed model. Finally, we detail an application of the proposed algorithm that we can prevent crisp-valued data leakage by transforming crisp-valued data into interval-valued data.

4.8 Appendix

4.8.1 Proof of Theorem 4.2

For all $\boldsymbol{h} \in \mathscr{H}$ and $\bar{\boldsymbol{h}} \in \bar{\mathscr{H}}$:

$$|R_{\mathscr{D}}(\boldsymbol{h}) - R_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})| \leq |R_{\mathscr{D}}(\boldsymbol{h}) - \widehat{R}_{\mathscr{D}}(\boldsymbol{h})| + |\widehat{R}_{\mathscr{D}}(\boldsymbol{h}) - \widehat{R}_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})| + |R_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}}) - \widehat{R}_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})|. \tag{4.6}$$

According to [101, 105] and Theorem 4.1, we have for any $\delta > 0$, with probability at least $1 - \delta/2$, the following holds for all $\boldsymbol{h} \in \mathscr{H}$ and $\bar{\boldsymbol{h}} \in \bar{\mathscr{H}}$:

$$
\begin{aligned}
|R_{\mathscr{D}}(\boldsymbol{h}) - \widehat{R}_{\mathscr{D}}(\boldsymbol{h})| &\leq 2\sqrt{2} L_\ell \widehat{\mathscr{R}}_{S_X}(\mathscr{H}) + 3C_\ell \sqrt{\frac{\log(4/\delta)}{2m}}, \\
|R_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}}) - \widehat{R}_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})| &\leq 2\sqrt{2} L_\ell \widehat{\mathscr{R}}_{S_{\bar{X}}}(\bar{\mathscr{H}}) + 3C_\ell \sqrt{\frac{\log(4/\delta)}{2m}}.
\end{aligned} \tag{4.7}
$$

Next, we consider the second term in Eq. (4.6)

$$
\begin{aligned}
|\widehat{R}_{\mathscr{D}}(\boldsymbol{h}) - \widehat{R}_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})| &= \frac{1}{m} |\sum_{i=1}^{m} [\ell(\boldsymbol{h}(\mathbf{x}_i), y_i) - \ell(\bar{\boldsymbol{h}}(\bar{\mathbf{x}}_i), y_i)]| \\
&\leq \frac{1}{m} \sum_{i=1}^{m} |[\ell(\boldsymbol{h}(\mathbf{x}_i), y_i) - \ell(\boldsymbol{h} \circ \boldsymbol{g}(\bar{\mathbf{x}}_i), y_i)]| \\
&\leq \frac{L_\ell}{m} \sum_{i=1}^{m} \| \boldsymbol{h}(\mathbf{x}_i) - \boldsymbol{h} \circ \boldsymbol{g}(\bar{\mathbf{x}}_i) \|_2 \\
&\leq \frac{L_\ell L_h}{m} \sum_{i=1}^{m} \| \mathbf{x}_i - \boldsymbol{g}(\bar{\mathbf{x}}_i) \|_2 = O(1/m^\gamma).
\end{aligned} \tag{4.8}
$$

Following from Eqs. (4.7) and (4.8), we have for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $\boldsymbol{h} \in \mathscr{H}$ and $\bar{\boldsymbol{h}} \in \bar{\mathscr{H}}$:

$$
|R_{\mathscr{D}}(\boldsymbol{h}) - R_{\bar{\mathscr{D}}}(\bar{\boldsymbol{h}})| \leq 2\sqrt{2} L_\ell (\widehat{\mathscr{R}}_{S_{\bar{X}}}(\bar{\mathscr{H}}) + \widehat{\mathscr{R}}_{S_X}(\mathscr{H})) + 6C_\ell \sqrt{\frac{\log(4/\delta)}{2m}} + O(1/m^\gamma).
$$

4.8.2 Proof for Theorem 4.3

For any $\boldsymbol{h}_v \in \mathscr{H}_v$, we let

$$
\begin{aligned}
\boldsymbol{h}_v : &\mathscr{X}_v \to \mathbb{R}^K \\
&\mathbf{x}_i^v \to (h_{v1}(\mathbf{x}_i^v), \cdots, h_{vK}(\mathbf{x}_i^v))^\top.
\end{aligned}
$$

Without loss of generality, we suppose that $\sum_{k=1}^{K} h_{vk}(\mathbf{x}_i^v) = 1$ and the predict function f_v of $\boldsymbol{h}_v$ is defined as

$$
f_v(\mathbf{x}_i^v) = \arg \max_{1 \leq k \leq K} h_{vk}(\mathbf{x}_i^v).
$$

Then, for any $\boldsymbol{h}_{\text{co}} \in \mathscr{H}_{\text{co}}$, we let

$$\begin{aligned}\boldsymbol{h}_{\text{co}} : &\mathscr{X}^{\text{Mv}} \to \mathbb{R}^K \\ &\mathbf{X}_i = (\mathbf{x}_i^1, \cdots, \mathbf{x}_i^c) \to (h_{\text{co}}^1(\mathbf{X}_i), \cdots, h_{\text{co}}^K(\mathbf{X}_i))^\top,\end{aligned}$$

where $\boldsymbol{h}_{\text{co}}^q(\mathbf{X}_i) = \sum_{v=1}^{c} \mathbf{w}_v^{q\top} \boldsymbol{h}_v(\mathbf{x}_i^v), \mathbf{w}_v^q = (w_{v1}^q, \cdots, w_{vK}^q)^\top$ and without loss of generality, we suppose $\sum_{q=1}^{K} h_{\text{co}}^q(\mathbf{X}_i) = 1$. Therefore, we have $\sup_{\boldsymbol{h}_{\text{co}} \in \mathscr{H}_{\text{co}}} \| \boldsymbol{h}_{\text{co}} \|_\infty \leq 1$. The predict function f_{co} of $\boldsymbol{h}_{\text{co}}$ is defined as

$$f_{\text{co}}(\mathbf{X}_i) = \arg \max_{1 \leq q \leq K} h_{\text{co}}^q(\mathbf{X}_i).$$

Without loss of generality, we suppose $\text{err}(f_1) \leq \cdots \leq \text{err}(f_c)$. First, we consider the case where $c = 2$. Then, we provide an upper bound on the error rate of f_{co}.

$$\begin{aligned}\text{err}(f_{\text{co}}) &= \mathbb{P}(f_{\text{co}}(\mathbf{X}) \neq y | \mathbf{X} \in \mathbf{D}_{\mathscr{F}}^C(f_1, f_2)) + \mathbb{P}(f_{\text{co}}(\mathbf{X}) \neq y | \mathbf{X} \in \mathbf{D}_{\mathscr{F}}(f_1, f_2)) \\ &\leq \frac{1}{2}[\text{err}(f_1) + \text{err}(f_2) - \mathbb{P}_{\mathscr{D}}(\mathbf{X} \in \mathbf{D}_{\mathscr{F}}(f_1, f_2))] + \mathbb{P}(f_{\text{co}}(\mathbf{X}) \neq y | X \in \mathbf{D}_{\mathscr{F}}(f_1, f_2)), \qquad (4.9)\end{aligned}$$

where $\mathbf{D}_{\mathscr{F}}^C(f_1, f_2)$ is denoted as the complement set of $\mathbf{D}_{\mathscr{F}}(f_1, f_2)$. According to Eq. (4.9) and $\text{err}(f_1) \leq \text{err}(f_2)$, if $\mathbb{P}(f_{\text{co}}(\mathbf{X}) \neq y | \mathbf{X} \in \mathbf{D}_{\mathscr{F}}(f_1, f_2)) \leq \frac{1}{2}[\text{err}(f_1) - \text{err}(f_2) + \mathbb{P}_{\mathscr{D}}(\mathbf{X} \in \mathbf{D}_{\mathscr{F}}(f_1, f_2))]$, we have $\text{err}(f_{\text{co}}) \leq \text{err}(f_1)$. Next, we consider the case where $c > 2$. For $c > 2$, we have $\boldsymbol{h}_{\text{co}} \in \mathscr{H}_{\text{co}}$,

$$h_{\text{co}}^q(\mathbf{X}) = \sum_{v=1}^{k+1} \mathbf{w}_v^{q\top} \boldsymbol{h}_v(\mathbf{x}^v) = \mathbf{w}_1^{q\top} \boldsymbol{h}_1(\mathbf{x}^1) + \sum_{v=2}^{c} \mathbf{w}_v^{q\top} \boldsymbol{h}_v(\mathbf{x}^v).$$

So exists $\alpha_q \in \mathbb{R}_+$, such that $\sum_{q=1}^{K} \alpha_q \sum_{v=2}^{c} \mathbf{w}_v^{q\top} \boldsymbol{h}_v(\mathbf{x}^v) = 1$, then exists $\boldsymbol{h}_{\text{co}}^{c-1} \in \mathscr{H}_{\text{co}}^{c-1}(\mathbf{x}^2, \cdots, \mathbf{x}^c)$, where

$$h_{\text{co}}^{c-1,q} = \alpha_q \sum_{v=2}^{c} \mathbf{w}_v^{q\top} \boldsymbol{h}_v(\mathbf{x}^v).$$

We combine the last $c-1$ views i.e.,

$$\mathbf{X}' = (\mathbf{x}^2, \cdots, \mathbf{x}^c), \mathbf{X} = (\mathbf{x}^1, \mathbf{X}').$$

So exists

$$\boldsymbol{h}_{\text{co}}^{c-1} \in \mathscr{H}_{\text{co}}^{c-1}(\mathbf{x}^2, \cdots, \mathbf{x}^c) \subset \mathscr{H}(\mathbf{X}^{'}),$$

such that

$$h_{\text{co}}^{q}(\mathbf{X}) = \mathbf{w}_1^{q\top} \boldsymbol{h}_1(\mathbf{x}^1) + \frac{1}{\alpha_q} h_{\text{co}}^{c-1,q}(\mathbf{X}^{'}).$$

Therefore we have $\boldsymbol{h}_{\text{co}} \in \mathscr{H}_{\text{co}}(\mathbf{x}^1, \mathbf{X}^{'})$. Let $f_{\text{co}}^{c-1}(\mathbf{X}) = \arg\max_{1\leq q\leq K} h_{\text{co}}^{c-1,q}(\mathbf{X})$ denoted as the predict function of $\boldsymbol{h}_{\text{co}}^{c-1}$. Because the conclusion is true when $c = 2$, so exists $M \in (0,1)$, such that

$$\text{if } \mathbb{P}(f_{\text{co}}(\mathbf{X}) \neq y | \mathbf{X} \in \mathbf{D}_{\mathscr{F}}(f_1, f_{\text{co}}^{c-1})) \leq M,$$

we have $\text{err}(f_{\text{co}}) \leq \text{err}(f_1)$. Because $\mathbf{D}_{\mathscr{F}}(f_1, f_{\text{co}}^{c-1}) \subset \mathbf{D}_{\mathscr{F}}(f_1, \cdots, f_c)$, so $\mathbb{P}(f_{\text{co}}(\mathbf{X}) \neq y | \mathbf{X} \in \mathbf{D}_{\mathscr{F}}(f_1, f_{\text{co}}^{c-1})) \leq \mathbb{P}(f_{\text{co}}(\mathbf{X}) \neq y | \mathbf{X} \in \mathbf{D}_{\mathscr{F}}(f_1, \cdots, f_c))$. Therefore, the conclusion is true when $c > 2$ which yields the result.

4.8.3 Proof of Theorem 4.4

Because $\sum_{q=1}^{K}\sum_{v=1}^{c}\sum_{k=1}^{K} w_{vk}^{q} h_{vk}(\mathbf{x}_i^v) = 1$ and for any $v \in [c], k \in [K], 0 \leq h_{vk}(\mathbf{x}_i^v) \leq 1$, so $\sum_{q=1}^{K}\sum_{v=1}^{c}\sum_{k=1}^{K} w_{vk}^{q} \leq 1$. Then,

$$\begin{aligned}
\mathscr{R}_{S_X^{\text{Mv}}}(\mathscr{H}_{\text{co}}) &= \frac{1}{m}\mathbb{E}_{\mathscr{D}^{\text{Mv}},\boldsymbol{\sigma}}[\sup_{\boldsymbol{h}_{\text{co}}\in\mathscr{H}_{\text{co}}} \sum_{i=1}^{m}\sum_{q=1}^{K} \sigma_{iq} h_{\text{co}}^{q}(\mathbf{X}_i)] \\
&= \frac{1}{m}\mathbb{E}_{\mathscr{D}^{\text{Mv}},\boldsymbol{\sigma}}[\sup_{\boldsymbol{h}_v\in\mathscr{H}_v, \|\mathbf{W}\|_2\leq\Lambda} \sum_{i=1}^{m}\sum_{q=1}^{K} \sigma_{iq} \sum_{v=1}^{c} \mathbf{w}_v^{q\top} \boldsymbol{h}_v(\mathbf{x}_i^v)] \\
&= \frac{1}{m}\mathbb{E}_{\mathscr{D}^{\text{Mv}},\boldsymbol{\sigma}}[\sup_{\boldsymbol{h}_v\in\mathscr{H}_v, \|\mathbf{W}\|_2\leq\Lambda} \sum_{v,q,k} w_{vk}^{q} \sum_{i=1}^{m} \sigma_{iq} h_{vk}(\mathbf{x}_i^v)] \\
&\leq \frac{1}{m}\mathbb{E}_{\mathscr{D}^{\text{Mv}},\boldsymbol{\sigma}}[\sup_{\boldsymbol{h}_v\in\mathscr{H}_v} \max_{v\in[c],q\in[K]} \sum_{i=1}^{m}\sum_{k=1}^{K} \sigma_{ik} h_{vk}(\mathbf{x}_i^v)] \\
&\leq \max_{v\in[c]} \mathscr{R}_{S_{X^v}}(\mathscr{H}_v) \\
&= \min_{v\in[c]} \mathscr{R}_{S_{X^v}}(\mathscr{H}_v) + \max_{v\in[c]} \mathscr{R}_{S_{X^v}}(\mathscr{H}_v) - \min_{v\in[c]} \mathscr{R}_{S_{X^v}}(\mathscr{H}_v)
\end{aligned}$$

CHAPTER 5

DOMAIN ADAPTATION WITH INTERVAL-VALUED OBSERVATIONS

5.1 Introduction

Traditional machine learning theories, as expounded by Shalev-Shwartz and Ben-David [133], are grounded in two fundamental assumptions: 1) that the data in both training and test sets are drawn from the same underlying distribution, and 2) that an ample supply of labeled training data is accessible. To ease the effects of these two assumptions, the research community has delved into the problem of *unsupervised domain adaptation* (UDA) [112]. UDA centers its efforts on augmenting the performance of an unlabeled target domain by harnessing insights from a source domain enriched with an adequate quantity of labeled observations. Drawing from the foundational concepts of classical domain adaptation theory [6], numerous well-regarded UDA algorithms [93] have been conceived to tackle the issue of domain shift. These UDA algorithms have demonstrated remarkable efficacy across a diverse array of application in spanning natural language

processing [89], and computer vision [94].

Most existing UDA works [93] predominantly concentrate on addressing large-scale image data characterized by crisp-valued features. However, in real-world applications, datasets often encapsulate uncertainties and imprecisions that cannot be adequately represented by single-point values. Interval-valued observations [8], which express a range or uncertainty associated with each data point, offer a more faithful representation of such inherent uncertainties. Consider medical data where patient health parameters fluctuate within a certain range, or environmental monitoring data capturing fluctuating sensor readings. In these contexts, interval-valued observations become indispensable for providing a nuanced and realistic depiction of the underlying phenomena. For example, Table 3.4 shows a real-world interval-valued dataset described by five interval-valued features and one category variable. Therefore, the utilization of interval-valued datasets not only aligns with the inherent nature of uncertainties present in many real-world scenarios but also facilitates more accurate and robust analyses, contributing to advancements in data-driven insights. Consequently, in this chapter, we focus on a more realistic and challenging problem named *domain adaptation with interval-valued observations* (DAINO). Within the DAINO context, we confront the scenario of having a source domain enriched with an adequate quantity of labeled observations and an unlabeled target domain, both exclusively comprising interval-valued observations. Our goal is to adapt the model trained on the source domain for the unlabeled target domain by minimizing domain shift between both domain with interval-valued observations.

Clearly, established and widely recognized UDA methodologies [48] cannot be directly applied to handle DAINO problems due to the distinctive nature of interval-valued data. An ostensibly straightforward solution entails taking the midpoint of the interval-valued features or considering the upper and lower bounds of the interval-valued features as two separate crisp-valued features, effectively converting the interval-valued data into

crisp-valued data. Subsequently, conventional UDA techniques can be employed on this converted crisp-valued data. However, this approach neglects the intrinsic uncertainty information inherently embedded within interval-valued data, ultimately resulting in suboptimal performance when addressing the DAINO problem. In this chapter, we embark on a comprehensive analysis and resolution of the presented problem from both theoretical and algorithmic perspectives. Drawing inspiration from classical domain adaptation theory [6] and building upon our previous work in imprecise data analysis [98], we derive an upper bound on the risk within the interval-valued target domain. This bound encapsulates the risk within a target domain characterized by interval-valued observations through three primary components: (i) the risk within the interval-valued source domain; (ii) the distribution discrepancy between the interval-valued source domain and target domain; and (iii) the combined error of the ideal joint hypothesis for the source and target domains. According to our theoretical analysis, the DAINO problem presents three pivotal challenges: 1) how to make full use of the inherent uncertainty information in interval-valued data; 2) how to align the distribution between the interval-valued source and target domains; 3) how to improve class discriminability of the target domain.

Guided by our theoretical analysis of the DAINO problem, we develop a new domain adaptation model, called SP-TSF, based on fuzzy techniques to address the aforementioned three challenges. Fuzzy set theory [68], rough set theory [116], possibility theory [41], and belief function theory [132] are all mathematical frameworks used for handling uncertainty and imprecision in different ways. Compared with other theories, fuzzy set theory is known for its simplicity, interpretability, and ability to handle gradual transitions. Fuzzy techniques, constructed based on fuzzy set theory, are flexible and can model complex relationships, nonlinearities, and intricate patterns in data. Moreover, it is relatively straightforward to implement, and its principles are easily comprehensible.

While rough set theory works with discrete sets and focuses on discernibility, possibility theory primarily deals with possibility measures, and belief function theory excels in handling situations where there is uncertainty, ignorance, and conflict in evidence. Therefore, in DAINO setting, we apply fuzzy techniques to address the uncertainty with interval-valued data. The applied fuzzy techniques includes fuzzy sets and a fuzzy rule-based model.

Regarding the first challenge, we adopt the T-S fuzzy rule-based model as the foundational model structure to capture the inherent uncertainty intrinsic to interval-valued data. The T-S fuzzy model [2], often extended as the TSK fuzzy model, represents a category of fuzzy logic systems extensively utilized in modeling and control applications. This model has garnered widespread recognition for its aptitude in handling uncertainty within the realm of transfer learning, as it is grounded in fuzzy logic, providing a systematic framework for the representation and reasoning of uncertainty. For example, Zuo *et al.* [181] introduced an innovative approach anchored in the T-S fuzzy model, combining an infinite Gaussian mixture model with active learning to enhance model performance and generalizability. Li *et al.* [80] formulated a novel model leveraging a deep neural network equipped with T-S fuzzy rules to tackle a challenging problem in multi-source domain adaptation, a scenario where source data is unavailable. These pioneering works collectively underscore the good properties of the T-S fuzzy model in dealing with uncertain problems. Moreover, a fuzzy transformation function is used to extract valuable crisp-valued information from the interval-valued observation. This fuzzy transformation function was designed in Chapter 4, wherein its efficacy in handling interval-valued data was rigorously validated, resulting in commendable performance outcomes.

To align the distribution between the source and target domains, characterized by crisp-valued observations, numerous existing UDA works turn to minimize the distri-

bution discrepancy between the source and target domains based on different integral probability metrics. Widely used metrics including maximum mean discrepancy [93] and Wasserstein distances [135]. Subsequently, we introduce a novel metric, an extension of the conventional maximum mean discrepancy, tailored to augment distribution alignment across the interval-valued source and target domains. As for the last challenge, a self-supervised pseudo-labeling strategy based on deep clustering [16] is developed to solve it. In its implementation, the pseudo-labeling strategy is initially employed to procure dependable pseudo labels for the target data. Following this, the target data with distributed pseudo labels are trained on the fuzzy rule-based classifiers to improve class discriminability of the interval-valued target domain.

To evaluate the efficacy of our proposed model, we conducted comparative assessments against several SOTA UDA algorithms. Our evaluation involved two synthetic datasets and six real-world tasks. The findings consistently reveal that the proposed method surpasses all competing baselines, demonstrating superior performance across both synthetic and real-world datasets. Furthermore, an ablation study is conducted to verify the rationality of our model's construction. Especially, we remove all the applied fuzzy techniques in our model and use none-fuzzy modules to replace them. The experimental results prove the excellent ability of fuzzy techniques in solving the DAINO problem. This can be attributed to the substantial uncertainty inherent in interval-valued data. In comparison to conventional non-fuzzy methods, fuzzy techniques offer pronounced advantages in navigating and mitigating these uncertainties.

The contributions of this chapter are summarized as follows.

1. We are the first to identify a challenging problem called DAINO, where we try to adapt the model trained on the source domain for the unlabeled target domain by minimizing domain shift between both domain with interval-valued observations. A theoretical bound on the target domain is provided as the theoretical analysis of

the DAINO problem.

2. A new model called SP-TSF is developed to solve the DAINO problem. Our model leverages the T-S fuzzy rule-based model as its foundational structure, aimed at capturing the intrinsic uncertainty inherent to interval-valued data. Furthermore, it introduces a novel integral probability metric design to align the distribution characteristics between the interval-valued source and target domains. Additionally, in our model, a deep clustering-based self-supervised pseudo-labeling strategy is developed to enhance class discriminability of the interval-valued target domain.

3. An extensive set of experiments shows significantly improved classification accuracy on unlabeled target domain with interval-valued observations, evidenced by comparison with several SOTA traditional UDA algorithms on both synthetic and real-world datasets.

5.2 Problem Setting

In this section, we introduce the DAINO problem. $\bar{\mathscr{X}} \subset \bar{\mathbb{R}}^p$ is denoted as the input space with the interval-valued observations and $\bar{\mathscr{D}}$ as the distribution of the input space $\bar{\mathscr{X}}$. Next, the score function of $\bar{\mathscr{X}}$ is defined as:

$$\begin{aligned} &\mathbf{f}(\bar{\mathbf{x}}) : \bar{\mathscr{X}} \to \mathbb{R}^K \\ &(\bar{x}_1, \cdots, \bar{x}_p)^\top \to (f_1(\bar{\mathbf{x}}), \cdots, f_K(\bar{\mathbf{x}}))^\top. \end{aligned} \tag{5.1}$$

If $\bar{\mathbf{x}} \in \bar{\mathscr{X}}$ belongs to the C-th class, then $f_C(\bar{\mathbf{x}}) = 1$ and for any $k \in [K], k \neq C, f_k(\bar{\mathbf{x}}) = 0$. Then, $\bar{\mathscr{X}}_S, \bar{\mathscr{X}}_T$ and $\bar{\mathscr{D}}_S, \bar{\mathscr{D}}_T$ denotes the input spaces and the distribution of the source and target domains, respectively.

Remark: We give this definition of $\mathbf{f}(\cdot)$ because our focus is on classification problems. It is noteworthy that, in certain specific cases, interval-valued data may not be separable.

However, this challenge is unavoidable even in traditional machine learning classification problems with crisp-valued data, as the Bayes error rate [58] is always present. Our objective is to train the optimal classifier to differentiate between various categories of interval-valued data in the target domain.

Let $\mathscr{H}$ be the hypothesis set and for any $\boldsymbol{h} \in \mathscr{H}$,

$$\begin{aligned} \boldsymbol{h}(\bar{\mathbf{x}}) : \bar{\mathscr{X}} &\to \mathbb{R}^K \\ (\bar{x}_1, \cdots, \bar{x}_p)^\top &\to (h_1(\bar{\mathbf{x}}), \cdots, h_K(\bar{\mathbf{x}}))^\top. \end{aligned} \tag{5.2}$$

Without loss of generality, suppose $\sum_{k=1}^{K} h_k(\bar{\mathbf{x}}_i) = 1$ and each $h_k(\bar{\mathbf{x}}_i)$ represents the probability of the instance $\bar{\mathbf{x}}_i$ belongs to the k-th category. Therefore, we have $\sup_{\boldsymbol{h} \in \mathscr{H}} \| \boldsymbol{h} \|_\infty \leq 1$. The loss function of $\boldsymbol{h}$ is defined as,

$$\ell : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}_+. \tag{5.3}$$

Let $\mathscr{L}_{\mathscr{H}} = \{\ell(\boldsymbol{h}^{'}(\bar{\mathbf{x}}), \boldsymbol{h}(\bar{\mathbf{x}})) : \boldsymbol{h}, \boldsymbol{h}^{'} \in \mathscr{H}, \bar{\mathbf{x}} \in \bar{\mathscr{X}}\}$ be the class of loss functions associated with $\mathscr{H}$. Next, we have the risks on the source and target domains, denoted as:

$$\begin{aligned} R_{\bar{\mathscr{D}}_S}(\boldsymbol{h}) &= \mathbb{E}_{\bar{\mathscr{D}}_S}[\ell(\boldsymbol{h}(\bar{\mathbf{x}}), \mathbf{f}_S(\bar{\mathbf{x}}))], \\ R_{\bar{\mathscr{D}}_T}(\boldsymbol{h}) &= \mathbb{E}_{\bar{\mathscr{D}}_T}[\ell(\boldsymbol{h}(\bar{\mathbf{x}}), \mathbf{f}_T(\bar{\mathbf{x}}))]. \end{aligned} \tag{5.4}$$

The definition of the DAINO problem is based on the definition of the ordinary domain adaptation problem:

Definition 5.1 (Domain Adaptation with Interval-valued Observations)**.** Let $\bar{X}_s = \{\bar{\mathbf{x}}_i^s\}_{i=1}^{n_s}$, $\bar{X}_t = \{\bar{\mathbf{x}}_i^t\}_{i=1}^{n_t}$ be i.i.d. observations from the interval probability distribution $\bar{\mathscr{D}}_S$ and $\bar{\mathscr{D}}_T$, respectively, where $\bar{\mathbf{x}}_i^s \in \bar{\mathscr{X}}_S$ and $\bar{\mathbf{x}}_i^t \in \bar{\mathscr{X}}_T$. $Y_s = \{y_i^s\}_{i=1}^{n_s}$ is the ground-truth label set corresponding to $\bar{X}_s$, and $y_i^s = \arg\max_{k \in [K]} f_k^S(\bar{\mathbf{x}}_i^s) \in \mathscr{Y}, i = 1, \cdots, n_s$. Here, $\mathscr{Y} = [K]$ is the label space. $\bar{\mathscr{S}} = \langle \bar{X}_s, Y_s \rangle$ denotes the source domain, and $\bar{\mathscr{T}} = \langle \bar{X}_t \rangle$ denotes the target domain, which only contains unlabeled samples. Our focus is on unsupervised domain

adaptation scenarios. Thus, the aim with domain adaptation given interval-valued observations is to train a classifier with $\bar{\mathscr{S}}$ and $\bar{\mathscr{T}}$ to accurately label each instance i.i.d. drawn from $\bar{\mathscr{D}}_T$. That is, our aim is to train a classifier $\boldsymbol{h}^t \in \mathscr{H}$ with $\bar{\mathscr{S}}$ and $\bar{\mathscr{T}}$ s.t. $\boldsymbol{h}^t = \arg\min_{\boldsymbol{h}\in\mathscr{H}} R_{\bar{\mathscr{D}}_T}(\boldsymbol{h})$.

5.3 Theoretical Analysis of DAINO

In this section, we present the theoretical analysis of the DAINO problem. Note that all proofs are provided in the Appendix. First, we define the discrepancy distance between the interval distributions of the source and target domains.

Definition 5.2. Let $\bar{\mathscr{S}}$ and $\bar{\mathscr{T}}$ be the source and target domain over $\bar{\mathscr{X}} \times \mathscr{Y}$, respectively. $\mathscr{Y} = [K]$ is the label space. The discrepancy distance between the two interval distributions $\bar{\mathscr{D}}_S$ and $\bar{\mathscr{D}}_T$ with respect to $\bar{\mathbf{x}}^s, \bar{\mathbf{x}}^t$ is defined as

$$\mathrm{disc}(\bar{\mathscr{D}}_S, \bar{\mathscr{D}}_T) = \sup_{(\boldsymbol{h},\boldsymbol{h}')\in\mathscr{H}\times\mathscr{H}} |\mathbb{E}_{\bar{\mathscr{D}}_S}[\ell(\boldsymbol{h},\boldsymbol{h}')] - \mathbb{E}_{\bar{\mathscr{D}}_T}[\ell(\boldsymbol{h},\boldsymbol{h}')]|. \tag{5.5}$$

We denote

$$\mathbf{x}^{\mathrm{s,l}} = (x_1^{\mathrm{s,l}}, \cdots, x_p^{\mathrm{s,l}})^\top \sim \mathscr{D}_S^{\mathrm{l}}, \mathbf{x}^{\mathrm{s,r}} = (x_1^{\mathrm{s,r}}, \cdots, x_p^{\mathrm{s,r}})^\top \sim \mathscr{D}_S^{\mathrm{r}},$$
$$\mathbf{x}^{\mathrm{t,l}} = (x_1^{\mathrm{t,l}}, \cdots, x_p^{\mathrm{t,l}})^\top \sim \mathscr{D}_T^{\mathrm{l}}, \mathbf{x}^{\mathrm{t,r}} = (x_1^{\mathrm{t,r}}, \cdots, x_p^{\mathrm{t,r}})^\top \sim \mathscr{D}_T^{\mathrm{r}}.$$

Then, we introduce the following lemma.

Lemma 5.1. *Let $\bar{\mathscr{S}}$ and $\bar{\mathscr{T}}$ be the source and target domain over $\bar{\mathscr{X}} \times \mathscr{Y}$, and let* $\mathrm{disc}(\bar{\mathscr{D}}_S, \bar{\mathscr{D}}_T)$ *be the discrepancy distance between the two interval distributions $\bar{\mathscr{D}}_S$ and $\bar{\mathscr{D}}_T$. Then, we have*

$$\mathrm{disc}(\bar{\mathscr{D}}_S, \bar{\mathscr{D}}_T) \leq \frac{1}{2}\mathrm{disc}(\mathscr{D}_S^{\mathrm{l}}, \mathscr{D}_T^{\mathrm{l}}) + \frac{1}{2}\mathrm{disc}(\mathscr{D}_S^{\mathrm{r}}, \mathscr{D}_T^{\mathrm{r}}). \tag{5.6}$$

where $\mathrm{disc}(\mathscr{D}_S^{\mathrm{l}}, \mathscr{D}_T^{\mathrm{l}})$ *is the discrepancy distance between the two distributions $\mathscr{D}_S^{\mathrm{l}}$ and $\mathscr{D}_T^{\mathrm{l}}$,*

$$\mathrm{disc}(\mathscr{D}_S^{\mathrm{l}}, \mathscr{D}_T^{\mathrm{l}}) = \sup_{(\boldsymbol{h},\boldsymbol{h}')\in\mathscr{H}\times\mathscr{H}} |\mathbb{E}_{\mathscr{D}_S^{\mathrm{l}}}[\ell(\boldsymbol{h},\boldsymbol{h}')] - \mathbb{E}_{\mathscr{D}_T^{\mathrm{l}}}[\ell(\boldsymbol{h},\boldsymbol{h}')]|. \tag{5.7}$$

The definition of $\mathrm{disc}(\mathscr{D}_S^{\mathrm{r}}, \mathscr{D}_T^{\mathrm{r}})$ *is same as* $\mathrm{disc}(\mathscr{D}_S^{\mathrm{l}}, \mathscr{D}_T^{\mathrm{l}})$.

According to Theorem 4.1 and Definition 5.2, we can directly prove the following lemma.

Lemma 5.2. *Suppose that* $\sup_{\|\boldsymbol{h}\|_\infty \le 1} \max_y \ell(\boldsymbol{h}, y) \le C_\ell$*, and all functions in* $\mathscr{L}_{\mathscr{H}}$ *are* L_ℓ*-Lipschitz functions. Let* $\bar{\mathscr{D}}$ *be an interval distribution over* $\bar{\mathscr{X}}$ *and we denote* $\widehat{\mathrm{disc}}(\bar{\mathscr{D}}, \bar{S}_{\bar{X}}) = \sup_{(\boldsymbol{h}, \boldsymbol{h}') \in \mathscr{H} \times \mathscr{H}} |\mathbb{E}_{\bar{\mathscr{D}}}[\ell(\boldsymbol{h}, \boldsymbol{h}')] - \widehat{\mathbb{E}}_{\bar{S}_{\bar{X}}}[\ell(\boldsymbol{h}, \boldsymbol{h}')]|$*. Then, for any* $\delta > 0$ *with a probability of at least* $1 - \delta$ *over the choice of sample* $\bar{S}_{\bar{X}}$*, we have*

$$\widehat{\mathrm{disc}}(\bar{\mathscr{D}}, \bar{S}_{\bar{X}}) \le 2\sqrt{2} L_\ell \widehat{\mathscr{R}}_{\bar{S}_{\bar{X}}}(\mathscr{H}) + 3C_\ell \sqrt{\frac{\log(2/\delta)}{2m}}. \tag{5.8}$$

Using this result, we can further prove the following corollary. Let $S_u = \{\bar{\mathbf{x}}_i^{\mathrm{s}}\}_{i=1}^{m_s}$ and $T_u = \{\bar{\mathbf{x}}_i^{\mathrm{t}}\}_{i=1}^{m_t}$ be two samples of size m_s and m_t drawn i.i.d. from $\bar{\mathscr{D}}_S$ and $\bar{\mathscr{D}}_T$.

Corollary 5.1. *Let* $\bar{\mathscr{S}}$ *and* $\bar{\mathscr{T}}$ *be the source and target domain over* $\bar{\mathscr{X}} \times \mathscr{Y}$*, respectively. Suppose that* $\sup_{\|\boldsymbol{h}\|_\infty \le 1} \max_y \ell(\boldsymbol{h}, y) \le C_\ell$*, and all functions in* $\mathscr{L}_{\mathscr{H}}$ *are* L_ℓ*-Lipschitz functions. Let* $\widehat{\mathrm{disc}}(S_u, T_u) = \sup_{(\boldsymbol{h}, \boldsymbol{h}') \in \mathscr{H} \times \mathscr{H}} |\widehat{\mathbb{E}}_{S_u}[\ell(\boldsymbol{h}, \boldsymbol{h}')] - \widehat{\mathbb{E}}_{T_u}[\ell(\boldsymbol{h}, \boldsymbol{h}')]|$ *denote the empirical discrepancy distance between the samples* S_u *and* T_u*. Then, for any* $\delta > 0$*, with a probability of at least* $1 - \delta$ *over the choice of samples* S_u *and* T_u*, we have*

$$\mathrm{disc}(\bar{\mathscr{D}}_S, \bar{\mathscr{D}}_T) \le \widehat{\mathrm{disc}}(S_u, T_u) + 2\sqrt{2} L_\ell (\widehat{\mathscr{R}}_{S_u}(\mathscr{H}) + \widehat{\mathscr{R}}_{T_u}(\mathscr{H})) + 3C_\ell (\sqrt{\frac{\log(4/\delta)}{2m_s}} + \sqrt{\frac{\log(4/\delta)}{2m_t}}). \tag{5.9}$$

To illustrate what this implies for the model's generalization guarantee, Theorem 5.1 covers the source and target error function using the discrepancy distance.

Theorem 5.1. *Let* $\bar{\mathscr{S}}$ *and* $\bar{\mathscr{T}}$ *be the source and target domain over* $\bar{\mathscr{X}} \times \mathscr{Y}$*, respectively. Suppose that* $\sup_{\|\boldsymbol{h}\|_\infty \le 1} \max_y \ell(\boldsymbol{h}, y) \le C_\ell$*, and all functions in* $\mathscr{L}_{\mathscr{H}}$ *are* L_ℓ*-Lipschitz functions.* $S_u^{\mathrm{l}} = \{\mathbf{x}_i^{\mathrm{s,l}}\}_{i=1}^{m_s}, S_u^{\mathrm{r}} = \{\mathbf{x}_i^{\mathrm{s,r}}\}_{i=1}^{m_s}$ *and* $T_u^{\mathrm{l}} = \{\mathbf{x}_i^{\mathrm{t,l}}\}_{i=1}^{m_t}, T_u^{\mathrm{r}} = \{\mathbf{x}_i^{\mathrm{t,r}}\}_{i=1}^{m_t}$*.* $\boldsymbol{h}^* = \arg\min_{\boldsymbol{h} \in \mathscr{H}} R_{\bar{\mathscr{D}}_S}(\boldsymbol{h}) +$

$R_{\bar{\mathscr{D}}_T}(\boldsymbol{h})$ denotes the ideal joint hypothesis for the source and target domains. Then, with a probability of at least $1-\delta$, the following holds for any $\boldsymbol{h} \in \mathscr{H}$, we have

$$
\begin{aligned}
(5.10) \qquad R_{\bar{\mathscr{D}}_T}(\boldsymbol{h}) &\leq R_{\bar{\mathscr{D}}_S}(\boldsymbol{h}) + \mathrm{disc}(\bar{\mathscr{D}}_S, \bar{\mathscr{D}}_T) + \lambda \\
&\leq \widehat{R}_{\bar{\mathscr{D}}}(\boldsymbol{h}) + \frac{1}{2}\widehat{\mathrm{disc}}(S_u^{\mathrm{l}}, T_u^{\mathrm{l}}) + \frac{1}{2}\widehat{\mathrm{disc}}(S_u^{\mathrm{r}}, T_u^{\mathrm{r}}) + 2\sqrt{2}L_\ell(2\widehat{\mathscr{R}}_{S_u}(\mathscr{H}) \\
&\quad + \widehat{\mathscr{R}}_{T_u}(\mathscr{H})) + 3C_\ell(2\sqrt{\frac{\log(6/\delta)}{2m_s}} + \sqrt{\frac{\log(6/\delta)}{2m_t}}) + \lambda,
\end{aligned}
$$

where $\lambda = R_{\bar{\mathscr{D}}_T}(\boldsymbol{h}^*) + R_{\bar{\mathscr{D}}_S}(\boldsymbol{h}^*)$.

Applying the triangle inequality to ℓ and according to Eqs. (4.2)(5.6)(5.9), we can easily prove Theorem 5.1. Theorem 5.1 gives an upper bound of the risk in the target domain. Based on the analysis of Theorem 4.1, if $\widehat{\mathscr{R}}_{S_u}(\mathscr{H}) = O(1/\sqrt{m_s})$ and $\widehat{\mathscr{R}}_{T_u}(\mathscr{H}) = O(1/\sqrt{m_t})$, we notice that as $m_s, m_t \to \infty$, $R_{\bar{\mathscr{D}}_T}(\boldsymbol{h}) \to \widehat{R}_{\bar{\mathscr{D}}}(\boldsymbol{h}) + \frac{1}{2}\widehat{\mathrm{disc}}(S_u^{\mathrm{l}}, T_u^{\mathrm{l}}) + \frac{1}{2}\widehat{\mathrm{disc}}(S_u^{\mathrm{r}}, T_u^{\mathrm{r}}) + \lambda$. Therefore, there are three main parts that need to be considered to reduce the risk on the target domain: (i) the empirical risk in the source samples ($\widehat{R}_{\bar{\mathscr{D}}}(\boldsymbol{h})$); (ii) the empirical discrepancy distance between $S_u^{\mathrm{l}}, S_u^{\mathrm{r}}$ and $T_u^{\mathrm{l}}, T_u^{\mathrm{r}}$ ($\frac{1}{2}\widehat{\mathrm{disc}}(S_u^{\mathrm{l}}, T_u^{\mathrm{l}}) + \frac{1}{2}\widehat{\mathrm{disc}}(S_u^{\mathrm{r}}, T_u^{\mathrm{r}})$); and (iii) the combined error λ of the ideal joint hypothesis $\boldsymbol{h}^*$ for the source and target domains.

According to Theorem 4.1, the first component of Eq. (5.10) can be minimized effectively when a sufficient number of reliable labeled source samples are available. Different from the traditional UDA problem, the main challenge to train a reliable source model (minimize the first term) is how to deal with the inherent uncertainty information present in interval-valued data. Regarding the second component, it necessitates the design of appropriate metrics to approximate the empirical discrepancy distance between the interval-valued source and target samples. Lastly, as elucidated by the theoretical analysis in [159], the third component λ is closely intertwined with the class discriminability of both the source and target domains. Consequently, in addressing the DAINO problem, three primary challenges emerge: 1) how to make full use of the inherent uncertainty information in interval-valued data to train a reliable source model (minimize the first

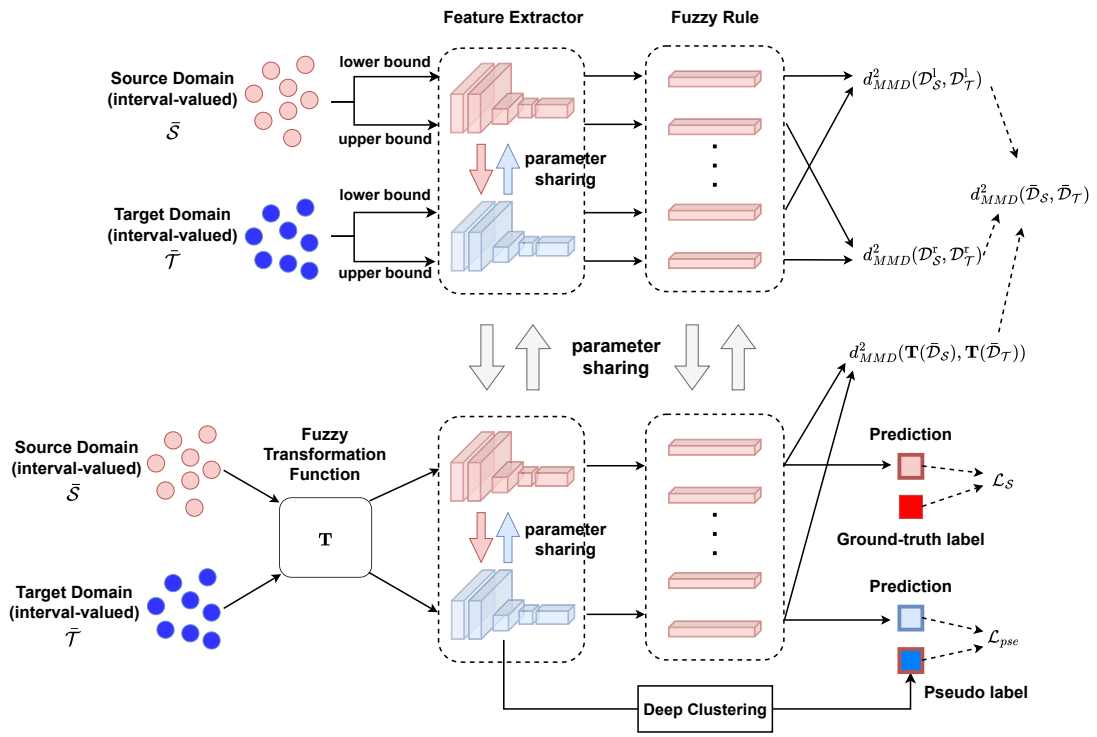


Figure 5.1: The procedure of SP-TSF. The Takagi-Sugeno fuzzy rule-based model serves as the fundamental model structure in our approach. Fuzzy transformation function **T** is employed to extract valuable crisp-valued information from interval-valued data. To attain distributional alignment between the interval-valued source and target domains, we introduce the concept of interval maximum mean discrepancy $d^2_{MMD}(\bar{\mathscr{D}}_{\mathscr{S}}, \bar{\mathscr{D}}_{\mathscr{T}})$. Finally, a deep clustering-based pseudo-labeling strategy is developed to acquire reliable pseudo labels for the target data, subsequently applying these pseudo-labeled target data to enhance the class discriminability within the interval-valued target domain.

term); 2) how to align the distribution between the interval-valued source and target domains (minimize the second term); 3) how to improve class discriminability of the interval-valued target domain (minimize the last term).

5.4 Model Construction

This section outlines our developed model SP-TSF, which is designed to address DAINO problems. The network framework of SP-TSF is shown in Fig. 5.1.

5.4.1 Takagi-Sugeno Fuzzy Rule-based Source Model Training

Drawing upon the theoretical analysis expounded in Section 5.3, our task necessitates the minimization of three key terms, as indicated in Eq. (5.10). To reduce the first term, the construction of a high-performance classifier within the source domain becomes imperative. In Chapter 4, we introduced an innovative framework boasting commendable classification accuracy, specifically tailored for resolving multi-class classification quandaries characterized by fuzzy-valued or interval-valued observations. This framework proposes one fuzzy transformation function, adept at extracting crisp-valued information from interval-valued data. Given its demonstrated proficiency in handling interval-valued data, we employ this function to process the interval-valued observations within both domains in this study. Here, $\mathbf{T}(\cdot;\beta)$ denotes the fuzzy transformation function,

$$\mathbf{T}(\bar{\mathbf{x}};\beta) = \mathrm{VAL}\circ\mathbf{F}(\bar{\mathbf{x}};\beta) = (\mathrm{VAL}\circ F(\bar{x}_1;\beta),\cdots,\mathrm{VAL}\circ F(\bar{x}_p;\beta))^\top, \tag{5.11}$$

where each $F(\bar{x}_j;\beta)$ ($j\in[p]$) is a triangular fuzzy number characterized by $\mathrm{Tr}(x_j^{\mathrm{l}}, \beta x_j^{\mathrm{l}} + (1-\beta)x_j^{\mathrm{r}}, x_j^{\mathrm{r}})$. Then, $X_s = \mathbf{T}(\bar{X}_s;\beta) \triangleq \{\mathbf{x}_i^s\}_{i=1}^{n_s} \sim \mathbf{T}(\bar{\mathscr{D}}_{\mathscr{S}})$ and $X_t = \mathbf{T}(\bar{X}_t;\beta) \triangleq \{\mathbf{x}_i^t\}_{i=1}^{n_t} \sim \mathbf{T}(\bar{\mathscr{D}}_{\mathscr{T}})$ denote as the extracted information from $\bar{X}_s$ and $\bar{X}_t$.

Next, we train the T-S fuzzy rule-based model on $\{\mathbf{x}_i^s, y_i^s\}_{i=1}^{n_s}$ to catch the intrinsic uncertainty information in interval-valued source domain. The trained rules are shown as following:

$$\begin{aligned}&\text{if } \mathbf{x}_i^s \ \ is \ \ A_l(\phi(\mathbf{x}_i^s)),\\ &\text{then } \mathbf{y}_i^s \ \ is \ \ P_l(\phi(\mathbf{x}_i^s)), l = 1,2,\cdots,L.\end{aligned}$$

ϕ is the shared feature extractor for both domains. Feature extractors transform original data to feature space $\mathbb{R}^d$. A_l represents the fuzzy condition of the l-th rule, P_l is a function transforming data from $\mathbb{R}^d$ to $\mathbb{R}^K$. L represents the number of rules. Then, the

final prediction of the T-S fuzzy model is the linear combining of the outputs of all rules:

$$\mathbf{y}_i^s = \sum_{l=1}^{L} \mu_s^l P_l(\phi(\mathbf{x}_i^s)), \tag{5.12}$$

where μ_s^l is the membership of $\mathbf{x}_i^s$ belonging to the l-th fuzzy set. To complete our source model, three problems need to be solved: 1) how to choose the fuzzy rule number L; 2) how to measure the membership μ_s^l; 3) how to optimize our source model.

To solve the first problem, we utilize the correlation coefficient between different pairs of classes as a criterion for grouping these classes. Subsequently, we determine the number of grouped classes as the count of fuzzy rules. In this context, the correlation coefficient matrix is denoted as:

$$\Sigma_\rho = \begin{pmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1K} \\ \rho_{21} & \rho_{22} & \cdots & \rho_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{K1} & \rho_{K2} & \cdots & \rho_{KK} \end{pmatrix}, \tag{5.13}$$

ρ_{ij} is the correlation coefficient between $\mathbf{x}_{k_i}$ and $\mathbf{x}_{k_j}$. where

$$\mathbf{x}_k = \sum_{i=1}^{n_k} \mathbf{1}_{\mathbf{y}_i^s=k}\ \phi(\mathbf{x}_i^s) / \sum_{i=1}^{n_k} \mathbf{1}_{\mathbf{y}_i^s=k}, \tag{5.14}$$

n_k denotes the number of source samples in the k-th class. When $\rho_{ij} > \rho$, where ρ is a threshold, we think classes k_i and k_j are similar, and they can share the same rule. After that, we group all classes in L different label sets $\mathbf{Y}_l, l \in [L]$, each $\mathbf{Y}_l$ contains similar classes, i.e., $\forall k_i, k_j \in \mathbf{Y}_l$, k_i, k_j share the same rule.

Next, FCM clustering is applied to address the second problem. In this work, cluster prototypes $\{v_s^l\}_{l=1}^L$ are initialized as the mean values of samples from the same grouped classes, expressed as:

$$v_s^l = \sum_{i=1}^{n_l} \mathbf{1}_{\mathbf{y}_i^s \in \mathbf{Y}_l}\ \phi(\mathbf{x}_i^s) / \sum_{i=1}^{n_l} \mathbf{1}_{\mathbf{y}_i^s \in \mathbf{Y}_l}, \tag{5.15}$$

n_l denotes the number of samples in $\mathbf{Y}_l$. The membership of $\mathbf{x}^s \in A_l$ is generally defined as:

$$\mu_s^l = 1/\sum_{l=1}^{L}(\frac{\|v_s^l - \phi(\mathbf{x}^s)\|}{\|v_s^i - \phi(\mathbf{x}^s)\|})^{\frac{2}{m-1}}. \tag{5.16}$$

Then, the cluster prototypes are updated with training processing using Eq. (5.16):

$$v_s^l = \sum_{i=1}^{n_s}(\mu_s^{li})^m \ \phi(\mathbf{x}_i^s)/\sum_{i=1}^{n_s}(\mu_s^{li})^m. \tag{5.17}$$

As for the last problem, we use the following loss function to optimize our source model:

$$\mathscr{L}_{\mathscr{S}} = \ell(P_l(\mathbf{x}^s), \mathbf{y}^s) + \ell(\sum_{l=1}^{L}\mu_s^l P_l(\phi(\mathbf{x}^s)), \mathbf{y}^s), \tag{5.18}$$

where ℓ denotes the adjusted cross-entropy loss function, incorporating the label smoothing technique introduced in [106] to enhance class discriminability. The first term aims to optimize P_l for each fuzzy rule, while the second term strives to minimize the discrepancy between the final prediction and the ground-truth source labels.

5.4.2 Interval Distribution alignment

Turning to the second term, $\widehat{\text{disc}}(S_u^{\text{l}}, T_u^{\text{l}})$ is the estimate distribution discrepancy between $\mathscr{D}_S^{\text{l}}$ and $\mathscr{D}_T^{\text{l}}$, and $\widehat{\text{disc}}(S_u^{\text{r}}, T_u^{\text{r}})$ is the estimate distribution discrepancy between $\mathscr{D}_S^{\text{r}}$ and $\mathscr{D}_T^{\text{r}}$. To minimize the second term, we need to reduce the distribution discrepancy between $X_s^{\text{l}}, X_t^{\text{l}}$ and $X_s^{\text{r}}, X_t^{\text{r}}$. *Maximum mean discrepancy* (MMD) is a widely used metrics to estimate distribution discrepancy between different distributions in traditional UDA works. Given two different distributions $\mathscr{D}_1$ and $\mathscr{D}_2$, the MMD between $\mathscr{D}_1$ and $\mathscr{D}_2$ is formulated as:

$$d_{MMD}^2(\mathscr{D}_1, \mathscr{D}_2) = \mathbb{E}_{\mathbf{x}_1, \mathbf{x}_1^{'} \sim \mathscr{D}_1}\mathbf{k}(\mathbf{x}_1, \mathbf{x}_1^{'}) + \mathbb{E}_{\mathbf{x}_2, \mathbf{x}_2^{'} \sim \mathscr{D}_2}\mathbf{k}(\mathbf{x}_2, \mathbf{x}_2^{'}) - 2\mathbb{E}_{\mathbf{x}_1 \sim \mathscr{D}_1, \mathbf{x}_2 \sim \mathscr{D}_2}\mathbf{k}(\mathbf{x}_1, \mathbf{x}_2), \tag{5.19}$$

Algorithm 4

Input: $\bar{\mathscr{S}} = (\bar{X}_s, Y_s) = \{(\bar{\mathbf{x}}_i^s, y_i^s)\}_{i=1}^{n_s}$, $X_s^{\mathrm{l}} = \{\mathbf{x}_i^{s,\mathrm{l}}\}_{i=1}^{n_s}$, $X_s^{\mathrm{r}} = \{\mathbf{x}_i^{s,\mathrm{r}}\}_{i=1}^{n_s}$, $\bar{\mathscr{T}} = \{\bar{\mathbf{x}}_i^t\}_{i=1}^{n_t}$, $X_t^{\mathrm{l}} = \{\mathbf{x}_i^{t,\mathrm{l}}\}_{i=1}^{n_t}$, $X_t^{\mathrm{r}} = \{\mathbf{x}_i^{t,\mathrm{r}}\}_{i=1}^{n_t}$, shape parameter β, learning rate LR, epoch T_{max}, and optimization algorithm (Adam algorithm [67] is selected);

Initial: feature extractor ϕ, the rule number L (see Eqs. (5.13)(5.14)), $\{P_l\}_{l=1}^{L}$;

Compute $\mathbf{X}_s = (\mathbf{T}(\bar{\mathscr{S}};\beta), Y_s) \triangleq \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}, X_t = \mathbf{T}(\bar{\mathscr{T}};\beta) \triangleq \{\mathbf{x}_i^t\}_{i=1}^{n_t}$;
for $T = 1, 2, \ldots, T_{max}$ **do**
    Calculate and update v_s^l, μ_s^l via Eqs. (5.15)(5.16)(5.17);
    Fetch mini-batches from $\mathbf{X}_s, X_s^{\mathrm{l}}, X_s^{\mathrm{r}}, X_t, X_t^{\mathrm{l}}, X_t^{\mathrm{r}}$;
    Train ϕ and $\{P_l\}_{l=1}^{L}$ with mini-batches from $\mathbf{X}_s$ to minimize $\mathscr{L}_{\mathscr{S}}$ (see Eq. (5.18));
    Train ϕ and $\{P_l\}_{l=1}^{L}$ with mini-batches from $\mathbf{X}_s, X_s^{\mathrm{l}}, X_s^{\mathrm{r}}, X_t, X_t^{\mathrm{l}}, X_t^{\mathrm{r}}$ to minimize $d_{MMD}^2(\bar{\mathscr{D}}_{\mathscr{S}}, \bar{\mathscr{D}}_{\mathscr{T}})$ (see Eq. (5.20));
    Obtain the pseudo labels of the target data via Eqs. (5.21)(5.22)(5.23);
    Calculate and update v_t^l, μ_t^l same as v_s^l, μ_s^l;
    Train ϕ and $\{P_l\}_{l=1}^{L}$ with mini-batches from X_t to minimize $\mathscr{L}_{pre}$ (see Eq. (5.24));
end for

Output: predicted target labels $\hat{y}_i^t = \sum_{l=1}^{L} \mu_t^l P_l(\phi(\mathbf{x}_i^t))$.

k is a kernel function. According to the second terms in Eq. (5.10), we design a new metrics called interval maximum mean discrepancy, an extension of the standard maximum mean discrepancy, to enhance distribution alignment on the interval-valued source and target domains. The interval maximum mean discrepancy between $\bar{\mathscr{D}}_{\mathscr{S}}$ and $\bar{\mathscr{D}}_{\mathscr{T}}$ is denoted as:

$$d_{MMD}^2(\bar{\mathscr{D}}_{\mathscr{S}}, \bar{\mathscr{D}}_{\mathscr{T}}) = d_{MMD}^2(\mathscr{D}_{\mathscr{S}}^{\mathrm{l}}, \mathscr{D}_{\mathscr{T}}^{\mathrm{l}}) + d_{MMD}^2(\mathscr{D}_{\mathscr{S}}^{\mathrm{r}}, \mathscr{D}_{\mathscr{T}}^{\mathrm{r}}) + d_{MMD}^2(\mathbf{T}(\bar{\mathscr{D}}_{\mathscr{S}}), \mathbf{T}(\bar{\mathscr{D}}_{\mathscr{T}})). \tag{5.20}$$

The first two terms are used to estimate the distribution discrepancy between $\mathscr{D}_S^{\mathrm{l}}, \mathscr{D}_T^{\mathrm{l}}$ and $\mathscr{D}_S^{\mathrm{r}}, \mathscr{D}_T^{\mathrm{r}}$, while $d_{MMD}^2(\mathbf{T}(\bar{\mathscr{D}}_{\mathscr{S}}), \mathbf{T}(\bar{\mathscr{D}}_{\mathscr{T}}))$ is aim to catch the additional uncertain distribution discrepancy of interval-valued data. Therefore, to achieve distribution alignment between the interval-valued source and target domains, our objective turn to minimize $d_{MMD}^2(\bar{\mathscr{D}}_{\mathscr{S}}, \bar{\mathscr{D}}_{\mathscr{T}})$ in the training process (see Fig. 5.1).

5.4.3 Enhance Class Discriminability of The Target Domain

As for minimizing the last term in Eq. (5.10), i.e., improving class discriminability of the interval-valued target domain, a self-supervised pseudo-labeling strategy based on deep clustering [16] is developed. First, we attain the centroid for each class in the target domain,

$$c_k^{(0)} = \frac{\sum_{i=1}^{n_t} \delta_k(\sum_{l=1}^{L} \mu_t^{l(0)} P_l(\phi(\mathbf{x}_i^t)))\ \phi(\mathbf{x}_i^t)}{\sum_{i=1}^{n_t} \delta_k(\sum_{l=1}^{L} \mu_t^{l(0)} P_l(\phi(\mathbf{x}_i^t)))}, \quad \mu_t^{l(0)} = \mu_s^l, \tag{5.21}$$

where δ_k denotes the k-th element in the soft-max output. Then, we obtain the pseudo labels via the nearest centroid classifier:

$$\widetilde{y}_t = \arg\min_k d_{cos}(\phi(\mathbf{x}_i^t), c_k^{(0)}), \tag{5.22}$$

where d_{cos} measures the cosine distance. Given the cosine distance's insensitivity to magnitude, computational efficiency, robustness to outliers, and high interpretability, we have opted for its use in measuring the distance between $\phi(\mathbf{x}_i^t)$ and the centroid. Then, we update the target centroids via the new pseudo labels and further update the pseudo labels via the updated target centroids:

$$c_k^{(1)} = \sum_{i=1}^{n_t} \mathbf{1}_{\widetilde{y}_t=k}\ \phi(\mathbf{x}_i^s) / \sum_{i=1}^{n_t} \mathbf{1}_{\widetilde{y}_t=k}, \quad \widetilde{y}_t = \arg\min_k d_{cos}(\phi(\mathbf{x}_i^t), c_k^{(1)}). \tag{5.23}$$

Subsequently, according to Eqs. (5.15)(5.16)(5.17) to obtain the updated membership μ_t^l of $\mathbf{x}^t$. Experiments verify that updating for once gives sufficiently good pseudo labels. Consequently, the obtained pseudo labels are employed to enhance target domain's class discriminability by minimizing the following loss function:

$$\mathscr{L}_{pre} = \ell(P_l(\mathbf{x}^t), \widetilde{y}_t) + \ell(\sum_{l=1}^{L} \mu_t^l P_l(\phi(\mathbf{x}^t)), \widetilde{y}_t). \tag{5.24}$$

Follow by Eqs. (5.18)(5.20)(5.24), the overall training objective of our proposed model is formulated as:

$$\mathscr{L}_{total} = \mathscr{L}_{\mathscr{S}} + \lambda_1 d_{MMD}^2(\bar{\mathscr{D}}_{\mathscr{S}}, \bar{\mathscr{D}}_{\mathscr{T}}) + \lambda_2 \mathscr{L}_{pre}, \tag{5.25}$$

where λ_1 and λ_2 are two trade-off parameters. More details of SP-TSF are provided in Algorithm 4.

5.5 Experiment

In this section, we substantiate the effectiveness of the proposed algorithm in addressing DAINO problems. To achieve this, we conduct a comparative assessment by benchmarking our approach against several baseline methods. We evaluate these approaches based on classification accuracy, utilizing both synthetic datasets and real-world domain adaptation tasks as the testing grounds.

5.5.1 Baselines

The baselines constructed for comparison with the proposed algorithm are presented in this section. Since no existing UDA algorithm can be used to directly address a DAINO problem, we applied two methods to transfer the interval-valued data into the crisp-valued data. **Midpoint:** The first method is take the midpoint of the interval-valued features. **Two Side:** The second method treats the upper and lower bounds of the interval-valued features as two crisp-valued features. Then, we executed several state-of-the-art UDA algorithms on these crisp-valued datasets: **DAN**[93], **DANN**[48], **CDAN**[94], **ATM**[77], **FixBi**[107], **DWL**[159], **DALN**[19], **CAF-A**[160], **SRDA**[14], and **AGE-CS**[142]. The term “Source only" refers to the use of the complete model trained on the source domain for target label prediction.

5.5.2 Experimental Setup

The settings for SP-TSF were set $T_{max} = 100$ with a learning rate of LR = 0.001. The setting for the trade-off parameters $\lambda_1 = 0.1, \lambda_2 = 1$, and the mini-batch size was set

to 200 for all methods. The feature extractor for all methods was a two-layer network with ReLU and Dropout in all the layers. There were 100 hidden layer units in all the layers. For each $l \in [L]$, P_l is a two-layer classifier ($100 \times$ #K). For all other baselines, the structure of the classifier is the same as P_l. We used Adam [67] as the optimization algorithm for all methods with a momentum of 0.9 and a weight decay of 0.0001. Further, for shape parameter β, we choose it that achieve the best performance on the source domain as the optimal β in this paper. The value of shape parameter β is selected from $\{0, 0.05, 0.1, \cdots, 0.95, 1\}$. The entire experimental process was repeated 5 times for all methods. Thus, the final results are reported in the form of "mean± standard deviation". We implemented the model with PyTorch 1.9.0. All experiments were conducted on an NVIDIA Quadro GV100 GPU with 32 GB memory.

5.5.3 Experiments on Synthetic Datasets

Data Generation: To create our synthetic datasets, we employed a data generation mechanism designed for producing interval-valued datasets tailored for domain adaptation scenarios. Initially, we generated a crisp-valued dataset with K categories using a random number generator, denoted as $\{(\mathbf{x}_i = (x_{i1}, x_{i2})^\top, y_i)\}_{i=1}^n$. Subsequently, we harnessed the generated crisp-valued dataset to formulate the interval-valued dataset $\{\bar{\mathbf{x}}_i = (\bar{x}_{i1}, \bar{x}_{i2})^\top, y_i\}_{i=1}^n$, where each $\bar{x}_{ij}$ represents an interval characterized by $[x_{ij} - a_{ij}, x_{ij} + b_{ij}]$. Here, a_{ij}, b_{ij} conform to uniform distributions.

Next, to construct the first synthetic dataset, we used a double moon data generator with $K = 2$ as a random number generator to generate two datasets with $n = 2000$. We let the added noise follow $\mathcal{N}(0, 0.05^2)$ and $\mathcal{N}(0, 0.2^2)$ for each of the two datasets. Additionally, we augmented the feature count for the two datasets by 10 and 8, correspondingly. Finally, the above mentioned method was used to convert the generated crisp-valued datasets into two interval-valued datasets - one for the source domain and

Table 5.1: Accuracy (mean ± std %) on the two synthetic datasets. The bold value represents the highest accuracy in each column.

Standards	Methods	1st	2nd
Midpoint	Source Only	80.45 ± 0.55	77.47 ± 1.00
	DAN[93]	80.49 ± 0.27	70.12 ± 0.68
	DANN[48]	80.71 ± 0.58	64.77 ± 0.19
	CDAN[94]	80.86 ± 0.46	66.07 ± 4.14
	ATM[77]	79.63 ± 1.13	64.62 ± 0.26
	FixBi[107]	80.92 ± 1.30	71.33 ± 4.53
	DWL[159]	81.77 ± 1.36	68.40 ± 0.20
	DALN[19]	80.88 ± 0.51	63.95 ± 1.32
	CAF-A[160]	78.91 ± 0.68	63.49 ± 0.42
	SRDA[14]	79.26 ± 0.31	63.33 ± 2.48
	AGE-CS[142]	79.96 ± 0.89	65.64 ± 1.89
Two Side	Source Only	85.56 ± 0.87	77.31 ± 0.67
	DAN[93]	81.00 ± 0.67	68.58 ± 0.43
	DANN[48]	79.92 ± 0.76	63.86 ± 0.83
	CDAN[94]	80.62 ± 0.39	64.49 ± 0.20
	ATM[77]	80.97 ± 0.79	65.46 ± 3.41
	FixBi[107]	80.87 ± 0.86	71.42 ± 2.75
	DWL[159]	84.87 ± 1.33	70.00 ± 4.96
	DALN[19]	81.94 ± 0.81	64.27 ± 0.60
	CAF-A[160]	80.17 ± 0.98	63.71 ± 0.60
	SRDA[14]	80.44 ± 0.46	67.87 ± 3.41
	AGE-CS[142]	82.45 ± 0.39	67.05 ± 2.45
	SP-TSF w/o pse	86.14 ± 0.55	79.27 ± 0.19
	SP-TSF w/o IMMD	85.89 ± 0.32	79.93 ± 0.37
	SP-TSF	**86.23 ± 0.35**	**80.44 ± 1.64**

the other for the target domain, where $a_{ij} \sim U[0.5,1], b_{ij} \sim U[1,2]$ for the source domain and $a_{ij} \sim U[1,1.5], b_{ij} \sim U[2,5]$ for the target domain.

To build the second synthetic dataset, we used a Gaussian data generator with $K = 3$ as a random number generator. We set the standard deviations of the clusters to $[0.9,2,4]$ and $[1.5,2,5]$, and the data ranges to $(0,20)$ and $(-25,5)$ for the source and target data, respectively. Then, the same intervalization method was used to generate the interval-valued source and target data, where $a_{ij} \sim U[0.5,1], b_{ij} \sim U[2,5]$ for both

domains.

Experimental Results: The experimental results for the two synthetic datasets are shown in Table 5.1. As demonstrated by the results, our method exhibited markedly superior performance, achieving classification accuracies of 86.23% and 80.44% for the respective datasets, surpassing all baseline methods. This underscores the efficacy of our approach in addressing DAINO problems. Notably, even the Source Only method utilizing a T-S fuzzy rule-based model outperformed all non-fuzzy baselines, which clearly shows the robust performance of fuzzy techniques when handling interval-valued data.

5.5.4 Experiments on Real-World Datasets

Dataset Description: We employed the Weather dataset as our real-world dataset, comprising meteorological data from three American cities [1] (from January 1, 2016 to December 31, 2021). These cities are Seattle Tacoma (**S**), Olympia (**O**), and Washington (**W**). Each instance within this dataset represents meteorological data for a single day in one of these American cities. The dataset encompasses five interval-valued variables (air temperature T, atmospheric pressure at weather station level P_0, atmospheric pressure reduced to main sea level P, humidity U, and dew-point temperature T_d), alongside a categorical variable indicating precipitation (0 ≡ No Precipitation, 1 ≡ Precipitation). For our evaluation, we formulated six domain adaptation tasks : $\mathbf{S} \rightarrow \mathbf{O}, \mathbf{O} \rightarrow \mathbf{S}, \mathbf{S} \rightarrow \mathbf{W}, \mathbf{W} \rightarrow \mathbf{S}, \mathbf{O} \rightarrow \mathbf{W}, \mathbf{W} \rightarrow \mathbf{O}$. The objective was to compare the performance of our method against baseline approaches across these six domain adaptation tasks.

[1] Extract from : https://rp5.ru/

Table 5.2: Accuracy (mean ± std %) on the real-world dataset for unsupervised domain adaptation. The bold value represents the highest accuracy in each column.

Standards	Methods	**S → O**	**O → S**	**S → W**	**W → S**	**O → W**	**W → O**	Average
Midpoint	Source Only	74.20 ± 0.38	74.50 ± 0.62	71.94 ± 0.40	73.35 ± 0.33	71.14 ± 1.13	72.50 ± 0.61	72.94
	DAN[93]	80.26 ± 2.47	77.45 ± 0.91	68.94 ± 2.97	77.01 ± 1.26	59.64 ± 1.54	78.46 ± 2.58	73.63
	DANN[48]	80.39 ± 0.69	78.31 ± 0.56	70.36 ± 1.73	74.80 ± 0.93	69.34 ± 1.56	75.14 ± 1.49	74.72
	CDAN[94]	79.30 ± 0.54	80.15 ± 0.54	69.36 ± 0.73	77.64 ± 0.23	66.47 ± 0.97	79.77 ± 0.78	75.25
	ATM[77]	81.27 ± 0.54	82.03 ± 0.32	69.49 ± 1.90	80.86 ± 0.19	69.79 ± 1.01	80.11 ± 0.16	77.26
	FixBi[107]	80.05 ± 1.39	77.06 ± 1.78	72.66 ± 0.63	76.85 ± 2.48	69.88 ± 1.79	78.28 ± 2.05	75.80
	DWL[159]	83.89 ± 0.82	80.21 ± 1.04	74.82 ± 1.10	73.20 ± 2.17	73.41 ± 1.39	80.50 ± 2.67	77.67
	DALN[19]	81.47 ± 0.20	82.04 ± 0.11	69.04 ± 1.09	81.06 ± 0.05	69.12 ± 1.17	80.19 ± 0.24	77.15
	CAF-A[160]	80.49 ± 0.33	79.54 ± 0.13	70.81 ± 1.30	80.07 ± 1.05	70.28 ± 1.35	80.15 ± 0.51	76.89
	SRDA[14]	81.08 ± 0.42	80.82 ± 0.26	72.15 ± 0.60	80.41 ± 0.23	69.60 ± 1.16	79.27 ± 0.14	77.22
	AGE-CS[142]	81.56 ± 0.75	81.82 ± 0.41	71.75 ± 0.85	80.56 ± 0.23	69.77 ± 1.45	80.49 ± 0.26	77.66
Two Side	Source Only	78.25 ± 0.36	77.21 ± 0.49	75.30 ± 0.42	78.37 ± 0.96	70.71 ± 1.12	75.81 ± 0.79	75.94
	DAN[93]	83.03 ± 0.27	81.90 ± 0.38	68.94 ± 1.02	81.59 ± 0.54	60.20 ± 0.43	79.63 ± 1.67	75.88
	DANN[48]	81.46 ± 0.12	79.92 ± 0.20	73.31 ± 0.39	81.01 ± 0.21	69.98 ± 0.93	80.41 ± 0.24	77.68
	CDAN[94]	81.62 ± 0.39	79.96 ± 0.37	74.24 ± 1.20	81.04 ± 0.85	71.16 ± 0.92	80.75 ± 0.35	78.13
	ATM[77]	**84.25 ± 0.67**	81.52 ± 1.93	70.98 ± 2.17	**83.43 ± 0.48**	70.58 ± 0.19	81.74 ± 0.54	78.75
	FixBi[107]	82.55 ± 0.39	78.06 ± 1.28	73.66 ± 0.56	79.09 ± 2.12	72.90 ± 1.34	80.56 ± 1.52	77.80
	DWL[159]	83.80 ± 0.76	79.80 ± 1.04	74.70 ± 1.44	74.68 ± 1.70	73.80 ± 1.20	81.30 ± 2.20	78.01
	DALN[19]	83.51 ± 1.46	**82.65 ± 0.67**	73.88 ± 0.97	82.50 ± 0.58	70.93 ± 0.85	80.19 ± 0.24	78.94
	CAF-A[160]	81.74 ± 0.70	82.56 ± 1.25	72.07 ± 1.61	81.56 ± 0.44	71.42 ± 1.78	79.67 ± 0.71	78.17
	SRDA[14]	82.96 ± 0.57	81.67 ± 0.39	74.31 ± 0.35	81.82 ± 0.68	72.49 ± 0.57	80.22 ± 1.15	78.91
	AGE-CS[142]	83.98 ± 0.78	81.16 ± 0.46	74.25 ± 0.42	82.05 ± 0.47	71.45 ± 1.08	80.98 ± 0.27	78.97
	SP-TSF w/o pse	82.36 ± 0.32	79.97 ± 0.33	74.60 ± 0.06	75.35 ± 0.23	73.16 ± 0.53	74.25 ± 0.14	76.62
	SP-TSF w/o IMMD	83.29 ± 0.33	80.90 ± 0.14	75.33 ± 0.12	79.51 ± 0.71	73.96 ± 0.44	79.87 ± 0.52	78.81
	SP-TSF	83.65 ± 0.17	82.47 ± 0.17	**75.61 ± 0.19**	81.20 ± 0.35	**74.19 ± 0.31**	**81.93 ± 0.24**	**79.84**

Experimental results: The experimental results for the six real-world tasks are presented in Table 5.2. Analyzing these results reveals that our algorithm demonstrated the highest performance in three tasks: $\mathbf{S} \rightarrow \mathbf{W}, \mathbf{O} \rightarrow \mathbf{W}, \mathbf{W} \rightarrow \mathbf{O}$, and it also achieved the best overall average performance. These findings once again underscore the superior effectiveness of our algorithm in addressing DAINO problems.

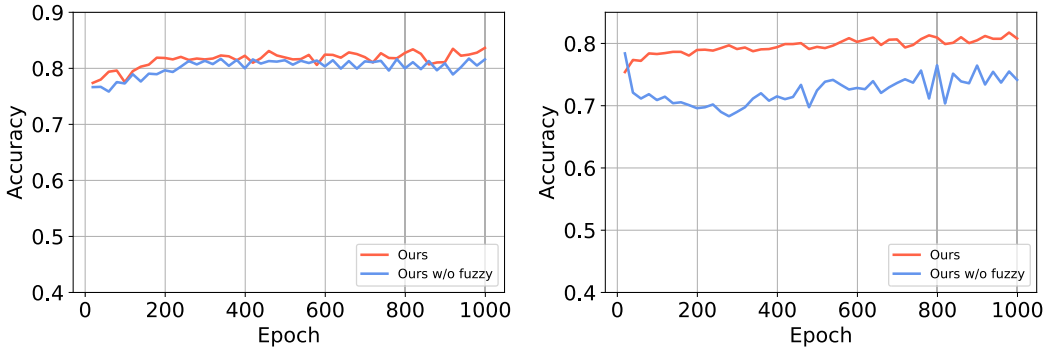
5.5.5 Influence of Fuzzy Techniques

To assess the effectiveness of the employed fuzzy techniques in handling interval-valued data, we systematically eliminate all fuzzy techniques used in our model, as well as the Source Only method. Initially, we replace the fuzzy transformation function with the **Midpoint** method. Since we need to minimize $d^2_{MMD}(\mathscr{D}^{\mathrm{l}}_{\mathscr{S}}, \mathscr{D}^{\mathrm{l}}_{\mathscr{T}}) + d^2_{MMD}(\mathscr{D}^{\mathrm{r}}_{\mathscr{S}}, \mathscr{D}^{\mathrm{r}}_{\mathscr{T}})$, **Two Side** method can not be used. As for interval maximum mean discrepancy, we remove $d^2_{MMD}(\mathbf{T}(\bar{\mathscr{D}}_{\mathscr{S}}), \mathbf{T}(\bar{\mathscr{D}}_{\mathscr{T}}))$ in Eq. (5.20). Finally, we substitute the T-S fuzzy rule-based model with a commonly used neural network-based model framework. The comparative experimental results are presented in Table 5.3. It is evident that the model's performance experiences a significant decline following the removal of fuzzy techniques. This observation demonstrates the capability of the applied fuzzy techniques in effectively capturing the inherent uncertainty within interval-valued data.

Furthermore, in Fig. 5.2, we depict the evolution of classification accuracy on the target domain as a function of the number of epochs for the tasks $\mathbf{S} \rightarrow \mathbf{O}$ and $\mathbf{W} \rightarrow \mathbf{O}$. It is noteworthy that in Figs. 5.2(a) and 5.2(b), the performance of SP-TSF w/o fuzzy exhibits substantial oscillations and a declining trend with increasing epochs, in stark contrast to the stability and improvement observed in our proposed method. These collective findings not only underscore the performance enhancement afforded by fuzzy techniques but also highlight their role in bolstering the robustness of our model.

Table 5.3: Accuracy (mean %) for analyzing the influence of fuzzy techniques.

Standards	Methods	1st	2st	**S → O**	**O → S**	**S → W**	**W → S**	**O → W**	**W → O**	Average
Midpoint	Source Only w/o fuzzy	80.13	64.34	**76.38**	**76.64**	63.70	67.47	60.22	72.46	70.17
	Source Only	**80.45**	**77.47**	74.20	74.50	**71.94**	**73.35**	**71.14**	**72.50**	**74.44**
Two Side	Source Only w/o fuzzy	77.59	62.42	**81.52**	**80.27**	69.02	78.25	64.77	75.68	73.69
	Source Only	**85.56**	**77.31**	78.25	77.21	**75.30**	**78.37**	70.71	**75.81**	**77.32**
	SP-TSF w/o fuzzy	80.47	64.01	81.95	81.88	74.06	80.60	73.38	80.14	77.06
	SP-TSF	**86.23**	**80.44**	**83.65**	**82.47**	**75.61**	**81.20**	**74.19**	**81.93**	**80.72**



(a) **S → O** (b) **W → O**

Figure 5.2: Classification accuracy on the target domain varies with the number of epochs.

5.5.6 Ablation Study

To validate the soundness of our model's construction, we conducted an ablation study. The variants utilized in this study are denoted as follows: SP-TSF w/o pse that removes $\mathscr{L}_{pre}$ in Eq. (5.25); SP-TSF w/o IMMD that removes the distribution alignment via minimizing interval maximum mean discrepancy, i.e., delete $d^2_{MMD}(\bar{\mathscr{D}}_{\mathscr{S}}, \bar{\mathscr{D}}_{\mathscr{T}})$ in Eq. (5.25). The experimental results of this ablation study on both synthetic and real-world datasets are presented in Tables 5.1 and 5.2. A comparison between SP-TSF w/o pse and the proposed full model exposes the efficacy of the self-supervised pseudo-labeling strategy in obtaining reliable pseudo-labels for the target data. Similarly, a comparison between SP-TSF w/o IMMD and SP-TSF underscores the advantageous properties of the designed interval maximum mean discrepancy in achieving distribution alignment between the interval-valued source and target domains.

5.6 Summary

In this chapter, we identify a challenging real-world problem called DAINO, which involves improving the classification accuracy of an unlabeled target domain by leveraging knowledge from a source domain with sufficient labeled data where both domains only contain interval-valued observations.

Given the absence of existing literature on the DAINO problem, we derive an upper bound on the risk within a target domain. These bounds elucidate three principal elements that necessitate consideration to minimize risk in the target domain featuring interval-valued observations. Drawing upon the theoretical analysis presented in Section 5.3, we develop a novel theoretically-guided model employing T-S fuzzy rules and a self-supervised pseudo-labeling strategy to tackle DAINO problems. Extensive experimentation on both synthetic and real-world datasets not only validates the soundness of

our theoretical analysis but also demonstrates the superior performance of our algorithm compared to several competitive baselines.

5.7 Appendix

5.7.1 Proof of Lemma 5.1

According to Definitions 4.6 and 5.2, we have

$$\begin{aligned}
\operatorname{disc}(\bar{\mathscr{D}}_S, \bar{\mathscr{D}}_T) &= \sup_{(\boldsymbol{h},\boldsymbol{h}')\in\mathscr{H}\times\mathscr{H}} |\mathbb{E}_{\bar{\mathscr{D}}_S}[\ell(\boldsymbol{h},\boldsymbol{h}')] - \mathbb{E}_{\bar{\mathscr{D}}_T}[\ell(\boldsymbol{h},\boldsymbol{h}')]| \\
&= \frac{1}{2} \sup_{(\boldsymbol{h},\boldsymbol{h}')\in\mathscr{H}\times\mathscr{H}} |\mathbb{E}_{\mathscr{D}_S^{\mathrm{l}}}[\ell(\boldsymbol{h},\boldsymbol{h}')] + \mathbb{E}_{\mathscr{D}_S^{\mathrm{r}}}[\ell(\boldsymbol{h},\boldsymbol{h}')] - \mathbb{E}_{\mathscr{D}_T^{\mathrm{l}}}[\ell(\boldsymbol{h},\boldsymbol{h}')]| - \mathbb{E}_{\mathscr{D}_T^{\mathrm{r}}}[\ell(\boldsymbol{h},\boldsymbol{h}')]| \\
&\leq \frac{1}{2} \sup_{(\boldsymbol{h},\boldsymbol{h}')\in\mathscr{H}\times\mathscr{H}} |\mathbb{E}_{\mathscr{D}_S^{\mathrm{l}}}[\ell(\boldsymbol{h},\boldsymbol{h}')] - \mathbb{E}_{\mathscr{D}_T^{\mathrm{l}}}[\ell(\boldsymbol{h},\boldsymbol{h}')]| + \frac{1}{2} \sup_{(\boldsymbol{h},\boldsymbol{h}')\in\mathscr{H}\times\mathscr{H}} |\mathbb{E}_{\mathscr{D}_S^{\mathrm{r}}}[\ell(\boldsymbol{h},\boldsymbol{h}')] - \mathbb{E}_{\mathscr{D}_T^{\mathrm{r}}}[\ell(\boldsymbol{h},\boldsymbol{h}')]| \\
&= \frac{1}{2}\operatorname{disc}(\mathscr{D}_S^{\mathrm{l}}, \mathscr{D}_T^{\mathrm{l}}) + \frac{1}{2}\operatorname{disc}(\mathscr{D}_S^{\mathrm{r}}, \mathscr{D}_T^{\mathrm{r}}).
\end{aligned}$$

5.7.2 Proof of Lemma 5.2

By Theorem 4.1 and using the definition of disc, for any $\delta > 0$, with probability at least $1-\delta$, the following inequality holds for any $\boldsymbol{h}, \boldsymbol{h}' \in \mathscr{H}$:

$$\begin{aligned}
\widehat{\operatorname{disc}}(\bar{\mathscr{D}}, \bar{S}_{\bar{X}}) &= |R_{\bar{\mathscr{D}}}(\boldsymbol{h},\boldsymbol{h}') - \widehat{R}_{\mathscr{D}}(\boldsymbol{h},\boldsymbol{h}')| \\
&\leq 2\sqrt{2}L_\ell \mathscr{R}_{\bar{S}_{\bar{X}}}(\mathscr{H}) + C_\ell \sqrt{\frac{\log(1/\delta)}{2m}} \\
&\leq 2\sqrt{2}L_\ell \widehat{\mathscr{R}}_{\bar{S}_{\bar{X}}}(\mathscr{H}) + 3C_\ell \sqrt{\frac{\log(2/\delta)}{2m}}.
\end{aligned}$$

5.7.3 Proof of Corollary 5.1

According to Eq. (5.8), for any $\delta > 0$, with probability at least $1-\delta$ over the choice of sample $\bar{S}_{\bar{X}}$, we have

$$\widehat{\mathrm{disc}}(\bar{\mathscr{D}}, \bar{S}_{\bar{X}}) \leq 2\sqrt{2} L_\ell \widehat{\mathscr{R}}_{\bar{S}_{\bar{X}}}(\mathscr{H}) + 3C_\ell \sqrt{\frac{\log(2/\delta)}{2m}}. \tag{5.26}$$

Then, the final results are obtained by using the triangle inequality twice

$$\mathrm{disc}(\bar{\mathscr{D}}_S, \bar{\mathscr{D}}_T) \leq \widehat{\mathrm{disc}}(S_u, \bar{\mathscr{D}}_S) + \widehat{\mathrm{disc}}(\widehat{\mathscr{D}}_T, T_u) + \widehat{\mathrm{disc}}(S_u, T_u),$$

and by applying the Eq. (5.26) to $\widehat{\mathrm{disc}}(S_u, \bar{\mathscr{D}}_S)$ and $\widehat{\mathrm{disc}}(\widehat{\mathscr{D}}_T, T_u)$.

5.7.4 Proof of Theorem 5.1

For any $\boldsymbol{h} \in \mathscr{H}$. Applying the triangle inequality to ℓ and incorporating Eqs. (4.2)(5.6)(5.9) gives the following result

$$\begin{aligned}
R_{\bar{\mathscr{D}}_T}(\boldsymbol{h}) &\leq R_{\bar{\mathscr{D}}_T}(\boldsymbol{h}, \boldsymbol{h}^*) + R_{\bar{\mathscr{D}}_T}(\boldsymbol{h}^*, f_T) \leq R_{\bar{\mathscr{D}}_T}(\boldsymbol{h}^*) + \mathrm{disc}(\bar{\mathscr{D}}_S, \bar{\mathscr{D}}_T) + R_{\bar{\mathscr{D}}_S}(\boldsymbol{h}, \boldsymbol{h}^*) \\
&\leq R_{\bar{\mathscr{D}}_T}(\boldsymbol{h}^*) + \mathrm{disc}(\bar{\mathscr{D}}_S, \bar{\mathscr{D}}_T) + R_{\bar{\mathscr{D}}_S}(\boldsymbol{h}) + R_{\bar{\mathscr{D}}_S}(\boldsymbol{h}^*) = R_{\bar{\mathscr{D}}_S}(\boldsymbol{h}) + \mathrm{disc}(\bar{\mathscr{D}}_S, \bar{\mathscr{D}}_T) + \lambda \\
&\leq \widehat{R}_{\bar{\mathscr{D}}}(\boldsymbol{h}) + \frac{1}{2}\widehat{\mathrm{disc}}(S_u^{\mathrm{l}}, T_u^{\mathrm{l}}) + \frac{1}{2}\widehat{\mathrm{disc}}(S_u^{\mathrm{r}}, T_u^{\mathrm{r}}) + 2\sqrt{2} L_\ell (2\widehat{\mathscr{R}}_{S_u}(\mathscr{H}) + \widehat{\mathscr{R}}_{T_u}(\mathscr{H})) \\
&\quad + 3C_\ell \left(2\sqrt{\frac{\log(6/\delta)}{2m_s}} + \sqrt{\frac{\log(6/\delta)}{2m_t}}\right) + \lambda.
\end{aligned}$$

CHAPTER 6

MULTI-SOURCE DOMAIN ADAPTATION WITH INTERVAL-VALUED TARGET DATA

6.1 Introduction

Remarkable achievements have been made in both theoretical and applied aspects of domain adaptation [93]. The most successful application areas include computer vision [44], biology [164], and natural language processing [123]. Recently, researchers have focused on more realistic and challenging domain adaptation problems, such as MSDA [117], and SFDA [83]. Specifically, MSDA [79] utilizes knowledge from multiple source domains to enhance prediction performance on the target domain. SFDA [37] involves source data that is unavailable during the adaptation process.

Most existing MSDA works [117] operate under the common assumption that the target domain only contain crisp-valued data [65]. However, in some real-world scenarios, we may encounter a situation where fully labeled data from multiple sources with crisp-valued features is available, but the target data is unlabeled and charac-

terized by interval-valued features [8, 38]. For example, a company operates multiple manufacturing facilities that produce similar but slightly different equipment. Each facility generates fully labeled data with crisp-valued features related to the performance and operational conditions of their machines. However, the company also has a central database that collects data from various sensors installed across all facilities, creating an unlabeled dataset characterized by interval-valued features. These sensors might capture measurements such as temperature ranges, vibration levels, or pressure intervals, which can vary due to different setups, environmental conditions, or machinery versions across facilities. The objective here is to develop a predictive maintenance model for the central database (unlabeled target data) by leveraging the knowledge and labeled data from multiple manufacturing facilities (fully labeled multiple source data).

Hence, in this chapter, we focus on a more realistic and challenging problem known as MSDA with interval-valued target data, aimed at addressing the aforementioned specific scenarios. This discrepancy in data representation poses a significant obstacle to effectively utilizing the wealth of information present in multiple source domains to improve the performance of models on the unlabeled target data. Traditional MSDA techniques [117] struggle to handle this unique scenario where the target data features are represented as intervals, limiting their applicability and performance.

To effectively tackle the proposed problem, two main challenges need to be resolved. The first challenge is how to handle the interval-valued target data. The second challenge is how to fully utilize the previously acquired knowledge obtained from multiple source domains. On one hand, interval-valued features contain a significant level of uncertainty compared to typical crisp-valued features. On the other hand, the correlation between each crisp-valued source domain and the interval-valued target domain cannot be clearly measured by traditional methods due to the presence of uncertainty. Therefore, in addressing these two main challenges, we need to account for the impact of these

uncertain problems.

Fuzzy techniques, such as fuzzy rule-based systems [140], fuzzy clustering [75], and fuzzy relations [25], were specifically designed to deal with uncertainty and imprecision. Most non-fuzzy methods [79, 176] typically assume crisp-valued data and deterministic relationships, which may not accurately capture the inherent uncertainty and imprecision present in real-world scenarios. Moreover, they often find it difficult to provide robust and reliable predictions or decisions when confronted with uncertain data. Recently, researchers have integrated fuzzy techniques [75, 140] into machine learning algorithms to solve various uncertain problems, such as handling outliers [55] and analyzing imprecise or noisy data [71]. These advancements have demonstrated the advantages of incorporating fuzzy techniques in handling uncertain problems. Motivated by the benefits of fuzzy techniques, we propose two fuzzy technique-based frameworks to effectively address the two main challenges posed by the proposed problem.

The first framework, called *fuzzy multi-adversarial training neural networks* (FUMAT-Net), contains four main components: i) fuzzy transformation function, ii) feature extractor, iii) adversarial training, and iv) classifiers. This fuzzy transformation function, that proposed in Chapter 4, is used to extract crisp-valued infromation from interval-valued features. Second, a feature extractor is designed to extract common representations from the multiple source domains and crisp-valued information of the target domain. Then, we apply adversarial training [48] to align the distribution between the multiple source domains and the target domain. Finally, we train the multiple classifiers to simultaneously minimize the misclassification loss on the multiple source domains and prediction discrepancy of the data in the target domain. In addition, we propose a new fuzzy relation to measure the correlation between the multiple source domains and the target domain. Further, this fuzzy relation is applied to select the optimized shape parameter β of the fuzzy transformation function and to derive the weight vector for the final prediction of

the target samples, which can improve the performance of FUMAT-Net.

The second framework, called *fuzzy distance-based information maximization neural networks* (FDIM-Net), consists of two main components. The first component utilizes the same fuzzy transformation function to extract valuable crisp-valued information from the interval-valued target data. Additionally, we prove a theorem (Theorem 6.2) that provides guidance on how to appropriately combine multiple outputs for the final prediction and the multiple losses trained on the multiple classifiers. Building upon this analysis, we propose four different types of fuzzy distances and utilize these distances to develop a novel method for estimating the distribution discrepancy between each crisp-valued source domain and the interval-valued target domain. The estimated distribution discrepancy is then used to calculate a weight vector for the appropriate combination.

Utilizing the calculated weight vector, we construct the second framework. Numerous SFDA works [80, 83] have shown improved adaptation performance on the target domain. In addition, without access source data, the adapted model will not face the negative impact caused by some source data that has a significantly distribution discrepancy with the target data. Hence, FDIM-Net is devised as an SFDA model. Firstly, in training phase, source private model is trained on the multiple source domains that contain a share feature extractor and multiple classifiers. Since the source data is not available during the adaptation process, our model mitigates domain gap by minimizing the information maximization loss [83] on the extracted target crisp-valued information. Subsequently, an additional loss [117] is introduced to minimize the discrepancy among all classifiers, thereby bolstering prediction reliability in the target domain. Ultimately, during the testing phase, predictions for the target data are obtained through a weighted combination of outputs from multiple classifiers.

In our experiments, we verify the superiority of the proposed frameworks by comparing them with several non-fuzzy baselines on both synthetic and real-world datasets.

These comparisons highlight the exceptional ability of fuzzy techniques in handling problems of uncertainty. Finally, the outcomes of the ablation study and parameter sensitivity analysis demonstrate the rationality of the proposed fuzzy techniques-based method.

The main contributions of this chapter are as follows.

1. It identifies a more realistic and challenging problem known as MSDA with interval-valued target data. In many real-world scenarios, the collection of imprecise data, such as interval-valued data, is inevitable. Thus, it becomes necessary to find an effective approach for analyzing these types of data. In our identified problem, we aim to enhance the prediction performance on interval-valued target data by leveraging the knowledge derived from multiple source data with crisp-valued features.

2. To address this identified problem, we develop two frameworks based on fuzzy techniques. The first is based on fuzzy relation and the second is based on fuzzy distance.

3. Experiments conducted on both synthetic and real-world datasets validate the superior performance of our proposed MSDA models compared to several state-of-the-art non-fuzzy methods in addressing the proposed problem.

6.2 Problem Setting

In this section, we introduce the problem of MSDA with interval-valued target data. Let $\mathscr{X} \subset \mathbb{R}^p$ be the input space with crisp-valued observations and $\mathscr{D}$ as the distribution of

the input space $\mathscr{X}$. Next, the ground truth function of $\mathscr{X}$ is defined as:

$$\begin{aligned} &\mathbf{f}: \mathscr{X} \to \mathbb{R}^K \\ &(x_1, \cdots, x_p)^\top \to (f_1(\mathbf{x}), \cdots, f_K(\mathbf{x}))^\top. \end{aligned} \tag{6.1}$$

If $\mathbf{x} \in \mathscr{X}$ belongs to the C-th class, then $f_C(\mathbf{x}) = 1$ and for any $k \in [K], k \neq C, f_k(\mathbf{x}) = 0$.

Let $\mathscr{H}$ be the hypothesis set and for any $\boldsymbol{h} \in \mathscr{H}$,

$$\begin{aligned} &\boldsymbol{h}: \mathscr{X} \to \mathbb{R}^K \\ &(x_1, \cdots, x_p)^\top \to (h_1(\mathbf{x}), \cdots, h_K(\mathbf{x}))^\top. \end{aligned} \tag{6.2}$$

The loss function of $\boldsymbol{h}$ is defined as,

$$\ell: \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}_+. \tag{6.3}$$

Let $\mathscr{L}_{\mathscr{H}} = \{\ell(\boldsymbol{h}'(\mathbf{x}), \boldsymbol{h}(\mathbf{x})) : \boldsymbol{h}, \boldsymbol{h}' \in \mathscr{H}, \mathbf{x} \in \mathscr{X}\}$ be the class of loss functions associated with $\mathscr{H}$.

Let $\bar{\mathbf{x}} = (\bar{x}_1, \cdots, \bar{x}_p)^\top$ be a p-dimension interval-valued vector, where $\bar{x}_j = [x_j^{\mathrm{l}}, x_j^{\mathrm{r}}], j \in [p]$. Here, we denote $[p] = \{1, \cdots, p\}$. $\bar{\mathbb{R}}$ denotes the set of all real-valued intervals (closed) and $\bar{\mathbb{R}}^p$ denotes as the set of all p-dimension interval-valued vectors, i.e., $\bar{\mathbb{R}} = \{[x^{\mathrm{l}}, x^{\mathrm{r}}] : x^{\mathrm{l}}, x^{\mathrm{r}} \in \mathbb{R}, x^{\mathrm{l}} \leq x^{\mathrm{r}}\}$ and $\bar{\mathbb{R}}^p = \{([x_1^{\mathrm{l}}, x_1^{\mathrm{r}}], \cdots, [x_p^{\mathrm{l}}, x_p^{\mathrm{r}}])^\top : x_j^{\mathrm{l}}, x_j^{\mathrm{r}} \in \mathbb{R}, x_j^{\mathrm{l}} \leq x_j^{\mathrm{r}}, j \in [p]\}$. $\bar{\mathscr{X}}^T \subset \bar{\mathbb{R}}^p$ denotes as the input space of the target domain with the interval-valued observations. Next, let $\boldsymbol{g}$ be a transformation function that aims to extract crisp-valued information from interval-valued observations,

$$\begin{aligned} &\boldsymbol{g}: \bar{\mathscr{X}}^T \to \mathscr{X}^T \\ &\bar{\mathbf{x}}^t = (\bar{x}_1^t, \cdots, \bar{x}_p^t)^\top \to (g_1(\bar{\mathbf{x}}^t), \cdots, g_K(\bar{\mathbf{x}}^t))^\top. \end{aligned} \tag{6.4}$$

Next, the risk is defined as follows:

$$R_{\mathscr{D}}(\boldsymbol{h}) = \mathbb{E}_{\mathscr{D}}[\ell(\boldsymbol{h}(\mathbf{x}), \mathbf{f}(\mathbf{x}))]. \tag{6.5}$$

Based on the definition of the ordinary MSDA problem, we identify our newly proposed problem.

Definition 6.1 (MSDA with Interval-valued Target Data)**.** Let $\mathscr{S} = \{\mathscr{S}^1, \mathscr{S}^2, \cdots, \mathscr{S}^N\}$ denote a set of source domains with N different sources, where $\mathscr{S}^n = \{(\mathbf{x}_i^{S_n}, y_i) | \mathbf{x}_i^{S_n} \in \mathscr{X}^n, y_i \in \mathscr{Y}\}_{i=1}^{m_n}$ is a single-source domain drawn i.i.d. from $\mathscr{D}^n$, $n \in [N]$. Here, $\mathscr{X}^n \subset \mathbb{R}^p$ denotes the feature space of each source domain and $\mathscr{Y} = \{1, 2, \cdots, K\}$ denotes the label space. $\bar{\mathscr{T}} = \{\bar{\mathbf{x}}_i^T | \bar{\mathbf{x}}_i^T \in \bar{\mathscr{X}}^T\}_{i=1}^{m_t}$ is the unlabeled target domain, where $\bar{\mathscr{X}}^T \subset \bar{\mathbb{R}}^p$ is the feature space of the target domain. Then, we denote $\mathscr{X}^T = \boldsymbol{g}(\bar{\mathscr{X}}^T)$ and $\mathscr{D}^T$ as the distribution of $\mathscr{X}^T$. Let $\mathbf{f}_T$ be the ground truth function of $\mathscr{X}^T$. Thus, our aim is to train a classifier $\boldsymbol{h}^t \in \mathscr{H}$ with $\mathscr{S}$ and $\bar{\mathscr{T}}$ s.t. $\boldsymbol{h}^t = \text{argmin}_{\boldsymbol{h} \in \mathscr{H}} R_{\bar{\mathscr{T}}}(\boldsymbol{h})$, where $R_{\bar{\mathscr{T}}}(\boldsymbol{h}) = \mathbb{E}_{\mathbf{x} \sim \mathscr{D}^T}[\ell(\boldsymbol{h}(\mathbf{x}), \mathbf{f}_T(\mathbf{x}))]$.

6.3 Theoretical Analysis

In this section, we give the theoretical analysis of the proposed problem.

First, we review the generation bound of the target risk. The discrepancy distance between the source and target domains disc_L-distance, is defined as follows:

$$\text{disc}_L(\mathscr{S}, \mathscr{T}) = 2 \sup_{(\boldsymbol{h}, \boldsymbol{h}') \in \mathscr{H} \times \mathscr{H}} |\mathbb{E}_{\mathscr{S}}[\ell(\boldsymbol{h}, \boldsymbol{h}')] - \mathbb{E}_{\mathscr{T}}[\ell(\boldsymbol{h}, \boldsymbol{h}')]|. \tag{6.6}$$

Using this notion, we have the following theorem.

Theorem 6.1 ([6])**.** *Let $\mathscr{H}$ be the hypothesis set of $\mathscr{X}$. Given two different domains, $\mathscr{S}$ and $\mathscr{T}$. Then, for any $\delta > 0$ with a probability of at least $1 - \delta$ (over the choice of the samples), for any $\boldsymbol{h} \in \mathscr{H}$:*

$$R_{\mathscr{T}}(\boldsymbol{h}) \leq R_{\mathscr{S}}(\boldsymbol{h}) + \frac{1}{2}\text{disc}_L(\mathscr{S}, \mathscr{T}) + \lambda, \tag{6.7}$$

where $\lambda = R_{\mathscr{S}}(\boldsymbol{h}^*) + R_{\mathscr{T}}(\boldsymbol{h}^*)$, $\boldsymbol{h}^* = \arg\min_{\boldsymbol{h} \in \mathscr{H}} R_{\mathscr{S}}(\boldsymbol{h}) + R_{\mathscr{T}}(\boldsymbol{h})$.

Let $\boldsymbol{h}_i, i \in [N]$ be the classifier trained on the single-source domain $\mathscr{S}_i$. Then, the multi-source classifier $\boldsymbol{h}^M$, which combines all single-source domain classifiers, is defined as:

$$\boldsymbol{h}^M(\mathbf{x}) = \sum_{i=1}^{N} \omega_i \boldsymbol{h}_i(\mathbf{x}), \tag{6.8}$$

where $\mathbf{w} = (\omega_1, \cdots, \omega_N)^\top \in \mathbb{R}^N$ is the weight vector and $\sum_{i=1}^{N} \omega_i = 1$. We note that for any $\boldsymbol{h}_i \in \mathscr{H}, \mathbf{w} \in \mathbb{R}^N$, $\boldsymbol{h}^M$ can cover the whole of $\mathscr{H}$.

Next, we derive the following theorem to bound the risk of the target domain.

Theorem 6.2. *Let $\mathscr{H}$ be the hypothesis set of $\mathscr{X}$. Let $\mathscr{S} = \{\mathscr{S}^1, \mathscr{S}^1, \cdots, \mathscr{S}^N\}$ denotes the multiple source domains and $\bar{\mathscr{T}}$ is the unlabeled target domain. We denote $\mathscr{X}^T = \boldsymbol{g}(\bar{\mathscr{X}}^T)$ and $\mathscr{D}^T$ as the distribution of $\mathscr{X}^T$. Suppose ℓ is a convex function. Then, for any $\delta > 0$ with a probability of at least $1-\delta$ (over the choice of the samples), for any $\boldsymbol{h}_i \in \mathscr{H}, i \in [N]$:*

$$R_{\bar{\mathscr{T}}}(\boldsymbol{h}^M) \leq \sum_{i=1}^{N} \omega_i R_{\mathscr{S}_i}(\boldsymbol{h}_i) + \frac{1}{2} \sum_{i=1}^{N} \omega_i \mathrm{disc}_L(\mathscr{S}_i, \mathscr{T}) + \lambda. \tag{6.9}$$

Proof.

$$\begin{aligned} R_{\bar{\mathscr{T}}}(\boldsymbol{h}^M) &= \mathbb{E}_{\mathbf{x} \sim \mathscr{D}^T}[\ell(\boldsymbol{h}^M(\mathbf{x}), \mathbf{f}_T(\mathbf{x}))] \\ &= \int_{\mathscr{D}^T} \ell(\sum_{i=1}^{N} \omega_i \boldsymbol{h}_i(\mathbf{x}), \mathbf{f}_T(\mathbf{x})) p_{\mathscr{D}^T}(\mathbf{x}) d\mathbf{x} \\ &\leq \sum_{i=1}^{N} \omega_i \int_{\mathscr{D}^T} \ell(\boldsymbol{h}_i(\mathbf{x}), \mathbf{f}_T(\mathbf{x})) p_{\mathscr{D}^T}(\mathbf{x}) d\mathbf{x} \\ &= \sum_{i=1}^{N} \omega_i R_{\bar{\mathscr{T}}}(\boldsymbol{h}). \end{aligned} \tag{6.10}$$

The derivation from the second line to the third line is because ℓ is a convex function, so we have $\ell(\sum_{i=1}^{N} \omega_i \boldsymbol{h}_i(\mathbf{x}), \mathbf{f}_T(\mathbf{x})) \leq \sum_{i=1}^{N} \omega_i \ell(\boldsymbol{h}_i(\mathbf{x}), \mathbf{f}_T(\mathbf{x}))$. According to Eq. (6.7), we have

$$
\begin{aligned}
R_{\bar{\mathcal{T}}}(\boldsymbol{h}^M) &\leq \sum_{i=1}^{N} \omega_i R_{\bar{\mathcal{T}}}(\boldsymbol{h}) \\
&\leq \sum_{i=1}^{N} \omega_i (R_{\mathcal{S}_i}(\boldsymbol{h}_i) + \frac{1}{2}\mathrm{disc}_L(\mathcal{S}_i, \mathcal{T}) + \lambda) \\
&= \sum_{i=1}^{N} \omega_i R_{\mathcal{S}_i}(\boldsymbol{h}_i) + \frac{1}{2}\sum_{i=1}^{N} \omega_i \mathrm{disc}_L(\mathcal{S}_i, \mathcal{T}) + \lambda
\end{aligned} \tag{6.11}
$$

■

This theorem indicates that the risk on the target domain is upper bounded by three terms: the weighted risk on the multi-source domain ($\sum_{i=1}^{N} \omega_i R_{\mathcal{S}_i}(\boldsymbol{h}_i)$), the weighted domain divergence distance between each source domain and the target domain ($\sum_{i=1}^{N} \omega_i \mathrm{disc}_L(\mathcal{S}_i, \mathcal{T})$), and the error of the ideal joint hypothesis (λ). Since λ does not depend on any particular $\boldsymbol{h}_i$, our focus should primarily be on minimizing the first two terms. To minimize the first term, we train $\boldsymbol{h}_i$ on each individual source domain. Regarding the reduction of the distribution discrepancy between the source and target domains, various approaches have been explored in existing domain adaptation works, such as MMD [93] or adversarial learning [48]. Furthermore, by assigning a smaller weight ω_i to larger $\mathrm{disc}_L(\mathcal{S}_i, \mathcal{T})$, the second term can be minimized. In the next section, we propose a novel fuzzy distance-based method for calculating the weight vector $\mathbf{w}$.

6.4 Model Construction

6.4.1 Fuzzy Multi-Adversarial Training Neural Networks

This section outlines our first model called *fuzzy multi-adversarial training neural networks* (FUMAT-Net). The framework of the proposed model is shown in Fig. 6.1.

First, we let $\boldsymbol{g} = \mathbf{T}(\bar{\mathbf{x}}; \beta)$ (See Eq. (5.11)) to extract crisp-valued information from the interval-valued observations in the target domain. Then, $\mathcal{T} = \mathbf{T}(\bar{\mathcal{T}}; \beta) \triangleq \{\mathbf{x}_i^T\}_{i=1}^{m_t}$ as the

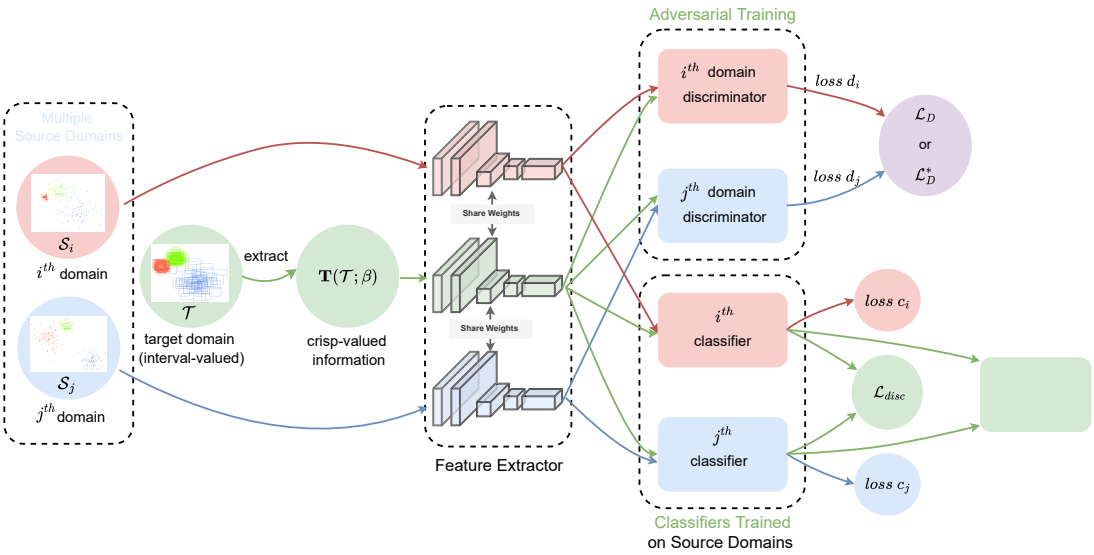


Figure 6.1: The framework of **Fuzzy Multi-Adversarial Training Neural Networks (FUMAT-Net)**. There are four main components: i) interval information extractor ii) feature extractor, iii) adversarial training, and iv) classifiers. Our model takes the labeled multi-source data with crisp-valued features and unlabeled target data with interval-valued features as input and transfers the learned knowledge to classify the unlabeled target samples. Without loss of generality, we show the i-th domain and j-th domain as an example. First, we use an interval information extractor to extract crisp-valued information from interval-valued target data. Then, the feature extractor maps the source domains into a common feature space. The adversarial training aims to align the distribution of the i-th and j-th source domains with the target domain. The final predictions of target samples are obtained by the weighted outputs of the i-th and j-th classifiers.

extracted information from $\tilde{\mathcal{T}}$. The first framework contains three main components, i.e., a feature extractor, adversarial training and classifiers (see Fig. 6.1).

Feature Extractor We design a feature extractor $F(\cdot)$ that shares weights on the multiple source domains and the target domain to extract common representations for all domains, which maps the features from the original feature space into a common feature space.

Adversarial Training We apply adversarial training to align the distribution between the multiple source domains and the target domain. Let $D_n, n \in [N]$ be the domain discriminators and D_n is used to discriminate $F(\mathbf{x}), \mathbf{x} \in \mathcal{S}_n \cup \mathcal{T}$ from $\mathcal{S}_n$ and $\mathcal{T}$. The

adversarial training loss of D_n is formulated as:

$$loss\ d_n = \sum_{i=1}^{m_n+m_t} L_d(D_n \circ F(\mathbf{x}_i), d_i), \tag{6.12}$$

where $\mathbf{x}_i \in \mathscr{S}_n \cup \mathscr{T}$, L_d is the loss function for predicting the domain labels, and d_i is the domain indicator (source: $d_i = 0$, target: $d_i = 1$). Then, the total adversarial training loss is formulated as:

$$\mathscr{L}_D = \frac{1}{N} \sum_{n=1}^{N} loss\ d_n. \tag{6.13}$$

Further, small value of $loss\ d_n, n \in [N]$ means that the distribution discrepancy between $\mathscr{S}_n$ and $\mathscr{T}$ is large. Therefore, it is hard to transfer valuable knowledge from this source domain with a small value of $loss\ d_n$ to the target domain. Based on above discussion, the optimization process of Eq. (6.13) may spend too much computational resources in optimizing the source domains that have large distribution discrepancy between the target domain. Then, we reformulate Eq. (6.13) to get the following soft version:

$$\mathscr{L}_D^* = \frac{1}{|n : L_n \geq \alpha|} \sum_{n:L_n \geq \alpha} loss\ d_n, \tag{6.14}$$

where $L_n = loss\ d_n / \sum_{n=1}^{N} loss\ d_n, n \in [N]$. $\mathscr{L}_D^*$ only considers to align the distribution between the source domain and the target domain with $L_n \geq \alpha, n \in [N], \alpha \in [0,1]$. We called this soft version as FUMAT-Net*.

Classifiers on Multiple Source Domains Let $C_n, n \in [N]$ be classifiers on multiple source domains and C_n is design to minimize the misclassification loss on $\mathscr{S}_n$. Then, the overall classification loss is shown as follow:

$$\mathscr{L}_C = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{m_n} L_c(C_n \circ F(\mathbf{x}_i^{\mathscr{S}_n}), y_i), \tag{6.15}$$

where L_c is the loss function for the category label prediction. In addition, there is only one target domain. Intuitively, the same target sample predicted by different classifiers

C_n should get the same prediction. Hence, we need to minimize the discrepancy among all classifiers. In this paper, the absolute values of the difference between all pairs of classifiers' probabilistic outputs of target sample are applied as discrepancy loss:

$$\mathscr{L}_{disc} = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N} \sum_{k=1}^{m_t} |C_i \circ F(\mathbf{x}_k^T) - C_j \circ F(\mathbf{x}_k^T)|. \tag{6.16}$$

In training process, $D_n, n \in [N]$ is connected to F via a gradient reversal layer that multiplies the gradient during the backpropagation-based training, which is inspired by DANN [48]. Let θ_f be the parameter of F, θ_{d_n} be the parameters of the domain discriminator D_n and θ_{c_n} be the parameters of the classifier C_n. Then, the overall training objective of FUMAT-Net is formulated as:

$$\begin{aligned} &\min_{\theta_f, \theta_{c_1}, \cdots, \theta_{c_N}} \max_{\theta_{d_1}, \cdots, \theta_{d_N}} \mathscr{L}_{total}, \\ &\mathscr{L}_{total} = \mathscr{L}_C + \lambda \mathscr{L}_D + \gamma \mathscr{L}_{disc}, \end{aligned} \tag{6.17}$$

where the parameters λ and γ are used to trade-off the adversarial training loss and discrepancy loss with the classification loss, respectively. For FUMAT-Net*, we replace $\mathscr{L}_D$ to $\mathscr{L}_D^*$ in above objective.

In the testing phase, testing data from the target domain are forwarded through the feature extractor and the N classifiers. The final prediction of the data in the unlabeled target domain is the weighted average of the outputs from the N classifiers, i.e., $\hat{y}^T = \sum_{n=1}^{N} w_n C_n \circ F(\mathbf{x}^T)$.

During above process, how to choose the shape parameter β and how to derive the weight vector $\mathbf{W} = (w_1, \cdots, w_N)^\top$ are two critical problem. To address these problems, a new fuzzy relation R is designed to measure the correlation between the multiple source domains and the target domain. Next, the definition of this new fuzzy relation R is shown as follow.

Definition 6.2. Given m fuzzy vectors, $\widetilde{\mathbf{x}}_i = (\widetilde{x}_{i1}, \cdots, \widetilde{x}_{ip})^\top$, $i \in [m]$, where $\mu_{\widetilde{x}_{il}}(t)$ is the

membership function of $\widetilde{x}_{il}$. We define an operator $R:(\widetilde{\mathbf{x}}_i,\widetilde{\mathbf{x}}_j)\rightarrow[0,1]$, where

$$R(\widetilde{\mathbf{x}}_i,\widetilde{\mathbf{x}}_j)=1-\frac{\sum_{l=1}^{p}|\int t\mu_{\widetilde{x}_{il}}(t)dt-\int t\mu_{\widetilde{x}_{jl}}(t)dt|^2}{R_{\max}},$$
$$R_{\max}=\max_{1\leq i,j\leq m}\sum_{l=1}^{p}\left|\int t\mu_{\widetilde{x}_{il}}(t)dt-\int t\mu_{\widetilde{x}_{jl}}(t)dt\right|^2.$$

Obviously, R satisfies the reflexivity and symmetry, so R is a fuzzy relation. Then, we apply this fuzzy relation to measure the correlation between the multiple source domains and the target domain. Let $\mathrm{Cor}(\mathscr{S}_n,\bar{\mathscr{T}})$ denote as the correlation between $\mathscr{S}_n$ and $\bar{\mathscr{T}}$, where

$$\mathrm{Cor}(\mathscr{S}_n,\bar{\mathscr{T}})=\frac{\sum_{i=1}^{m_n}\sum_{j=1}^{m_t}R(\widetilde{\mathbf{x}}_i^{S_n},\widetilde{\mathbf{x}}_j^{T})}{\sum_{n=1}^{N}\sum_{i=1}^{m_n}\sum_{j=1}^{m_t}R(\widetilde{\mathbf{x}}_i^{S_n},\widetilde{\mathbf{x}}_j^{T})}.$$

Here, $\widetilde{\mathbf{x}}_i^{S_n}=(\widetilde{x}_{i1}^{S_n},\cdots,\widetilde{x}_{i1}^{S_n})$ is a fuzzy vector that the membership function of $\widetilde{x}_{ik}^{S_n},k\in[p]$ is denoted as

$$\mu_{\widetilde{x}_{ik}^{S_n}}(t)=\begin{cases}1, & t=x_{ik}^{S_n}\\ 0, & \text{else,}\end{cases}$$

and $\widetilde{\mathbf{x}}_j^{T}=\mathbf{F}(\mathbf{x}_j^{T})$. The larger $\mathrm{Cor}(\mathscr{S}_n,\bar{\mathscr{T}})$, the more similar between $\mathscr{S}_n$ and $\bar{\mathscr{T}}$. Therefore, we select the parameter β from $\{0,0.1,\cdots,1\}$ that can achieve the largest $\max_{n\in[N]}\mathrm{Cor}(\mathscr{S}_n,\bar{\mathscr{T}})$, i.e.,

$$\beta^*=\underset{\beta\in\{0,0.1,\cdots,1\}}{\operatorname{argmax}}\{\max_{n\in[N]}\mathrm{Cor}(\mathscr{S}_n,\bar{\mathscr{T}})\}.$$

In addition, we set $w_n=\mathrm{Cor}(\mathscr{S}_n,\bar{\mathscr{T}}),n\in[N]$ to handle the second problem.

6.4.2 Fuzzy Distance-based Information Maximization Neural Networks

This section outlines our second model called *fuzzy distance-based information maximization neural networks* (FDIM-Net). The overall framework of FDIM-Net is shown in Fig

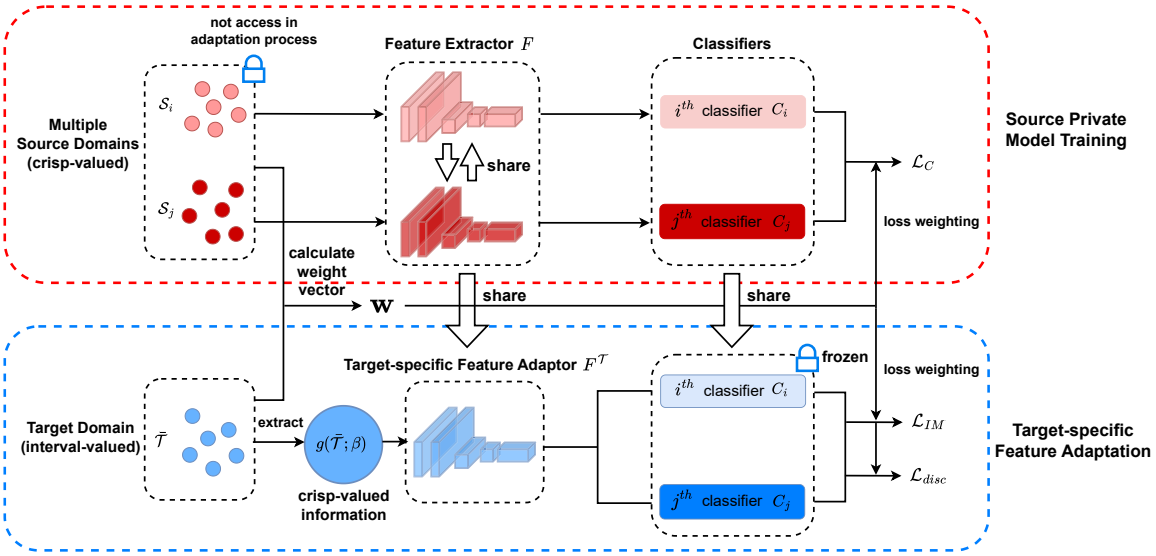


Figure 6.2: FDIM-Net framework. In the training process, we first calculate the weight vector and train the feature extractor and each classifier on each single source domain. Then, target data is applied to adapt the feature extractor by minimizing the information maximization loss and discrepancy loss. In the testing phase, the final prediction of the data in the unlabeled target domain is the weighted average of the outputs from the multiple classifiers.

6.2. which comprises three main components, i.e., a feature extractor F, a target-specific feature adaptor $F^{\mathcal{T}}$ and multiple classifiers $C_n, n \in [N]$.

6.4.2.1 Fuzzy Distance-based Fuzzy Kernel Function

We first introduce four distinct fuzzy distances designed to measure the distance between fuzzy numbers. These proposed fuzzy distances serve as the foundation for identifying a new fuzzy kernel function, which quantifies the correlation between the multiple source domains and the target domain.

First, a formal definition of fuzzy distance is present which is a natural extension of Euclidean distance.

Definition 6.3. Let $\mathscr{F}_{\mathbb{R}}$ be a set of all fuzzy real numbers induced by the real number system $\mathbb{R}$. Let $\widetilde{d} : \mathscr{F}_{\mathbb{R}} \times \mathscr{F}_{\mathbb{R}} \to \mathbb{R}$ denote a fuzzy distance if it obeys the following properties:

- **Symmetric**: $\forall \widetilde{A}_1, \widetilde{A}_2 \in \mathscr{F}_{\mathbb{R}}, \widetilde{d}(\widetilde{A}_1, \widetilde{A}_2) = \widetilde{d}(\widetilde{A}_2, \widetilde{A}_1)$.

- **Positive**:$\forall \widetilde{A}_1, \widetilde{A}_1 \in \mathscr{F}_{\mathbb{R}}, \widetilde{d}(\widetilde{A}_1, \widetilde{A}_2) > 0$, while $\forall \widetilde{A} \in \mathscr{F}_{\mathbb{R}}, \widetilde{d}(\widetilde{A}, \widetilde{A}) = 0$.

- **Triangle inequality** : $\forall \widetilde{A}_1, \widetilde{A}_2, \widetilde{A}_2 \in \mathscr{F}_{\mathbb{R}}, \widetilde{d}(\widetilde{A}_1, \widetilde{A}_2) + \widetilde{d}(\widetilde{A}_2, \widetilde{A}_3) \geq \widetilde{d}(\widetilde{A}_1, \widetilde{A}_3)$.

For $\widetilde{\mathbf{A}}_i = (\widetilde{A}_{i1}, \cdots, \widetilde{A}_{ip})^\top \in \mathscr{F}_{\mathbb{R}^p}, i = 1,2$, $\widetilde{d}(\widetilde{\mathbf{A}}_1, \widetilde{\mathbf{A}}_2) = \sum_{j=1}^{p} \widetilde{d}(\widetilde{A}_{1j}, \widetilde{A}_{2j})$.

Fuzzy distance gives a crisp value to measure the distance between two fuzzy numbers, which helps us intuitively feel the direct discrepancy between two fuzzy numbers. Next, we present four different types of fuzzy distances. Let $\widetilde{A}_1, \widetilde{A}_2 \in \mathscr{F}_{\mathbb{R}}$ be two fuzzy numbers with fuzzy membership functions $\mu_{\widetilde{A}_1}(t), \mu_{\widetilde{A}_2}(t)$ and $[\widetilde{A}_\alpha^L, \widetilde{A}_\alpha^U]$ is the α-cut of a fuzzy number $\widetilde{A}$. Then, the four different types of fuzzy distances are defined as follows:

Type 1. $\widetilde{d}_1(\widetilde{A}_1, \widetilde{A}_2) = |\int \mu_{\widetilde{A}_1}(t)dt - \int \mu_{\widetilde{A}_2}(t)dt|$.

Type 2. $\widetilde{d}_2(\widetilde{A}_1, \widetilde{A}_2) = |\int t\mu_{\widetilde{A}_1}(t)dt - \int t\mu_{\widetilde{A}_2}(t)dt|$.

Type 3. $\widetilde{d}_3(\widetilde{A}_1, \widetilde{A}_2) = [\int_0^1 (\widetilde{A_{1\alpha}}^U - \widetilde{A_{2\alpha}}^U)^2 dt - \int_0^1 (\widetilde{A_{1\alpha}}^L - \widetilde{A_{2\alpha}}^L)^2 d\alpha]^{\frac{1}{2}}$.

Type 4.[144] $\widetilde{d}_4(\widetilde{A}_1, \widetilde{A}_2) = \left[\frac{\int_0^1 [(\frac{\widetilde{A_{1\alpha}}^L + \widetilde{A_{1\alpha}}^U}{2} - \frac{\widetilde{A_{2\alpha}}^L + \widetilde{A_{2\alpha}}^U}{2})^2 + \frac{1}{3}(\frac{\widetilde{A_{1\alpha}}^U - \widetilde{A_{1\alpha}}^L}{2})^2 + \frac{1}{3}(\frac{\widetilde{A_{2\alpha}}^U - \widetilde{A_{2\alpha}}^L}{2})^2] \alpha d\alpha}{\int_0^1 \alpha d\alpha} \right]^{\frac{1}{2}}$

The first two are based on fuzzy membership functions and the last two are based on the α-cut. According to Definition 6.3, we can easily prove $\widetilde{d}_1, \widetilde{d}_2, \widetilde{d}_3, \widetilde{d}_4$ are all fuzzy distances. Based on the four defined fuzzy distances, the distribution discrepancy between the source and target domains can be estimated. MMD is the most common statistic used to estimate the discrepancy between two different distributions. Let $\mathscr{S} = \{(\mathbf{x}_i^S, y_i) | \mathbf{x}_i^S \in \mathscr{X}, y_i \in \mathscr{Y}\}_{i=1}^{m_s}$ and $\bar{\mathscr{T}} = \{\bar{\mathbf{x}}_i^T | \bar{\mathbf{x}}_i^T \in \bar{\mathscr{X}}^T\}_{i=1}^{m_t}$. From [53], we can estimate MMD using the U-statistic estimator that is unbiased for MMD^2 and has nearly minimal variance among

unbiased estimators:

$$\widehat{\mathrm{MMD}}_u^2(\mathscr{S},\bar{\mathscr{T}};k) = \frac{1}{m_s(m_s-1)}\sum_{i=1}^{m_s}\sum_{j\neq i}^{m_s} k(\mathbf{x}_i^S,\mathbf{x}_j^S) + \frac{1}{m_t(m_t-1)}\sum_{i=1}^{m_t}\sum_{j\neq i}^{m_t} k(\bar{\mathbf{x}}_i^T,\bar{\mathbf{x}}_j^T) - \frac{2}{m_s m_t}\sum_{i=1}^{m_s}\sum_{j=1}^{m_t} k(\mathbf{x}_i^S,\bar{\mathbf{x}}_j^T), \tag{6.18}$$

where $k(\cdot,\cdot)$ is a kernel function. In the proposed problem, the target domain only contains interval-valued data so traditional kernel functions can not be used. A large degree of uncertainty exists in the interval-valued data. Fuzzy techniques can improve machine learning algorithms by providing a way to handle different types of uncertain problems. Therefore, we apply fuzzy techniques to handle the above problem. First, we design two fuzzilization functions $\widetilde{f}_s, \widetilde{f}_t$ to transfer crisp-valued data ($\mathbf{x}_i^S$) and interval-valued data ($\bar{\mathbf{x}}_i^T$) into fuzzy-valued data. $\widetilde{f}_t(\cdot;\beta) = \mathbf{F}_t(\cdot;\beta)$ (See Eq. (5.11)). Then, a new fuzzy distance-based kernel function is defined:

$$k_{\widetilde{d}}(\widetilde{\mathbf{x}}_i,\widetilde{\mathbf{x}}_j) = \exp\left(-\frac{\widetilde{d}(\widetilde{\mathbf{x}}_i,\widetilde{\mathbf{x}}_j)^2}{2\sigma^2}\right), \tag{6.19}$$

where $\widetilde{d} \in \{\widetilde{d}_1,\cdots,\widetilde{d}_4\}$. Then, Eq. (6.18) can be revised by using this fuzzy distance-based kernel function:

$$\begin{aligned}\widehat{\mathrm{MMD}}_u^2(\mathscr{S},\bar{\mathscr{T}};k_{\widetilde{d}},\widetilde{f}_s,\widetilde{f}_t) = {} & \frac{1}{m_s(m_s-1)}\sum_{i=1}^{m_s}\sum_{j\neq i}^{m_s} k_{\widetilde{d}}(\widetilde{f}_s(\mathbf{x}_i^S),\widetilde{f}_s(\mathbf{x}_j^S)) \\ & + \frac{1}{m_t(m_t-1)}\sum_{i=1}^{m_t}\sum_{j\neq i}^{m_t} k_{\widetilde{d}}(\widetilde{f}_t(\bar{\mathbf{x}}_i^T),\widetilde{f}_t(\bar{\mathbf{x}}_j^T)) \\ & - \frac{2}{m_s m_t}\sum_{i=1}^{m_s}\sum_{j=1}^{m_t} k_{\widetilde{d}}(\widetilde{f}_s(\mathbf{x}_i^S),\widetilde{f}_t(\bar{\mathbf{x}}_j^T)).\end{aligned} \tag{6.20}$$

Let $\mathbf{d}^{\mathrm{MMD}} = (d_1^{\mathrm{MMD}},\cdots,d_N^{\mathrm{MMD}})^\top$, where each $d_n^{\mathrm{MMD}} = \widehat{\mathrm{MMD}}_u^2(\mathscr{S}^n,\bar{\mathscr{T}};k_{\widetilde{d}},\widetilde{f}_s,\widetilde{f}_t), n \in [N]$ is the estimate of distribution discrepancy between $\mathscr{S}^i$ and $\bar{\mathscr{T}}$. According to the theoretical analysis in Section 6.3, the weight vector $\mathbf{w} = (\omega_1,\cdots,\omega_N)^\top$ is calculated as follows:

$$\mathbf{w} = \mathbf{d}^{'}/\sum_{n=1}^{N} d_n^{'}, \mathbf{d}^{'} = 1 - \frac{\mathbf{d}^{\mathrm{MMD}}}{\max_{n\in[N]}|d_n^{\mathrm{MMD}}|}. \tag{6.21}$$

From Eq. (6.21), larger d_n^{MMD}, smaller ω_n and $\sum_{n=1}^{N}\omega_n = 1$.

However, the data in the multiple source domains are crisp-valued from our setting, we need to design a fuzzification function $\widetilde{f}_s$ (see Eq. (6.20)) to transfer them in to interval-valued features. Let $\mathbf{x}_i^{S_n} = (x_{i1}^{S_n}, \cdots, x_{ip}^{S_n})^\top$ be crisp-valued data in the source domain $\mathscr{S}^n, n \in [N]$. $\widetilde{f}_s$ is defined as follows:

$$\widetilde{f}_s(\mathbf{x}_i^{S_n}) = (\widetilde{f}_s(x_{i1}^{S_n}), \cdots, \widetilde{f}_s(x_{ip}^{S_n}))^\top,$$
$$\widetilde{f}_s(x_{ij}^{S_n}) = \mathrm{Tr}(x_{ij}^{S_n} - \epsilon, x_{ij}^{S_n}, x_{ij}^{S_n} + \epsilon), j \in [p].$$

where $\widetilde{x} = \mathrm{Tr}(a_1, b_1, a_2)$ is a a triangular fuzzy number.

6.4.2.2 Source Private Model Training

First, we train the feature extractor F and multiple classifiers $C_n, n \in [N]$ to obtain the multiple source private model. Feature extractor F shares weights across multiple source domains. Let $h_n = C_n \circ F, n \in [N]$. The overall classification loss on the multiple source domain $\mathscr{S}$ is shown as follows:

$$\mathscr{L}_C = \sum_{n=1}^{N} \omega_n \ell_c(h_n(\mathbf{x}^{\mathscr{S}_n}), y), \tag{6.22}$$

where ℓ_c is the loss function for the category label prediction. $\mathbf{w} = (\omega_1, \cdots, \omega_N)^\top$ is the weight vector that is calculated by our proposed fuzzy distance-based method (see Eq. (6.21)). The commonly used loss is the cross-entropy loss function. In this paper, we apply the adjusted cross-entropy loss function with the label smoothing technique to increase the discriminability as proposed in [106].

6.4.2.3 Crisp-valued Information Extracting

Similar as the first proposed model, we let $\boldsymbol{g} = \mathbf{T}(\bar{\mathbf{x}}; \beta)$ (See Eq. (5.11)) to extract crisp-valued information from the interval-valued observations in the target domain. During this process, how to choose the shape parameter β is a critical problem. We apply the

estimate of distribution discrepancy $d_n^{\text{MMD}} = \widehat{\text{MMD}}_u^2(\mathscr{S}^n, \bar{\mathscr{T}}; k_{\tilde{d}}, \widetilde{f}_s, \widetilde{f}_t), n \in [N]$ to address this problem. Specifically, parameter β is selected from $\{0, 0.1, \cdots, 1\}$ that can obtain the smallest $\sum_{n=1}^{N} d_n^{\text{MMD}}$, i.e.,

$$\beta^* = \arg \min_{\beta \in \{0, 0.1, \cdots, 1\}} \sum_{n=1}^{N} d_n^{\text{MMD}}. \tag{6.23}$$

The selected β^* makes the distribution between $\mathscr{S}$ and $\bar{\mathscr{T}}$ closer. Therefore, we can more easily adapt the trained multiple source private model to fit the target domain.

6.4.2.4 Target-specific Feature Adaptation

During adaptation process on the target domain, we can not access the source data, only the trained source private model is available. Therefore, a target-specific feature adaptor is designed to fine-tun the learned feature extractor from multiple source domains. This process promotes the feature extractor to extract target-specific representations. Inspired by [83], the *information maximization* (IM) loss is adopted to achieve this purpose. Adopting the IM loss will ensure that the target outputs are individually certain yet globally diverse. This implies that the target outputs can be similar to one-hot encoding but differ from each other. This particular form represents the ideal target outputs when mitigating the gap between the source and target domain. Therefore, the IM loss stands as a powerful tool for SFDA. The adaptation loss is shown as follows:

$$\mathscr{L}_{IM} = \sum_{n=1}^{N} \omega_n \ell_{IM}(h_n(\mathbf{x}^T)), \tag{6.24}$$

where $\mathbf{x}^T = \boldsymbol{g}(\bar{\mathbf{x}}^T; \beta)$ and ℓ_{IM} is the IM loss function. ℓ_{IM} is defined as follows:

$$\begin{aligned} \ell_{IM}(h_n(\mathbf{x}^T)) = &- \mathbb{E}_{\mathbf{x}^T} \sum_{k=1}^{K} \delta_k(h_n(\mathbf{x}^T)) \log(\delta_k(h_n(\mathbf{x}^T))) \\ &+ \sum_{k=1}^{K} \mathbb{E}_{\mathbf{x}^T}(\delta_k(h_n(\mathbf{x}^T))) \log \mathbb{E}_{\mathbf{x}^T}(\delta_k(h_n(\mathbf{x}^T))), \end{aligned} \tag{6.25}$$

where $\delta_k(a)$ denotes the k-th element in the soft-max output of a K-dimensional vector a.

Additionally, we try to minimize the discrepancy among all classifiers. In this chapter, the absolute values of the difference between all pairs of classifiers' probabilistic outputs of the target sample are applied as discrepancy loss:

$$\mathscr{L}_{disc} = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N} \sum_{k=1}^{m_t} |\delta(h_i(\mathbf{x}_k^T)) - \delta(h_j(\mathbf{x}_k^T))|. \tag{6.26}$$

Then, the overall training objective of our proposed model is formulated as:

$$\mathscr{L}_{total} = \mathscr{L}_C + \mathscr{L}_{IM} + \lambda \mathscr{L}_{disc}, \tag{6.27}$$

where parameters λ are used to trade-off $\mathscr{L}_{disc}$ with $\mathscr{L}_C$ and $\mathscr{L}_{IM}$.

In the testing phase, the final prediction of the data in the unlabeled target domain is the weighted average of the outputs from the N classifiers, i.e.,

$$\hat{y}^T = \arg\max_{k \in [K]} \sum_{n=1}^{N} w_n \delta_k(h_n(\mathbf{x}^T)).$$

More details of FDIM-Net are provided in Algorithm 5. Step 2 aims to select the optimal parameter β^* for the fuzzy transformation function. Step 3 involves calculating the extracted information from the interval-valued target domain and the weight vector $\mathbf{w}$. Steps 4 to 6 encompass the process of training models on multiple source domains based on Eq. (6.22). Subsequently, the trained source models are adapted to the target domain by minimizing L_{IM} (see Eq. (6.24)) and L_{disc} (see Eq. (6.26)) in Steps 7 to 8. Finally, Step 9 predicts the data in the unlabeled target domain via the weighted average of the outputs from the learned multiple classifiers. Furthermore, our algorithm does not require access to the source data during the adaptation process, making our proposed method an SFDA method. The source data owners can simply send the trained models to the target data owner for model adaptation. Therefore, if the source data owners and the target data owner belong to different entities, applying our method can prevent source data privacy leakage.

Algorithm 5 FDIM-Net

Input: data $\mathscr{S}^n = \{(\mathbf{x}_i^{S_n}, y_i) | \mathbf{x}_i^{S_n} \in \mathscr{X}^n, y_i \in \mathscr{Y}\}_{i=1}^{m_n}, n \in [N]$, $\bar{\mathscr{T}} = \{\bar{\mathbf{x}}_i^T | \bar{\mathbf{x}}_i^T \in \bar{\mathscr{X}}^T\}_{i=1}^{m_t}$, learning rate LR, epochs T_0, T_{max}, and optimization algorithm (Stochastic Gradient Descent (SGD) [11] is selected);
Initial: parameters of $F, C_1, \cdots, C_N$;

1: Select optimal parameter β^* according to Eq. (6.23);
2: Compute $\mathscr{T} = \boldsymbol{g}(\bar{\mathscr{T}}; \beta^*) \triangleq \{\mathbf{x}_i^T\}_{i=1}^{m_t}$ and weight vector $\mathbf{w}$ according to Eq. (6.21);
3: Fetch mini-batches from $\mathscr{S}^n, n \in [N]$;
4: **for** $i = 1, 2, \ldots, T_0$ **do** (// *Train on the multiple source domains*)
5: Train $F, C_1, \cdots, C_N$ with mini-batches from $\mathscr{S}^n, n \in [N]$ to minimize L_C (see Eq. (6.22));
6: **end for**
7: Fetch mini-batches from $\mathscr{T}$;
8: **for** $T = 1, 2, \ldots, T_{max}$ **do** (// *Adapt trained source model for the target domain*)
9: Train $F, C_1, \cdots, C_N$ with mini-batches from $\mathscr{T}$ to minimize L_{IM} (see Eq. (6.24)) and L_{disc} (see Eq. (6.26));
10: **end for**

Output: $\hat{y}^T = \arg\max_{k \in [K]} \sum_{n=1}^{N} w_n \delta_k(h_n(\mathbf{x}^T))$.

6.5 Experiments

In this section, we initially compare the performance of our proposed second model using four different fuzzy distances on synthetic datasets to determine the most suitable fuzzy distance. Subsequently, we validate the effectiveness of the proposed algorithms in addressing the proposed problem by comparing it with several baseline methods in terms of classification accuracy on both synthetic and real-world datasets. Finally, we conduct experiments to present the results of the ablation study and the parameter sensitivity analysis of our proposed method.

6.5.1 Baselines

First, we briefly introduce several state-of-the-art baselines for comparison with our proposed algorithms. Since no existing algorithm can be used to directly address the proposed problem, we use the midpoint of each interval-valued feature to convert the interval-valued datasets into crisp-valued datasets. Then, the baseline methods can be used to solve the proposed problem. Single-domain adaptation algorithms include:

- DAN: Deep Adaptation Network [93];
- DANN: Domain-adversarial Neural Network[48];
- CDAN: Conditional Domain Adversarial Networks [94];
- DWL: Dynamic Weighted Learning [159].
- G-SFDA: Generalized source-free domain adaptation [168].

MSDA algorithms include:

- M^3SDA: Moment Matching [117];
- CMSS: Curriculum Manager for Source Selection [167];
- SHOT: Source Hypothesis Transfer with Information Maximization [83];
- LtC-MSDA: Learning to Combine [150];
- CAiDA: Confident anchor-induced [39];
- PTMDA: Multi-source Unsupervised Domain Adaptation via Pseudo Target Domain [124];
- SF-FDN: Source-Free Multi-Domain Adaptation with Fuzzy Rule-based Deep Neural Networks [80].

The term “Source model only" refers to the use of the complete model trained on the source domain for target label prediction. In the case of single-domain adaptation algorithms, we employ two standards for evaluation. (1) *Single Best*: we report the best performance of the single-source domain adaptation algorithm among all single source domains. (2) *Source Combine*: We combine all multiple source domains into a single source domain, and then domain adaptation is performed in a traditional single-domain adaptation scenario.

Table 6.1: Parameter setting of our method.

Parameter	Synthetic dataset	Real-world dataset
Learning rate LR	0.01	0.01
T_0	50	50
T_{max}	500	100
ϵ	0.0001	0.0001
σ	0.5	0.5
Batch size	100	500

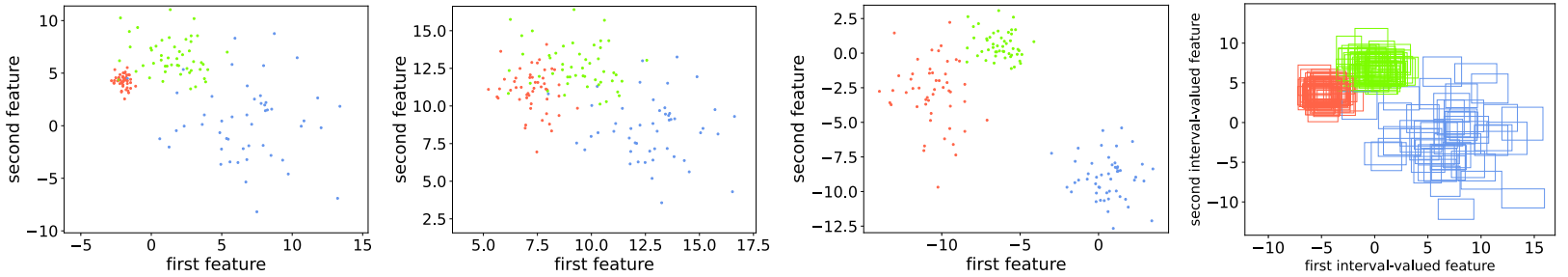
6.5.2 Experimental Setup

The parameter settings are presented in Table 6.1. The trade-off parameter λ was set to match that of DANN [48] to ensure a fair comparison. For consistency, the feature extractor used in all methods was a three-layer network with ReLU activation and Dropout applied in all layers. The hidden layer consisted of 100 units across all layers. The classifiers of our proposed algorithms and all baselines were implemented as one-layer networks with dimensions of (100 × #classes). Stochastic gradient descent (SGD) [11] was employed as the optimization algorithm for all methods, with a momentum of 0.9 and a weight decay of 1×10^{-8}. The final experiment results were obtained by averaging the outcomes of 5 repeated experiments. Classification accuracy was used as the evaluation metric for the performance on the target domain across all methods.

Data Generation: To generate our synthetic datasets, we propose an intervalization method designed for our problem setting. Firstly, a crisp-valued dataset with K categories was generated using a random number generator, denoted as $\mathscr{T} = \{(\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})^\top, y_i)\}_{i=1}^n$. Subsequently, we utilized the generated crisp-valued dataset to construct the interval-valued dataset $\bar{\mathscr{T}} = \{\bar{\mathbf{x}}_i = (\bar{x}_{i1}, \cdots, \bar{x}_{ip})^\top, y_i\}_{i=1}^n$, where each $\bar{x}_{ij}$ is an interval characterized by $[x_{ij} - a_{ij}, x_{ij} + b_{ij}]$. Here, $a_{ij} \sim U[a_1, b_1], b_{ij} \sim U[a_2, b_2]$ and $U[a, b]$ denotes a uniform distribution over $[a, b]$.

Table 6.2: Parameter setting to generate the synthetic dataset.

Parameter	$\mathscr{S}_1$	$\mathscr{S}_1$	$\mathscr{S}_1$	$\bar{\mathscr{T}}$
M	500	500	500	500
K	3	3	3	3
p	2	2	2	2
$\mathbf{C}$	[0.5, 1.8, 3.5]	[1.2, 1.5, 2]	[2, 1, 1.5]	[0.6, 1, 3.5]
$\mathbf{R}$	(-5, 10)	(5, 15)	(-15, 5)	(-10, 10)
$[a_1, b_1], [a_2, b_2]$	-	-	-	[0.1, 0.5], [2, 4]



(a) The first source domain. (b) The second source domain. (c) The third source domain. (d) The target domain.

Figure 6.3: Synthetic datasets. (a),(b),(c) depict the samples from the generated multiple crisp-valued source domains and (d) depicts the samples from the generated interval-valued target domain.

Next, we describe the construction of a synthetic dataset designed to align with our identified problem setting. In this chapter, we utilize a Gaussian data generator denoted as $\mathbb{G}(M,K,p,\mathbf{C},\mathbf{R})$ as a random number generator. Here, M represents the number of generated data points, K represents the number of categories, p represents the dimension of the generated data, $\mathbf{C}$ controls the standard deviations of the generated categories, and $\mathbf{R}$ controls the ranges of the generated data. Subsequently, we apply this data generator to generate multiple source domains $\mathscr{S} = \{\mathscr{S}^1, \mathscr{S}^1, \cdots, \mathscr{S}^N\}$ using $\mathbb{G}(M_i^s, K, p, \mathbf{C}_i^s, \mathbf{R}_i^s)$ for $i \in [N]$. Additionally, a crisp-valued target dataset $\mathscr{T}$ is generated using $\mathbb{G}(M^t, K, p, \mathbf{C}^t, \mathbf{R}^t)$. Finally, we employ the intervalization method mentioned earlier to generate the interval-valued target domain $\bar{\mathscr{T}}$.

Following the data generation mechanism described above, we generated a synthetic dataset consisting of three source domains and one target domain. The parameter values used for the dataset generation are provided in Table 6.2. Visualizations of the samples in the generated synthetic dataset are depicted in Fig. 6.3.

Experiments Description: In the first experiment, we evaluate the performance of FDIM-Net using four different fuzzy distances on the synthetic dataset to select the optimal fuzzy distance. In subsequent experiments, we employ the selected optimal fuzzy distance to obtain the experiment results of FDIM-Net. Furthermore, we validate the effectiveness of our proposed algorithms by comparing it with several baselines in terms of classification accuracy on the synthetic dataset.

Results and Analysis: Figure 6.4 illustrates the experimental results of the first experiment, indicating that the type 3 fuzzy distance achieves the best performance compared to the other types. This can be attributed to the fact that the type 3 fuzzy distance allows for the selection of the optimal shape parameter β. By utilizing the optimal shape parameter β, our proposed method can effectively extract valuable crisp-valued information from the interval-valued target domain.

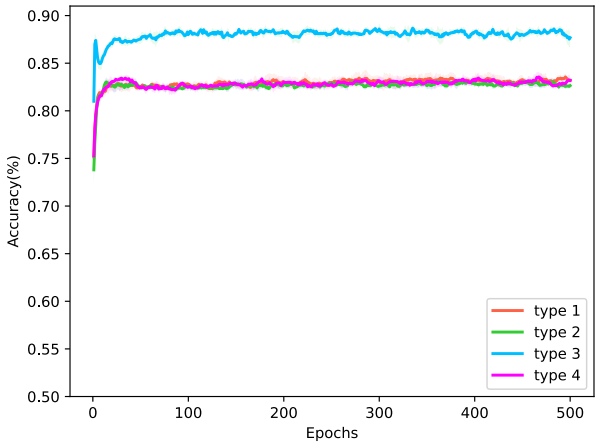
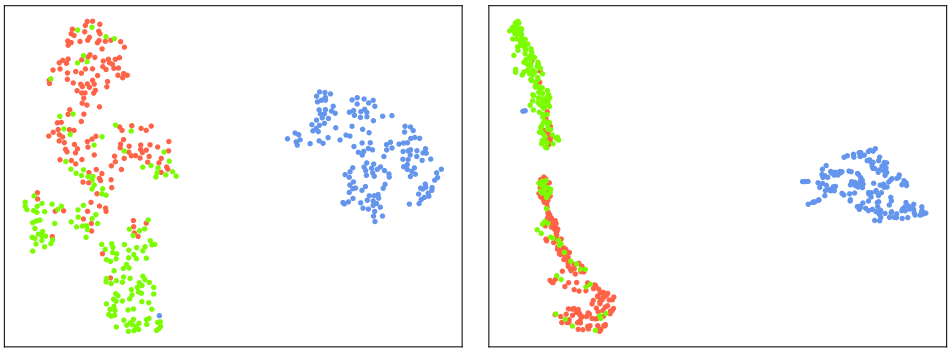


Figure 6.4: Classification accuracy on the synthetic dataset with different fuzzy distances.



(a) Source only. (b) Ours.

Figure 6.5: The t-SNE visualizations of the target data features on the synthetic dataset.

The comparison results on the synthetic dataset are presented in Table 6.3. The table clearly shows that our proposed method outperforms all non-fuzzy baselines, thereby demonstrating the superiority of the proposed fuzzy techniques-based method in addressing our proposed problems. Fig. 6.5 depicts the data visualization of the target data features within the synthetic dataset. It is observable that after the application of FDIM-Net for adaptation, the target classes are distinctly separated from each other. Only a small portion of the target data exhibits less pronounced separation.

Additionally, we observe that the results obtained from the 'Single Best' methods surpass those obtained from the 'Source Combine' methods and most multi-source

Table 6.3: Performance Comparison of Classification Accuracy on the synthetic dataset.

Standards	Methods	Mean accuracy(%)	STD(%)	Time(s)
Single Best	Source model only	85.08	1.15	41.7
	DAN[93]	85.64	1.20	52.3
	DANN[48]	86.36	0.75	42.3
	CDAN+E[94]	85.88	0.79	65.6
	DWL[159]	86.95	0.75	156.3
	G-SFDA[168]	85.85	0.55	178.6
Source Combine	Source model only	73.88	2.96	13.9
	DAN[93]	73.96	4.89	17.4
	DANN[48]	74.04	2.09	14.1
	CDAN+E[94]	76.28	5.61	21.9
	DWL[159]	82.20	1.02	52.1
	G-SFDA[168]	72.32	1.06	59.6
Multi-source	M^3SDA[117]	84.84	0.77	36.1
	CMSS[167]	76.40	3.66	33.1
	SHOT[83]	86.44	1.20	56.8
	LtC-MSDA [150]	84.98	0.92	67.1
	CAiDA [39]	85.55	2.39	98.3
	PTMDA [124]	83.28	2.45	116.2
	SF-FDN [80]	86.55	1.01	289.6
	FUMAT-Net	85.25	1.58	227.9
	FUMAT-Net*	86.71	1.04	234.9
	FDIM-Net	**87.48**	0.76	245.8

The bold value represents the highest accuracy.

baselines. This can be attributed to the similarity between the distribution of the first source domain and the target domain, while the distributions of the last two source domains significantly differ from the target domain (refer to Fig. 6.3). Consequently,

during the adaptation process, the 'Source Combine' methods and most multi-source baselines are influenced by the dissimilarities introduced by the last two source domains, leading to a degradation in performance. As our method does not require the use of source data during the adaptation process, it successfully overcomes this drawback. Furthermore, our method outperforms another SOTA fuzzy logic-based MSDA method [80]. The primary reason for this superiority lies in the fact that the referenced method overlooks the inherent uncertainty correlation between different source domains and the target domain, whereas our proposed fuzzy distance-based neural network effectively addresses this critical issue.

Moreover, we present the running time of a single complete experiment for the proposed algorithms and all baselines in Table 6.3. Utilizing fuzzy techniques inherently leads to an increase in the computational complexity for our algorithm. However, considering the substantial improvement in model performance, a marginal increase in algorithm complexity remains acceptable.

6.5.3 Experiments on Real-world Datasets

Dataset Description: The Weather dataset, obtained from the RP5 website, serves as our real-world dataset and includes meteorological data for four American cities and one UK city, spanning from January 1, 2016 to December 31, 2021. The dataset comprises the following cities: Seattle Tacoma (**S**), Olympia (**O**), Portland (**P**), Washington (**W**), and London (**L**). In each instance of **O**, **W**, **P**, and **L**, the meteorological data represents measurements at a specific time and is represented using crisp values. Conversely, in **S**, each instance corresponds to the meteorological data recorded over the course of a day and is represented using interval values. The data from both the source and target datasets consist of six variables: air temperature (T), atmospheric pressure at the weather station level (P_0), atmospheric pressure reduced to the main sea level

Table 6.4: Performance Comparison of Ablation Study on the real-world dataset (**OPW → S**).

Standards	Methods	Mean accuracy(%)	STD(%)
Multi-source	FUMAT-mean	78.12	1.07
	FUMAT-Net	78.39	1.07
	FUMAT-mean*	79.45	1.54
	FUMAT-Net*	79.70	1.47

(P), humidity (U), dew-point temperature (T_d), and a categorical variable indicating the presence or absence of precipitation (encoded as 0 for No Precipitation and 1 for Precipitation).

Experiments Description: From the set of available datasets **O**, **W**, **P**, and **L**, we select two datasets to serve as the multiple source domains, while **S** is designated as the target domain. As a result, we construct six distinct adaptation tasks for our experiments: **OW → S**, **OP → S**, **OL → S**, **WP → S**, **WL → S**, and **PL → S**. For each of these tasks, we report the experiment results of both the baselines and FDIM-Net in Table 6.6.

Results and Analysis: From Table 6.6, it can be seen that our proposed method achieves the highest average accuracy and the lowest standard deviation on the real-world dataset. Notably, FDIM-Net achieves the best accuracy in four of the six tasks: **OW → S**, **OL → S**, **WP → S**, and **PL → S**. These results further reinforce the superiority of the proposed fuzzy technique-based method. Through the utilization of fuzzy techniques, FDIM-Net enhances the efficiency of transfer learning and achieves robust transfer performance when dealing with interval-valued target data.

6.5.4 Ablation Study

Frist, we verify the rationality of the proposed new fuzzy relation for measuring the correlation between the multiple source domains and the target domain. We let the

Table 6.5: Ablation Study Results.

Task	Ours w/o fuzzy	Ours
Synthetic	85.60 ± 1.54	87.48 ± 0.76
OW → S	78.03 ± 1.98	80.24 ± 0.19
OP → S	78.51 ± 0.68	80.60 ± 0.27
OL → S	77.85 ± 1.24	79.56 ± 0.22
WP → S	78.19 ± 1.97	80.25 ± 0.65
WL → S	78.50 ± 0.82	79.79 ± 0.66
PL → S	77.19 ± 2.08	80.25 ± 0.41

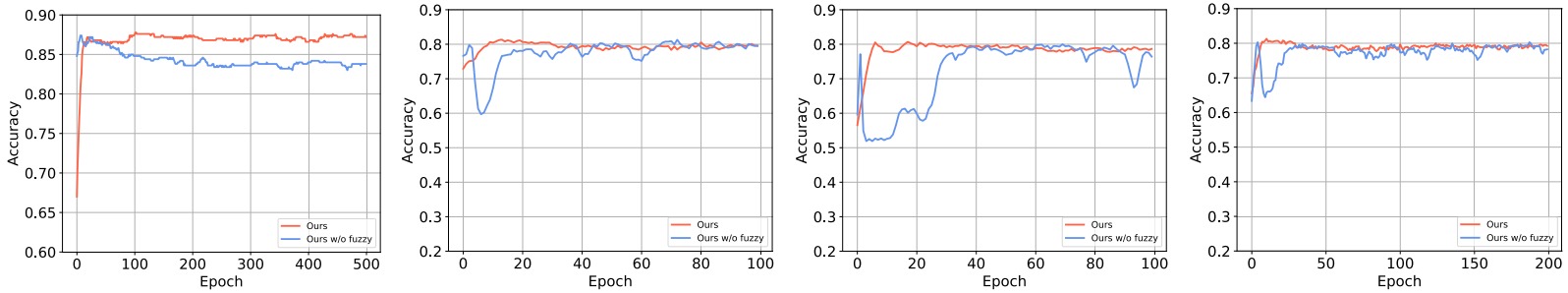
weight vector $\mathbf{W} = (\frac{1}{N}, \cdots, \frac{1}{N})^\top$ to form two new methods, denoted as FUMAT-mean and FUMAT-mean*. Then, we compare these two methods with our methods on the real-world dataset. From Table 6.4, our methods gain better performance than FUMAT-mean and FUMAT-mean*, which demonstrates the rationality of the proposed new fuzzy relation.

Second, we validate the rationality of the proposed new fuzzy distance in addressing our proposed problems. We set the weight vector $\mathbf{w} = (1, \cdots, 1)^\top$ in the loss function and $\mathbf{w} = (\frac{1}{N}, \cdots, \frac{1}{N})^\top$ to obtain the final prediction, which is denoted as Ours w/o fuzzy. We present the comparison results on both synthetic and real-world datasets in Table 6.5. Additionally, Fig. 6.6 illustrates the variation of classification accuracy on the target domain with the number of epochs. Notably, in Figs. 6.6(b), 6.6(c), and 6.6(d), the performance of Ours w/o fuzzy exhibits significant fluctuations as the number of epochs increases. These results collectively demonstrate that the proposed fuzzy distance not only improves performance but also enhances the robustness of our model.

Table 6.6: Accuracy (mean ± std %) on the real-world dataset.

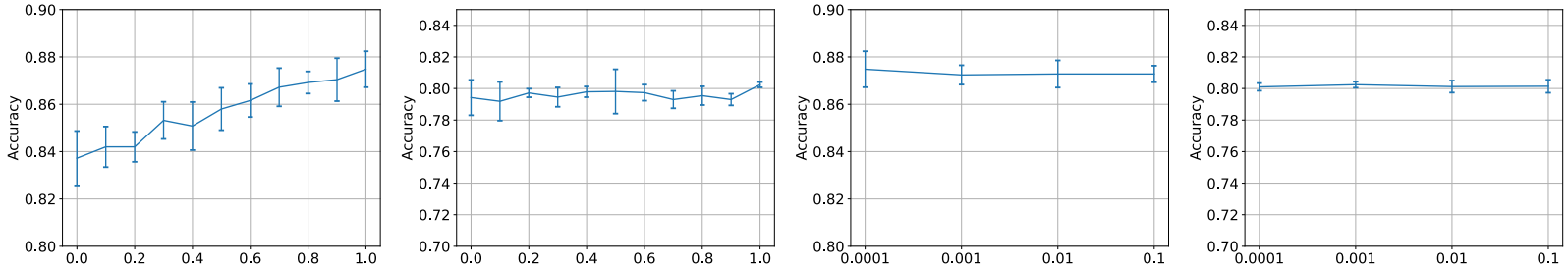
Standards	Methods	**OW → S**	**OP → S**	**OL → S**	**WP → S**	**WL → S**	**PL → S**	Average
Single Best	Source model only	72.75 ± 3.93	78.76 ± 0.69	71.59 ± 1.77	78.76 ± 0.69	72.75 ± 3.93	78.76 ± 0.69	75.56
	DAN[93]	69.44 ± 1.35	71.36 ± 1.27	66.17 ± 1.78	71.36 ± 1.27	69.44 ± 1.35	71.36 ± 1.27	69.86
	DANN[48]	72.77 ± 1.83	79.11 ± 0.87	72.50 ± 0.56	79.11 ± 0.87	72.77 ± 1.83	79.11 ± 0.87	75.90
	CDAN+E[94]	70.30 ± 0.94	72.63 ± 0.95	65.98 ± 0.57	72.63 ± 0.95	70.30 ± 0.94	72.63 ± 0.95	71.14
	DWL[159]	79.47 ± 1.06	79.47 ± 1.06	79.47 ± 1.06	79.15 ± 0.50	78.66 ± 1.05	79.15 ± 0.50	79.23
	G-SFDA[168]	78.56 ± 0.99	79.60 ± 2.39	78.12 ± 1.89	79.60 ± 2.39	78.56 ± 0.99	79.60 ± 2.39	79.01
Source Combine	Source model only	74.35 ± 1.81	73.44 ± 2.60	59.09 ± 0.80	77.79 ± 2.35	53.5 ± 0.43	60.16 ± 0.99	66.39
	DAN[93]	66.12 ± 1.47	67.28 ± 1.02	52.57 ± 1.48	70.19 ± 1.90	57.98 ± 3.18	57.58 ± 3.26	61.95
	DANN[48]	70.74 ± 0.99	72.60 ± 2.02	62.17 ± 0.77	74.23 ± 1.77	57.15 ± 0.99	64.26 ± 0.62	66.86
	CDAN+E[94]	66.57 ± 0.79	66.86 ± 0.67	53.46 ± 1.03	70.73 ± 0.50	53.10 ± 3.02	55.78 ± 0.78	61.08
	DWL[159]	79.57 ± 0.51	80.48 ± 0.74	77.93 ± 1.03	78.15 ± 1.45	**80.51 ± 1.26**	80.23 ± 0.83	79.48
	G-SFDA[168]	80.16 ± 1.68	80.80 ± 1.25	79.26 ± 1.21	80.04 ± 0.57	78.32 ± 2.46	76.16 ± 2.65	79.12
Multi-source	M^3SDA[117]	71.35 ± 1.35	75.84 ± 0.48	63.28 ± 1.29	75.02 ± 0.77	61.18 ± 1.99	66.72 ± 1.24	68.89
	CMSS[167]	71.08 ± 1.75	73.62 ± 1.62	62.42 ± 1.16	74.31 ± 0.73	57.68 ± 1.29	63.96 ± 1.35	67.18
	SHOT[83]	77.16 ± 0.66	81.21 ± 1.12	78.35 ± 0.46	78.15 ± 0.94	76.75 ± 0.96	75.14 ± 1.49	77.79
	LtC-MSDA [150]	78.80 ± 1.47	80.17 ± 0.79	76.14 ± 1.98	79.48 ± 1.15	76.18 ± 1.58	75.51 ± 2.41	77.71
	CAiDA [39]	79.85 ± 0.60	79.75 ± 0.57	79.50 ± 0.76	80.10 ± 0.46	79.30 ± 0.19	79.75 ± 0.63	79.71
	PTMDA [124]	77.62 ± 2.33	77.73 ± 2.30	78.06 ± 1.83	77.99 ± 2.31	76.94 ± 2.13	79.01 ± 1.56	77.89
	SF-FDN[80]	78.76 ± 0.56	**81.81 ± 0.72**	79.15 ± 0.32	79.75 ± 0.54	78.06 ± 0.84	77.32 ± 1.12	79.14
	Ours	**80.24 ± 0.19**	80.60 ± 0.27	**79.56 ± 0.22**	**80.25 ± 0.65**	79.79 ± 0.66	**80.25 ± 0.41**	**80.13**

The bold value represents the highest accuracy in each column.



(a) Synthetic dataset (b) **OW → S** (c) **WL → S** (d) **PL → S**

Figure 6.6: Evaluation metrics on the target domain varies with the number of epochs.



(a) Synthetic dataset (b) **OW → S** (c) Synthetic dataset (d) **OW → S**

Figure 6.7: Evaluation metrics on the target domain varies with the value of parameters β and ϵ. (a),(b) show the parameter sensitivity analysis of β on the synthetic dataset and real-world task **OW → S**. (c),(d) show the parameter sensitivity analysis of ϵ on the synthetic dataset and real-world task **OW → S**.

6.5.5 Parameter Sensitivity Analysis

In this section, the parameter sensitivity analysis of our proposed method is detailed.

To analyze parameter β, we select values from $\{0, 0.1, \cdots, 1\}$. For each value of β, we obtain the results using the same experimental procedure as described in Section 6.5.2. Figs. 6.7(a) and 6.7(b) depict the mean and standard deviation of the classification accuracy on the synthetic dataset and the real-world task $\mathbf{OW} \rightarrow \mathbf{S}$. From these figures, it can be observed that the optimal value of β is equal to 1.0 for both the synthetic and real-world datasets. This finding validates the rationality of our proposed fuzzy distance, as the selected optimal β of 1.0 consistently aligns with the fuzzy distance defined in Eq. (6.23). Regarding parameter ϵ, we select values from $\{0.0001, 0.001, 0.01, 0.1\}$. The experiment process is identical to that of the shape parameter β. Figs. 6.7(c) and 6.7(d) demonstrate that FDIM-Net is not highly sensitive to parameter ϵ.

6.6 Summary

This chapter identifies a more realistic problem called MSDA with interval-valued target data, where we aim to learn a new model for interval-valued target data by leveraging knowledge from crisp-valued multiple source domains. To address this problem, two new fuzzy technique-based models are developed. The applied fuzzy techniques contain a fuzzy transformation function, fuzzy relation-based method, and a fuzzy distance-based method, where the fuzzy transformation function is applied to extract valuable crisp-valued information from interval-valued target data and the fuzzy relation-based or fuzzy distance-based method is designed to appropriately combine multi-source models. The experiment results on both synthetic and real-world datasets show the outstanding performance of our proposed fuzzy technique-based MSDA methods.

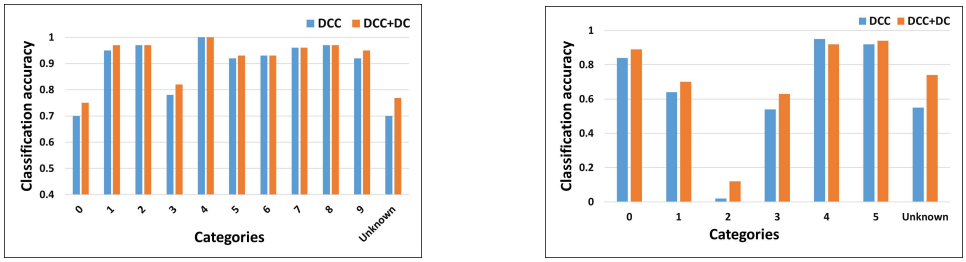
CHAPTER 7

DISTRACTION-CONTROL FOR UNIVERSAL DOMAIN ADAPTATION

7.1 Introduction

Domain adaptation [114] has significantly contributed to the advancement of image recognition tasks [45, 152]. It addresses the challenge of insufficient labeled data in developing a well-performing model by leveraging knowledge from a different domain (referred to as the source domain) that has abundant labeled data and transferring this knowledge to the target domain. Collecting adequate labeled data for constructing a model in the target domain can be extremely time-consuming and even infeasible in certain scenarios. Therefore, the ability to design a robust adaptation model capable of achieving outstanding performance in the target domain holds immense significance.

In traditional domain adaptation [136, 169], also known as closed-set domain adaptation (CDA) [48], the primary focus is on minimizing the discrepancy in feature distributions between the source and target domains, which typically exhibit different



(a) OfficeHome: **P** → **R** (b) VisDA

Figure 7.1: The horizontal coordinate of Figures (a) and (b) represents the categories in the target domain, while the vertical coordinate represents their classification accuracy. Figure (a) shows that when using a SOTA method (DCC), the classification accuracy for categories 0, 3, and “unknown" (the hard transfer categories) remains low. Similarly, in Figure (b), the classification accuracy for the hard transfer categories 1, 2, 3, and “unknown" is also low. The main reason for this is that DCC cannot effectively handle the significant discrepancies between the distributions present in the hard transfer categories. By contrast, DCC+DC shows a significant improvement in classification accuracy for these categories. These outcomes demonstrate the excellent ability of our proposed strategy to enhance performance on hard transfer categories.

distributions. However, CDA assumes that the source and target domains share a common label set, an assumption that does not always hold in real-world scenarios. Consequently, three special cases of domain adaptation have been identified: 1) partial domain adaptation (PDA), where the source domain contains private categories [15]; 2) open-set domain adaptation (OSDA), where the target domain contains private categories [44]; and 3) open-partial domain adaptation (OPDA), where both the source and target domains contain private categories [72]. Recently, a more general case called universal domain adaptation (UniDA) [171] has been proposed to address scenarios with no prior knowledge of the label set in the target domain. Importantly, UniDA encompasses all other special cases of domain adaptation, including CDA, PDA, OSDA, and OPDA.

The main challenge with UniDA is to address both the domain shift and category shift between the source and target domains. Existing UniDA methods [21, 129] have made remarkable achievements in reducing the negative effects caused by the private label set of the source domain and detecting unknown samples in the target domain.

However, a remaining drawback has been identified through numerous experiments with state-of-the-art (SOTA) UniDA methods: the classification accuracy in some categories remains low even after applying these methods. We refer to these categories as "hard transfer categories." For example, as shown in Figure 7.1, after applying DCC [76] to two real-world domain adaptation tasks in the OPDA setting, the classification accuracy for some categories still remains low, with some falling below 50%. Improving the accuracy for these hard transfer categories would lead to an overall performance improvement. Yet, the large differences in distribution between the source and target data in these particular categories make it difficult for existing UniDA methods to align the distributions. Moreover, existing methods assign equal attention to different categories, producing locally optimal solutions, which leads to their failure on hard transfer categories.

In this chapter, we provide a formal definition for identifying a hard transfer category along with a theoretical analysis that serves as a guide to addressing this problem. Following the theoretical analysis, we present our novel dynamic reweighted loss learning strategy, called Distraction-Control (DC), which enhances UniDA performance on the hard transfer categories. The aim of DC is to construct a weight vector that up-weights the contribution of the target data within the hard transfer categories. Additionally, the weight vector is dynamically updated during the adaptation process to ensure each hard transfer category receives an appropriate amount of attention as the distributions shift through adaptation. As a result, the domain adaptation model focuses more on improving performance in the hard transfer categories than in other easier categories. More importantly, by allocating more attention to the hard transfer categories, our strategy not only effectively mitigates domain shift but also proves highly effective in addressing category shift. In UniDA setting, we treat target-private categories as a single category called “unknow". From Figure 7.1, the category “unknown" has lower classification accuracy than most other categories and hence usually qualifies as a hard

transfer category. Therefore, by applying our strategy to existing models, these models can better focus on detecting target-private categories. The experiment result in Figure 7.1 of DCC+DC shows that using DC can enhance the ability to detect unknown classes. Similar to target-private categories, source-private categories also typically qualify as hard transfer categories. Consequently, applying DC can enhance the detection ability of source-private categories.

Understanding the main idea behind DC is best done through a real-world example. In our daily work, we often need to manage multiple tasks simultaneously. However, the complexity of these tasks varies, and so we often allocate more attention to solving the most challenging tasks to improve overall work efficiency. If we distributed our attention and effort equally among all tasks, the more difficult tasks might not be handled effectively. In our context, the hard transfer categories can be likened to the challenging tasks, and applying DC is analogous to our brain allocating more energy to difficult tasks.

To validate the efficacy of DC, we integrated it into several SOTA UniDA models, creating new variants of these models. We then compared the performance of the original models with the performance of their new variants on various real-world datasets through a comprehensive series of experiments. The results demonstrate that DC significantly improves the classification accuracy of the hard transfer categories in the target domain. Consequently, our proposed dynamic reweighted loss learning strategy effectively addresses the issue of hard transfer categories in UniDA scenarios and provides an overall enhancement in model performance.

Our contribution can be summarized as follows:

1. We have identified an unresolved issue in UniDA: the existing UniDA methods are not achieving satisfactory adaptation performance in certain categories, specifically the hard transfer categories.

2. To tackle this issue, we designed a theory to gain insights into this issue. Then, we develop a novel dynamic reweighted loss learning strategy called Distraction-control to handle this identified issue.

3. Extensive experiments on standard benchmarks demonstrate the effectiveness of the proposed strategy in enhancing model performance on hard transfer categories.

7.2 Problem Setting

Let $\mathscr{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ denote the source domain, where $\mathbf{x}_i^s \in \mathscr{X}_s$ represents the feature space and $y_i^s \in \mathscr{Y}_s$ represents the label space of the source domain. Similarly, let $\mathscr{D}_t = \{\mathbf{x}_i^t\}_{i=1}^{n_t}$ represent the unlabeled target domain, where $\mathbf{x}_i^t \in \mathscr{X}_t$ corresponds to the feature space of the target domain, and the label space of the target domain is denoted as $\mathscr{Y}_t$. $\mathscr{Y}_c = \mathscr{Y}_s \cap \mathscr{Y}_t$ denotes the common label set; $\bar{\mathscr{Y}}_s = \mathscr{Y}_s / \mathscr{Y}_c$ denotes the private label set of the source domain; and $\bar{\mathscr{Y}}_t = \mathscr{Y}_t / \mathscr{Y}_c$ denotes the private label set of the target domain. The main objective of UniDA is to train a target classifier that correctly classifies known target samples into their respective known classes, while classifying unknown target samples as “unknown".

Existing methods for UniDA employ various model frameworks to simultaneously handle the domain shift and category shift between the source and target domains. However, these methods commonly suffer from an extremely low classification accuracy for certain categories in the target domain. This issue can be attributed to significant distribution differences between the source and target domains, specifically for the hard transfer categories. In the next section, we present a theoretical analysis that guides the development of our strategy for addressing this problem.

7.3 Theoretical Analysis

This analysis begins with the formal definition of a hard transfer category. In the context of UniDA, the objective is to learn a classifier from both labeled source data and unlabeled target data to classify the target samples into $K = |\mathcal{Y}_c| + 1$ classes, where private samples in the target domain are considered to be a uniform unknown class. Let $\mathcal{H}$ be the hypothesis set of the UniDA task, and for any $\boldsymbol{h} \in \mathcal{H}$:

$$\begin{aligned} \boldsymbol{h} : \mathcal{X}_s &\to \mathbb{R}^K \\ \mathbf{x}_i^s &\to (h_1(\mathbf{x}_i^s), \cdots, h_K(\mathbf{x}_i^s))^\top. \end{aligned}$$

The loss function of $\boldsymbol{h}$ is defined as

$$\ell : \mathbb{R}^K \times \mathcal{Y}_t \to \mathbb{R}_+.$$

A hard transfer category is then defined as follows.

Definition 7.1 (L-Hard Transfer Category)**.** For any $\boldsymbol{h} \in \mathcal{H}$, $k \in \mathcal{Y}_c \cup$"unknown" $\triangleq \mathcal{Y}_c^u$ is denoted as a L-hard transfer category of $\boldsymbol{h}$, if there exists a positive constant L such that

$$\mathscr{L}^k = \sum_{i:y_i^t=k} \ell(\boldsymbol{h}(\mathbf{x}_i^t), k) / \sum_{i=1}^{n_t} \mathbf{1}_{y_i^t=k} > L. \tag{7.1}$$

Without loss of generality, we suppose ℓ is the $0-1$ loss. Then, $\mathscr{L}^k$ is equal to the misclassification rate of $\boldsymbol{h}$ on the k-th category. Therefore, Definition 7.1 treats a category with classification accuracy less than or equal to L as a L-hard transfer category. Figure 7.2 provides a visual display of the hard transfer category. Give a hypothesis function $\boldsymbol{h} \in \mathcal{H}$, let

$$\mathcal{K}_{\boldsymbol{h}} = \{k \in \mathcal{Y}_c^u | k \text{ is a } L\text{-hard transfer category of } \boldsymbol{h}\}$$

be the set of all L-hard transfer categories of $\boldsymbol{h}$. The empirical risk of $\boldsymbol{h}$ on the target domain is denoted as:

$$\widehat{R}_{\mathscr{D}_t}(\boldsymbol{h}) = \frac{1}{n_t} \sum_{i=1}^{n_t} \ell(\boldsymbol{h}(\mathbf{x}_i^t), y_i^t). \tag{7.2}$$

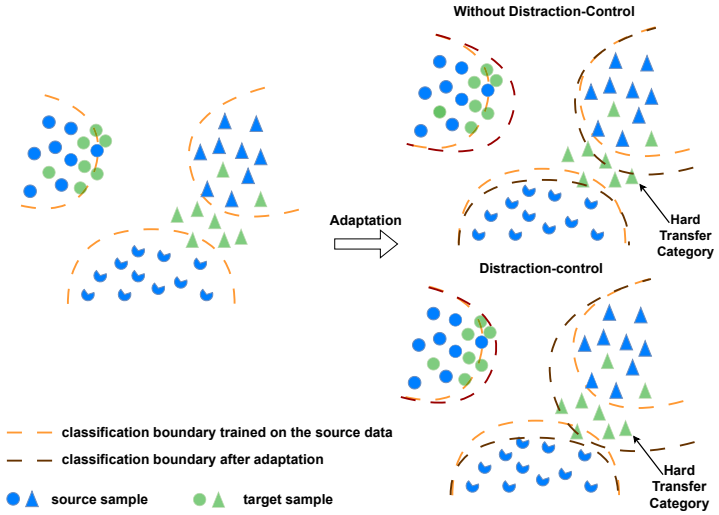


Figure 7.2: **Distraction-control.** In the top right of the figure, the trained classifiers of existing UniDA methods achieve unsatisfactory results on the hard transfer category. This is because these methods cannot change the amount of focus they give to each category. By contrast, in the bottom right of the figure, the DC method enables the trained model to exert more effort on the hard transfer category, thereby enhancing its performance.

Thus, we have the following theorem:

Theorem 7.1. *For any $\boldsymbol{h} \in \mathscr{H}$, the empirical risk of $\boldsymbol{h}$ on the target domain can be bounded as:*

$$\widehat{R}_{\mathscr{D}_t}(\boldsymbol{h}) \leq \frac{1}{n_t} \sum_{k \in \mathscr{K}_{\boldsymbol{h}}} \mathscr{L}^k \sum_{i=1}^{n_t} \mathbf{1}_{y_i^t = k} + \frac{LC}{n_t}, \tag{7.3}$$

where $C = |\{i \in [n_t] | y_i^t \in \mathscr{K}_{\boldsymbol{h}}^c\}|$. Here $\mathscr{K}_{\boldsymbol{h}}^c$ is complement of $\mathscr{K}_{\boldsymbol{h}}$ with respect to $\mathscr{Y}_c^u$.

Proof. First, we divide $\mathscr{Y}_t$ into two parts such that $\mathscr{Y}_t = \mathscr{K}_{\boldsymbol{h}} \cup \mathscr{K}_{\boldsymbol{h}}^c$. Then, according to Definition 7.1, we have

$$\begin{aligned}
\widehat{R}_{\mathscr{D}_t}(\boldsymbol{h}) &= \frac{1}{n_t} \sum_{i=1}^{n_t} \ell(\boldsymbol{h}(\mathbf{x}_i^t), y_i^t) \\
&= \frac{1}{n_t} [\sum_{y_i^t \in \mathscr{K}_{\boldsymbol{h}}} \ell(\boldsymbol{h}(\mathbf{x}_i^t), y_i^t) + \sum_{y_i^t \in \mathscr{K}_{\boldsymbol{h}}^c} \ell(\boldsymbol{h}(\mathbf{x}_i^t), y_i^t)] \\
&\leq \frac{1}{n_t} \sum_{k \in \mathscr{K}_{\boldsymbol{h}}} \mathscr{L}^k \sum_{i=1}^{n_t} \mathbf{1}_{y_i^t = k} + \frac{LC}{n_t}.
\end{aligned}$$

■

According to Theorem 7.1, $\widehat{R}_{\mathscr{D}_t}(\boldsymbol{h})$ is upper bounded by two terms, which are the empirical risk of $\boldsymbol{h}$ on $\mathscr{K}_{\boldsymbol{h}}$ and $\mathscr{K}_{\boldsymbol{h}}^c$, respectively. Since $|\{i \in [n_t] | y_i^t \in \mathscr{K}_{\boldsymbol{h}}^c\}| < n_t$ and $L \ll n_t$, we have $LC/n_t = O(1/n_t)$. The SOTA UniDA models have achieved remarkable success in minimizing the last term in Eq. (7.3), but they find it challenging to minimize the first term. Therefore, to enhance the overall performance of UniDA, our strategy intends to improve performance on the hard transfer categories, i.e., it helps to minimize the first term $\frac{1}{n_t} \sum_{k \in \mathscr{K}_{\boldsymbol{h}}} \mathscr{L}^k \sum_{i=1}^{n_t} \mathbf{1}_{y_i^t = k}$.

7.4 Distraction-control for UniDA

This section introduces the main idea of DC and how it is implemented.

7.4.1 Objective of Distraction-control

The problem in UniDA that we aim to resolve through theoretical analysis is how to improve classification accuracy in hard transfer categories. Existing UniDA methods pay equal attention to each category when aligning distributions throughout the adaptation process. However, it is evident that aligning distributions for hard transfer categories is more challenging than for other categories. Consequently, current UniDA methods often fail to align these distributions effectively, leading to lower classification accuracy. Our solution is a novel dynamic reweighted loss learning strategy called Distraction-control.

The key idea behind DC is to enable the trained model to focus more on aligning the distributions of the hard transfer categories. During the training process, higher weights are assigned to the loss for these hard transfer categories, ensuring that the model pays more attention to aligning the distributions in these challenging categories. Additionally, we observed that the hard transfer categories might change during training. Therefore,

Algorithm 6 Distraction-control

Input: source data $\mathscr{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$, target data $\mathscr{D}_t = \{\mathbf{x}_i^t\}_{i=1}^{n_t}$, and epochs T_{max}, T_0, T_1
Initial: network parameters of feature extractor F and classifier C and weight vectors $\boldsymbol{W} = (1, \cdots, 1)^\top$
Output: trained networks F, C

1: Fetch mini-batches from $\mathscr{D}_s$ and $\mathscr{D}_t$.
2: **for** $T = 1, 2, \ldots, T_{max}$ **do**
3: Train F, C with mini-batches from $\mathscr{D}_s$ and $\mathscr{D}_t$ to minimize weighted overall loss function $\ell((\mathbf{x}_i^s, y_i^s), \mathbf{x}_i^t; \boldsymbol{W})$.
4: **if** $T > T_0$ and T mod $T_1 = 0$ **then**
5: Calculate $C(\mathbf{x}_i^t) = (\hat{y}_{i1}^t, \cdots, \hat{y}_{i|\mathscr{Y}_s|+1}^t)^\top$ on target data $\mathscr{D}_t$.
6: Update the weight vector $\boldsymbol{W}$ using Eq.(7.5).
7: **end if**
8: **end for**

we incorporated a dynamic reweighting mechanism to ensure that the calculated weight vector consistently matches the hard transfer categories. An overview of DC is provided in Figure 7.2.

Applying DC not only effectively mitigates domain shift but also proves highly effective in addressing category shift. This is because source-private categories and target-private categories usually qualify as hard transfer categories. Consequently, through the application of DC, the model will pay more attention to detecting source-private and target-private categories. Thus, using our strategy can concurrently address both domain shift and category shift in UniDA.

7.4.2 The Implementation of Distraction-control

Our first step in implementing DC was to design a weight vector to assign higher weights to the hard transfer categories. According to Definition 7.1, the larger the value of $\mathscr{L}^k$, the more difficult it is to transfer knowledge to the corresponding category (k-th). Thus, such categories should be given a higher weight. If the value of $\mathscr{L}^k$ is known, the weight

vector $\boldsymbol{W}_0 = (\omega_1, \cdots, \omega_{|\mathscr{Y}_s|+1})^\top$ can be designed in the following form:

$$\omega_k = \mathscr{L}^k / \sum_{k=1}^{|\mathscr{Y}_s|+1} \mathscr{L}^k, \tag{7.4}$$

where $\mathscr{L}^k = 0$ when $k \in \bar{\mathscr{Y}}_s$. Unfortunately, labels for the target data are generally unavailable. Therefore, the weight vector cannot be calculated directly from Eq. (7.4). However, we observe that, if one category is a hard transfer category, i.e., it has a large value for $\mathscr{L}^k$, the average predicted confidence score of this category will be small. This implies that a larger weight should be assigned to a category with a small average predicted confidence score.

Let C be the trained classifier for the data prediction in the target domain, and let $C(\mathbf{x}_i^t) = (\hat{y}_{i1}^t, \cdots, \hat{y}_{i|\mathscr{Y}_s|+1}^t)^\top$ be the predicted confidence score of $\mathbf{x}_i^t$. $k^{'} = \arg\max_{k \in [|\mathscr{Y}_s|+1]} \hat{y}_{ik}^t$ denotes the category prediction made by the classifier C. Let $u_i = (u_{i1}, \cdots, u_{i|\mathscr{Y}_s|+1})^\top$, where $u_{ik'} = \hat{y}_{ik'}^t$, $u_{ik} = 0$ when $k \neq k^{'}$, $v_i = (v_{i1}, \cdots, v_{i|\mathscr{Y}_s|+1})^\top$, where $v_{ik'} = 1$, $v_{ik} = 0$ when $k \neq k^{'}$. The average predicted confidence score for each category is denoted as $\beta = (\beta_1, \cdots, \beta_{|\mathscr{Y}_s|+1})^\top$, where $\beta_k = \sum_{i=1}^{n_t} u_{ik} / \sum_{i=1}^{n_t} v_{ik}$.

Next, $\widehat{\mathscr{L}}^k = 1 - \beta_k / \max_{k \in [|\mathscr{Y}_s|]} \beta_k$ to replace $\mathscr{L}^k$ in Eq. (7.4) is used to obtain the new weight vector $\boldsymbol{W} = (\omega_1, \cdots, \omega_{|\mathscr{Y}_s|+1})^\top$:

$$\omega_k = \widehat{\mathscr{L}}^k / \sum_{k=1}^{|\mathscr{Y}_s|+1} \widehat{\mathscr{L}}^k. \tag{7.5}$$

In Eq. (7.5), a smaller value of β_k will result in a larger value of ω_k. Therefore, applying Eq. (7.5) to reweight the distribution alignment loss function will achieve DC's objective, which is to exert more effort to aligning the distributions of the hard transfer categories. In addition, since the average predicted confidence score associated with the source-private category typically remains low, through the application of DC to existing models, these models can also better focus on detecting source-private categories. Last, the weight vector $\boldsymbol{W}$ is dynamically updated to match any changes in the hard transfer categories in the overall training process. A more detailed description of how DC works can be found in Algorithm 6.

We modified two SOTA UniDA methods, namely DCC [76] and OVANet [129] to incorporate DC. With DCC, the distribution alignment loss (Contrastive Domain Discrepancy) was formulated as:

$$\mathscr{L}_{cdd} = \frac{1}{|\mathscr{Y}_s|}\sum_{k=1}^{|\mathscr{Y}_s|}\widehat{\mathscr{D}}^{kk} - \frac{1}{|\mathscr{Y}_s|(|\mathscr{Y}_s|-1)}\sum_{k=1}^{|\mathscr{Y}_s|}\sum_{k'=1,k'\neq k}^{|\mathscr{Y}_s|}\widehat{\mathscr{D}}^{kk'}. \tag{7.6}$$

More detail on $\widehat{\mathscr{D}}^{kk}$ can be found in [76]. Then, we reweighted the above loss function with $\boldsymbol{W}$ to obtain a new weighted loss function $\mathscr{L}_{cdd}^{DC}$:

$$\mathscr{L}_{cdd}^{DC} = \frac{1}{|\mathscr{Y}_s|}\sum_{k=1}^{|\mathscr{Y}_s|}\omega_k\widehat{\mathscr{D}}^{kk} - \frac{1}{|\mathscr{Y}_s|(|\mathscr{Y}_s|-1)}\sum_{k=1}^{|\mathscr{Y}_s|}\omega_k\sum_{k'=1,k'\neq k}^{|\mathscr{Y}_s|}\widehat{\mathscr{D}}^{kk'}. \tag{7.7}$$

With OVANet, the distribution alignment loss (Open-set Entropy Minimization) was formulated as:

$$\mathscr{L}_{ent} = -\frac{1}{n_t|\mathscr{Y}_s|}\sum_{\mathbf{x}^t\in\mathscr{D}_t}\sum_{k=1}^{|\mathscr{Y}_s|}\hat{y}_k^{t,op}\log(\hat{y}_k^{t,op}) + (1-\hat{y}_k^{t,op})\log(1-\hat{y}_k^{t,op}). \tag{7.8}$$

Where $\hat{y}_k^{t,op}$ is predicted by a One-vs-All open-set classifier (see [129]), $C_{\text{op}}(F(\mathbf{x}^t)) = (\hat{y}_1^{t,op}, 1-\hat{y}_1^{t,op}; \cdots; \hat{y}_{|\mathscr{Y}_s|}^{t,op}, 1-\hat{y}_{|\mathscr{Y}_s|}^{t,op})$. Then, we applied $\boldsymbol{W}$ in Eq. (7.8) to get a new weighted loss $\mathscr{L}_{ent}^{DC}$,

$$\mathscr{L}_{ent}^{DC} = -\frac{1}{n_t|\mathscr{Y}_s|}\sum_{\mathbf{x}^t\in\mathscr{D}_t}\sum_{k=1}^{|\mathscr{Y}_s|}\omega_k(\hat{y}_k^{t,op}\log(\hat{y}_k^{t,op}) + (1-\hat{y}_k^{t,op})\log(1-\hat{y}_k^{t,op})). \tag{7.9}$$

Remark 1. DCC and OVANet use additional structure to identify unknown classes. The distribution alignment loss of DCC and OVANet is designed for both known and unknown classes. Therefore, applying our strategy to rewieghted the distribution alignment loss can improve the performance on both known and unknown classes. Moreover, since the loss functions of OVANet and DCC are based on different categories (see Eqs. (7.6)(7.8)), we can directly apply our strategy to reweight the corresponding loss functions without obtaining target pseudo-labels (see Eqs. (7.7)(7.9)).

Remark 2. Our proposed dynamic reweighted loss learning strategy can be applied to other domain adaptation scenarios, such as source-free domain adaptation [83]. We provide additional results in the supplementary material to verify this.

7.5 Experiments

Our experiments were designed to verify the efficacy of DC in UniDA scenarios. To this end we integrated the DC into both DCC [76] and OVANet [129] and compared their performance to the original variants on a range of different datasets.

7.5.1 Experimental Setup

Datasets. We used four benchmark datasets in domain adaptation areas for validation. **Office** [47] consists of approximately 4700 images in 31 categories from three domains: Amazon (A), DSLR (D), and Webcam (W). **OfficeHome** [147] includes 15500 images from 65 categories across four domains: artistic images (A), clip-art images (C), product images (P), and real-world images (R). **VisDA** [118] is a large-scale dataset with 12 categories. The source domain of VisDA contains about $150K$ synthetic images (S) and the target domain contains $50K$ real-world images (R). Additionally, we used the highly-challenging domain adaptation dataset **DomainNet** [117]. This dataset contains six domains, with each domain containing 345 categories of common objects. In our experiments, we used three domains to design our domain adaptation tasks, namely painting images (P), real images (R), and sketch images (S). We followed the standard protocols from [47, 129, 171] to split these datasets into a common label set $\mathscr{Y}_c$, the private label set of the source domain $\bar{\mathscr{Y}}_s$, and the private label set of the target domain $\bar{\mathscr{Y}}_t$. The notation ($\mathscr{Y}_c$/$\bar{\mathscr{Y}}_s$/$\bar{\mathscr{Y}}_t$) represents the split method used in each experiment. More details are shown in our supplementary material.

Evaluation Metric. Following [129], we used H-score [13] as the evaluation metric. An H-score can be calculated as follows:

$$\mathbf{H} = \frac{2acc_c \cdot acc_t}{acc_c + acc_t}, \tag{7.10}$$

where acc_c is the accuracy on the common label set $\mathscr{Y}_c$ and acc_t is the accuracy on

the private label set of the target domain (unknown) $\bar{\mathcal{Y}}_t$. However, H-score is not a suitable evaluation with a small number of unknown target samples. Thus, we have also reported other evaluation metrics in these cases, including **ACL**, measuring accuracy at recognizing known samples without rejection, and **UNK**, which denotes the accuracy of unknown sample identification.

Implementation. Following the original variants of DCC and OVANet, we used ResNet50 [31] pre-trained on ImageNet [59] as our backbone network. DCC and OVANet were implemented using the code provided in the original paper. DCC+DC and OVANet+DC denote the variants incorporating DC. The parameter settings of DCC and OVENet are same as the original paper. For DCC+DC, we set $T_0 = 500$ for Office and OfficeHome datasets, and $T_0 = 1000$ for VisDA and DomainNet datasets. For OVENet+DC, we set $T_0 = 500$ for Office, OfficeHome and VisDA datasets, and $T_0 = 5000$ for DomainNet datasets. We set $T_1 = 100$ for DCC+DC and OVENet+DC on all datasets. We used PyTorch 1.9.0 to implement these models. All experiments were conducted on an NVIDIA Quadro GV100 GPU with 32 GB memory. The learning rate was decayed with inverse learning rate decay scheduling [128].

7.5.2 Results Analysis

This section reports the results of our experiments coupled with a discussion on the results observed.

OPDA Setting. Tables 7.1 and 7.9 report the H-score comparison of DCC, OVANet, DCC+DC, and OVANet+DC. These results demonstrate that using DC improves performance in almost all domain adaptation tasks. In terms of the H-scores for the Office dataset, DCC+DC outperformed DCC by 3.4%, while OVANet+DC outperformed OVANet by 1.9%. With the OfficeHome dataset, DCC+DC outperformed DCC by 2.0%, and OVANet+DC outperformed OVANet by 1.5%. As for the large-scale VisDA dataset,

DCC+DC surpassed DCC by 5.4%, and OVANet+DC surpassed OVANet by 4.0%. Moreover, on the high challenge domain adaptation dataset, DomainNet, our proposed strategy achieved a near 1.0% improvement for both DCC and OVANet. All these outcomes verify the efficacy of DC for OPDA setting.

Tables 7.2, 7.3 and 7.4 present the comparison results of different evaluation metrics on all datasets in OPDA setting. DCC+DC and OVANet+DC outperform DCC and OVANet in terms of all three evaluation metrics across almost all tasks. This demonstrates that DC can both improve accuracy on the common label set and enhance the accuracy of unknown sample identification. Collectively, these results provide strong evidence for the effectiveness of our proposed strategy in addressing both domain shift and category shift.

OSDA Setting. From the results in Table 7.5, both DCC+DC and OVANet+DC achieved a better average H-score than DCC and OVANet on Office, OfficeHome, and VisDA datasets, especially for the VisDA dataset. More specifically, in terms of the H-score on the Office dataset, DCC+DC outperformed DCC by 0.5%, and OVANet+DC outperformed OVANet by 1.3%. With the OfficeHome dataset, DCC+DC outperformed DCC by 2.0%, and OVANet+DC outperformed OVANet by 1.0%. In the case of the large-scale VisDA dataset, DCC+DC surpassed DCC by 6.2%, and OVANet+DC surpassed OVANet by 5.3%. The supplementary material provides the comparisons between different evaluation metrics for all datasets in an OSDA setting. Similar to the OPDA setting, DCC+DC and OVANet+DC outperformed DCC and OVANet on all three evaluation metrics in almost all tasks. In summary, the above analysis further verifies the efficiency of DC in enhancing model performance in OSDA setting.

Table 7.1: H-score (%) comparison on Office (10/10/11), OfficeHome (10/5/50), and VisDA (6/3/3) for open-partial domain adaptation. The better results of DCC+DC (OVANet+DC) with the original version are highlighted in red for each column. In addition, we report the experiment outcomes of other SOTA baselines to verify the efficiency of DC.

	Office (10/10/11)							OfficeHome (10/5/50)												VisDA(6/3/3)
Methods	A2D	A2W	D2A	D2W	W2A	W2D	Avg	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg	S2R
UAN	59.7	58.6	60.1	70.6	60.3	71.4	63.5	51.6	51.7	54.3	61.7	57.6	61.9	50.4	47.6	61.5	62.9	52.6	65.2	56.6	30.5
CMU	68.1	67.3	71.4	79.3	72.2	80.4	73.1	56.0	56.9	59.2	67.0	64.3	67.8	54.7	51.1	66.4	68.2	57.9	69.7	61.6	34.6
DANCE	79.6	75.8	82.9	90.9	77.6	87.6	82.3	61.0	60.4	64.9	65.7	58.8	61.8	73.1	61.2	66.6	67.7	62.4	63.7	63.9	42.8
GATE	87.7	81.6	84.2	94.8	83.4	94.1	87.6	63.8	75.9	81.4	74.0	72.1	79.8	74.7	70.3	82.7	79.1	71.5	81.7	75.6	56.4
DCC	83.1	85.7	82.1	84.8	77.3	85.6	83.1	62.4	78.3	79.8	67.6	71.8	73.6	69.1	52.7	80.1	71.8	55.4	79.1	70.2	56.3
DCC+DC	86.4	88.6	81.5	88.9	82.4	91.0	86.5	62.1	79.8	83.0	69.1	71.9	78.5	69.4	56.9	82.7	73.4	57.9	82.0	72.2	61.7
OVANet	83.8	78.4	80.7	95.9	82.7	95.5	86.2	61.2	77.1	79.7	68.9	69.3	76.4	71.0	59.3	81.1	76.1	63.5	79.8	71.9	53.1
OVANet+DC	86.5	82.1	82.9	97.3	83.5	96.4	88.1	62.3	79.1	82.8	70.2	69.1	77.6	71.1	60.8	82.4	78.6	64.1	82.2	73.4	57.1

Table 7.2: Different evaluation metrics on Office (10/10/11) and VisDA (6/3/3) for open-partial domain adaptation. The best results are highlighted in red for each column.

	Office (10/10/11)																		VisDA (6/3/3)		
Methods	A2D			A2W			D2A			D2W			W2A			W2D			S2R		
Evaluation	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK
DCC	83.1	92.2	77.7	85.7	89.7	82.5	82.1	84.6	79.7	84.8	96.5	74.3	77.3	90.8	66.1	85.6	97.7	74.9	54.3	52.3	55.1
DCC+DC	86.4	93.5	81.1	88.6	90.1	88.5	81.5	88.5	71.9	88.9	95.1	82.5	82.4	86.4	78.4	91.0	98.5	83.4	61.7	65.1	74.2
OVANet	83.8	86.6	89.7	78.4	84.5	81.8	80.7	88.6	86.1	95.9	99.7	94.4	82.7	91.5	84.4	95.5	100.0	90.5	53.1	67.3	73.5
OVANet+DC	86.5	87.3	95.2	82.1	84.6	92.1	82.9	89.5	88.5	97.3	99.5	96.1	83.5	90.6	87.3	96.4	100.0	93.1	57.1	69.1	74.8

Table 7.3: Different evaluation metrics on OfficeHome (10/5/50) for open-partial domain adaptation. The best results are highlighted in red for each column.

	OfficeHome (10/5/50)																	
Methods	A2P			A2R			C2A			C2R			P2C			R2P		
Evaluation	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK
DCC	78.3	78.3	77.3	79.8	91.6	69.3	67.6	73.1	62.2	73.6	80.7	66.5	51.7	48.5	61.6	79.1	78.1	79.7
DCC+DC	79.8	84.2	75.1	83.0	91.8	76.9	69.1	75.5	66.2	78.4	76.5	80.0	56.8	51.4	73.4	82.0	81.0	82.3
OVENet	77.1	88.8	74.3	79.7	96.7	69.7	68.9	79.2	77.5	76.4	90.2	74.5	59.3	61.1	77.6	81.1	92.0	79.9
OVENet+DC	79.8	89.4	81.6	83.0	96.4	74.1	70.2	77.3	83.6	77.6	91.3	75.6	60.8	63.6	78.7	82.4	93.0	80.1

Table 7.4: Different evaluation metrics on DomainNet (150/50/145) for open-partial domain adaptation. The best results are highlighted in red for each column.

	DomainNet (150/50/145)																	
Methods	P2R			P2S			R2P			R2S			S2P			S2R		
Evaluation	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK
DCC	56.9	47.2	69.8	42.4	30.9	56.8	50.4	39.2	66.5	42.2	29.9	60.6	44.9	31.4	66.7	56.8	48.4	67.5
DCC+DC	59.6	54.2	66.7	42.7	31.2	60.4	51.2	40.2	68.4	42.5	32.5	61.7	47.6	34.5	62.9	57.8	50.3	67.8
OVENet	56.1	64.2	53.8	47.2	47.5	68.3	51.7	53.4	75.4	44.8	41.7	69.4	47.9	49.1	70.3	57.0	64.1	57.6
OVENet+DC	56.6	64.7	54.3	47.9	47.7	68.9	52.2	54.1	76.2	45.2	42.2	73.1	49.2	51.8	78.0	58.4	64.5	61.4

Table 7.5: H-score (%) comparison on Office (10/0/11), OfficeHome (25/0/40), and VisDA (6/0/6) for open-set domain adaptation. The better results of DCC+DC (OVANet+DC) with the original version are highlighted in red for each column. In addition, we report the experiment outcomes of other SOTA baselines to verify the efficiency of DC.

	Office (10/0/11)							OfficeHome (25/0/40)												VisDA(6/0/6)
Methods	A2D	A2W	D2A	D2W	W2A	W2D	Avg	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg	S2R
OSBP	82.4	82.7	75.1	97.2	73.7	91.1	83.7	55.1	65.2	72.9	64.3	64.7	70.6	63.2	53.2	73.9	66.7	54.5	72.3	64.7	52.3
UAN	38.9	46.8	68.0	68.8	54.9	53.0	55.1	40.3	41.5	46.1	53.2	48.0	53.7	40.6	39.8	52.5	53.6	43.7	56.9	47.5	51.9
CMU	52.6	55.7	76.5	75.9	65.8	64.9	65.2	45.1	48.3	51.7	58.9	55.4	61.2	46.5	43.8	58.0	58.6	50.1	61.8	53.3	54.2
DANCE	84.9	78.8	79.1	78.8	68.3	88.9	79.8	61.9	61.3	63.7	64.2	58.6	62.6	67.4	61.0	65.5	65.9	61.3	64.2	63.0	67.5
GATE	88.4	86.5	84.2	95.0	86.1	96.7	89.5	63.8	70.5	75.8	66.4	67.9	71.7	67.3	61.5	76.0	70.4	61.8	75.1	69.1	70.8
DCC	87.1	89.4	84.3	93.0	84.3	93.0	88.5	52.3	69.0	74.2	58.0	65.3	68.5	63.9	53.2	73.5	65.5	57.3	71.9	64.4	58.6
DCC+DC	88.3	89.4	85.1	93.0	85.2	93.1	89.0	54.4	70.3	76.3	58.7	67.6	73.8	64.2	52.9	75.4	68.4	57.6	73.0	66.1	64.8
OVANet	89.9	86.6	87.2	95.6	86.7	98.0	90.6	58.8	66.6	69.8	60.2	64.4	69.0	60.7	52.6	69.2	67.0	59.6	64.8	63.6	66.1
OVANet+DC	90.8	89.3	86.5	96.6	88.7	99.1	91.9	58.8	67.1	72.8	59.2	65.3	69.6	60.2	53.4	72.2	68.5	59.4	69.2	64.6	71.8

Table 7.6: Different evaluation metrics on Office (10/0/11) and VisDA (6/0/6) for open-set domain adaptation. The best results are highlighted in red for each column.

	Office (10/0/11)						OfficeHome (25/0/40)												VisDA (6/0/6)		
Methods	A2D			W2A			A2R			C2P			P2R			R2A			S2R		
Evaluation	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK	H	ACL	UNK
DCC	87.1	92.8	83.0	84.3	86.7	86.7	74.2	75.1	72.7	65.3	56.4	79.4	73.5	74.2	71.5	65.5	61.6	66.8	58.6	49.8	64.0
DCC+DC	88.3	92.6	85.6	85.2	89.9	80.3	76.3	75.2	76.5	67.6	63.5	71.4	75.4	76.2	73.1	68.4	64.8	69.0	64.8	70.3	64.1
OVENet	89.9	93.0	87.4	86.7	94.6	88.2	69.8	90.1	61.1	64.4	72.3	68.9	69.2	86.5	65.0	67.0	79.1	68.8	66.1	65.3	86.9
OVENet+DC	90.8	93.6	90.3	88.7	94.7	93.8	72.8	89.6	67.1	65.3	71.6	74.9	72.2	87.1	71.4	68.5	80.9	71.6	71.8	71.1	87.2

Table 7.7: Accuracy (%) comparison on Office (10/21/0), OfficeHome (25/40/0), and VisDA (6/6/0) for partial domain adaptation. The better results of DCC+DC (OVANet+DC) with the original version are highlighted in red for each column. In addition, we report the experiment outcomes of other SOTA baselines to verify the efficiency of DC.

	Office (10/21/0)							OfficeHome (25/40/0)												VisDA(6/6/0)
Methods	A2D	A2W	D2A	D2W	W2A	W2D	Avg	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg	S2R
ETN	95.0	94.5	96.2	100.0	94.6	100.0	96.7	59.2	77.0	79.5	62.9	65.7	75.0	68.3	55.4	84.4	75.7	57.7	84.5	70.5	59.8
UAN	79.7	76.8	82.7	93.4	83.7	98.3	85.8	24.5	35.0	41.5	34.7	32.3	32.7	32.7	21.2	43.0	39.7	26.6	46.0	34.2	39.7
CMU	84.1	84.2	69.2	97.2	66.8	98.8	83.4	50.9	74.2	78.4	62.2	64.1	72.5	63.5	47.9	78.3	72.4	54.7	78.9	66.5	65.5
DANCE	77.1	71.2	83.7	93.6	92.6	96.8	86.0	53.6	73.2	84.9	70.8	67.3	82.6	70.0	50.9	84.8	77.0	55.9	81.8	71.1	73.7
GATE	89.5	86.2	93.5	100.0	94.4	98.6	93.7	55.8	75.9	85.3	73.6	70.2	83.0	72.1	59.5	84.7	79.6	63.9	83.8	73.9	75.6
DCC	90.7	91.3	95.3	100.0	94.3	100.0	95.3	48.0	73.6	83.0	44.6	68.8	72.3	59.9	47.5	79.9	85.5	78.2	82.6	68.7	75.9
DCC+DC	93.8	93.9	96.3	100.0	96.1	100.0	96.7	52.6	79.7	84.3	45.7	68.2	74.1	62.1	46.3	81.9	86.2	79.1	82.9	70.3	76.7
OVANet	69.4	61.7	61.4	90.2	66.4	98.7	74.6	34.1	54.6	72.1	42.4	47.3	55.9	38.2	26.2	61.7	56.7	35.8	68.9	49.5	34.3
OVANet+DC	73.3	66.4	72.0	97.3	74.5	100.0	80.6	37.3	58.7	74.8	45.5	49.6	58.4	42.3	29.3	65.2	60.9	39.4	71.2	52.7	36.7

Table 7.8: Accuracy (%) comparison on Office (31/0/0), OfficeHome (65/0/0), and VisDA (12/0/0) for closed-se domain adaptation. The better results of DCC+DC (OVANet+DC) with the original version are highlighted in red for each column. In addition, we report the experiment outcomes of other SOTA baselines to verify the efficiency of DC.

	Office (31/0/0)							OfficeHome (65/0/0)												VisDA(12/0/0)
Methods	A2D	A2W	D2A	D2W	W2A	W2D	Avg	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg	S2R
MDD	93.5	94.5	74.6	98.4	72.2	100.0	88.9	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1	74.6
UAN	97.0	86.5	69.6	100.0	68.7	84.5	84.4	45.0	63.6	71.2	51.4	58.2	63.2	52.6	40.9	71.0	63.3	48.2	75.4	58.7	66.4
CMU	78.3	79.6	62.3	98.1	63.4	97.6	79.9	42.8	65.6	74.3	58.1	63.1	67.4	54.2	41.2	73.8	66.9	48.0	78.7	61.2	56.9
DANCE	89.4	88.6	69.5	97.5	68.2	100.0	85.4	54.3	75.9	78.4	64.8	72.1	73.4	63.2	53.0	79.4	73.0	58.2	82.9	69.1	70.2
GATE	91.3	90.5	73.4	98.7	75.9	100.0	88.3	54.6	76.9	79.8	66.1	73.5	74.2	65.3	54.8	80.6	73.9	59.5	83.7	70.2	74.8
DCC	85.8	81.5	65.9	97.2	68.9	100.0	83.2	35.4	66.0	73.2	46.0	51.3	62.7	47.8	29.8	68.9	57.8	41.0	79.2	54.6	67.1
DCC+DC	89.2	83.2	72.5	98.3	69.9	100.0	85.5	39.2	65.9	74.8	49.8	59.2	63.8	48.3	31.1	70.3	60.2	44.5	81.2	57.4	72.2
OVANet	72.5	67.3	43.4	94.8	44.9	99.6	70.4	34.5	55.8	67.1	40.9	52.8	56.9	35.4	26.2	61.8	53.8	35.4	70.8	49.3	36.2
OVANet+DC	74.5	69.6	47.6	95.2	48.6	99.7	72.5	37.2	59.7	69.6	43.5	55.0	58.9	38.8	29.1	64.5	55.0	37.5	72.0	51.7	37.9

Table 7.9: H-score (%) comparison on DomainNet (150/50/145) for open-partial domain adaptation. The better results of DCC+DC (OVANet+DC) with the original version are highlighted in red for each column. In addition, we report the experiment outcomes of other SOTA baselines to verify the efficiency of DC.

	DomainNet (150/50/145)						
Methods	P2R	P2S	R2P	R2S	S2P	S2R	Avg
UAN	41.9	39.1	43.6	38.7	39.0	43.7	41.0
CMU	50.8	45.1	52.2	45.6	44.8	51.0	48.3
DANCE	55.7	47.0	51.1	46.4	47.9	55.7	50.6
GATE	57.4	48.7	52.8	47.6	49.5	56.3	52.1
DCC	56.9	42.4	50.4	42.2	44.9	56.8	48.9
DCC+DC	59.6	45.1	51.2	43.8	47.6	57.8	50.2
OVANet	56.1	47.2	44.8	51.7	47.9	57.0	50.8
OVANet+DC	56.6	47.9	45.2	52.2	49.2	58.4	51.6

Table 7.6 present the comparison results of different evaluation metrics in OSDA setting. DCC+DC and OVANet+DC outperform DCC and OVANet in terms of all three evaluation metrics across almost all tasks. This again demonstrates that DC can both improve accuracy on the common label set and enhance the accuracy of unknown sample identification.

PDA setting. Table 7.7 reports the accuracy comparison of DCC+DC and OVANet+DC with their original versions in PDA setting. In terms of the classification accuracy for the Office dataset, DCC+DC outperformed DCC by 1.4%, while OVANet+DC outperformed OVANet by 6.0%. With the OfficeHome dataset, DCC+DC outperformed DCC by 1.6%, and OVANet+DC outperformed OVANet by 3.2%. As for the large-scale VisDA dataset, DCC+DC surpassed DCC by 0.8%, and OVANet+DC surpassed OVANet by 2.4%. These outcomes verify the efficiency of DC in PDA setting.

CDA setting. Table 7.8 reports the accuracy comparison of DCC+DC and OVANet+DC with their original versions in CDA setting. In terms of the classification accuracy for the Office dataset, DCC+DC outperformed DCC by 2.3%, while OVANet+DC outperformed OVANet by 2.1%. With the OfficeHome dataset, DCC+DC outperformed DCC by 2.8%, and OVANet+DC outperformed OVANet by 2.4%. As for the large-scale VisDA dataset,

Table 7.10: Accuracy (%) comparison on Office. The best results are highlighted in red for each column.

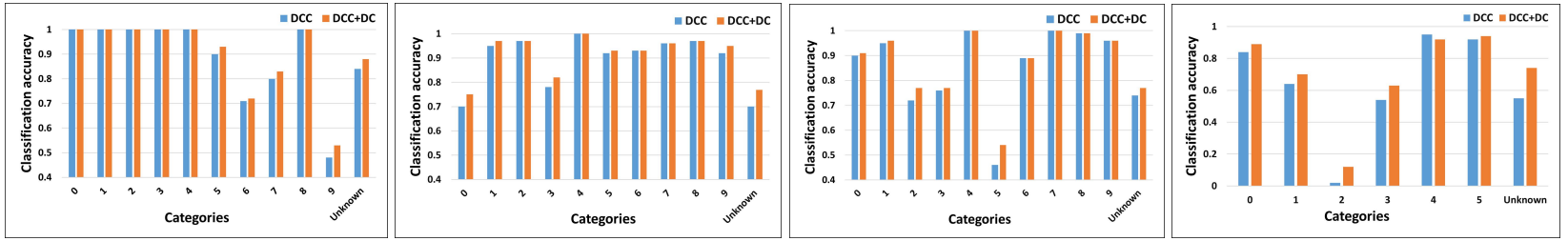
	Office (31/0/0)						
Methods	A2D	A2W	D2A	D2W	W2A	W2D	Avg
SHOT	94.1	90.3	73.7	98.7	73.4	100.0	88.4
SHOT+DC	94.4	91.1	74.1	99.2	74.3	100.0	88.8

DCC+DC surpassed DCC by 5.1%, and OVANet+DC surpassed OVANet by 1.7%.

In addition, with our proposed strategy, DCC and OVENet achieve comparable results to GATE [21], which is the latest UniDA method, on almost all datasets and all settings. All these results verify the efficiency of our proposed dynamic reweighted loss learning strategy for UniDA problem in all four settings.

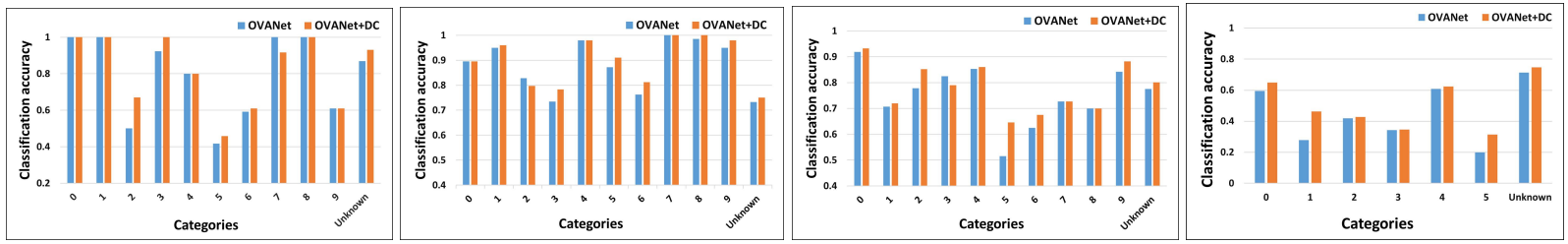
Finally, we apply DC to one source-free domain adaptation [83] method to verify that our proposed dynamic reweighted loss learning strategy can be applied to other domain adaptation scenarios. The experiment results are shown in Table 7.10, which illustrates that the performance has improved by applying our proposed dynamic reweighted loss learning strategy.

Further Analysis. To further demonstrate that applying DC can improve the performance on the hard transfer categories, Figures 7.3 and 7.4 depict the classification accuracy for each class in an OPDA setting. As shown in Figures 7.3 and 7.4, by applying our proposed strategy, the performance of the original UniDA methods DCC and OVANet is enhanced not only on the common classes but also on the “unknown" classes. For example, let $L = 0.2$ in Definition 7.1, i.e., we define categories with an accuracy rate less than 80% as hard transfer categories. From Figure 7.3 (d), DCC with DC achieves a significant accuracy improvement on hard transfer categories $(1, 2, 3,$ “unknown") compared with DCC. From Figure 7.4 (c), compared with OVANet, the classification accuracy of OVANet with DC on nearly all categories has been improved to varying degrees, especially on hard transfer categories $1, 2, 5, 6,$ “unknown". Overall, the above analysis again provides strong evidence that DC can effectively mitigate domain shift and category shift.



(a) **Office : A → W** (b) **OfficeHome : A → R** (c) **OfficeHome : P → R** (d) **VisDA**

Figure 7.3: **Classification accuracy of DCC and DCC+DC for each class in open-partial domain adaptation. Blue:** DCC. **Red:** DCC with Distraction-Control. These are histograms of the classification accuracy for each class on Office (10/10/11), OfficeHome (10/5/50), and VisDA (6/3/3). From all sub-figures, DCC with DC achieves classification accuracy improvement on almost all categories compare with DCC, especially on hard transfer categories.



(a) **Office : A → D** (b) **OfficeHome : P → R** (c) **OfficeHome : R → A** (d) **VisDA**

Figure 7.4: **Classification accuracy of OVANet and OVANet+DC for each class in open-partial domain adaptation. Blue:** OVANet. **Red:** OVANet with DC. These are histograms of the classification accuracy for each class on Office (10/10/11), OfficeHome (10/5/50), and VisDA (6/3/3). From all sub-figures, OVANet with DC achieves classification accuracy improvement on almost all categories compare with OVANet, especially on hard transfer categories.

Table 7.11: Accuracy (%) comparison on Office in PDA setting. The best results are highlighted in red for each column.

	Office (10/21/0)						
Methods	A2D	A2W	D2A	D2W	W2A	W2D	Avg
PADA	82.2	86.5	92.7	99.3	95.4	100.0	92.7
ARPDA	96.8	93.5	95.5	100.0	96.0	99.7	96.9
DCC+DC	93.8	93.9	96.3	100.0	96.1	100.0	96.7

7.5.3 Compare with Other Reweighted Learning Strategies

This section compare DC with other reweighted learning strategies in domain adaptation. The experimental results are shown in Table 7.11. We can find that DCC+DC achieves better performance than PADA [15] and gets comparable results to ARPDA [54]. Moreover, PADA and ARPDA are only good at dealing with PDA setting, while our proposed strategy can be applied to all domain adaptation settings, especially for OPDA and OSDA settings, which verifies the superiority of our proposed strategy.

7.6 Summary

In this chapter, we presented a novel dynamic reweighted loss learning strategy, called Distraction-control, for enhancing UniDA performance with hard transfer categories. The strategy prompts a trained model to allocate more effort to the hard transfer categories, thereby improving the overall distribution alignment between the source and target domains. Through extensive experiments on four well-known real-world domain adaptation datasets, the results show that applying DC to some SOTA UniDA models can significantly enhance the ability to address domain shift and category shift problems.

CHAPTER 8

CONCLUSIONS AND FUTURE STUDY

This chapter concludes the entire thesis and provides some further research directions.

8.1 Conclusions

Recent research in transfer learning has achieved significant advancements in many areas, including medical imaging, natural language processing, and computer vision. However, most existing work predominantly focuses on addressing large-scale image data characterized by crisp values, while imprecise data, a common type in real-world scenarios, has been severely neglected. To fill this gap, this thesis focuses on solving transfer learning with imprecise observations in different scenarios. There are four main challenges addressed in this work: i) constructing a theoretical basis for imprecise data analysis; ii) transferring knowledge when both source and target domains contain only imprecise data in a single-source scenario; iii) transferring knowledge across multiple source domains with crisp-valued data and a target domain with imprecise data; and iv) transferring knowledge when the source or target domain has imprecise data in a

UniDA scenario.

To solve the above-mentioned challenges, this thesis proposed four concrete research questions and corresponding research objectives. The findings of this study are summarized as follows:

1. The theoretical foundation is built for analyzing imprecise data, and we propose a new framework to address MCIMO problem. (to achieve RO 1)

2. We provide the theoretical analysis for the problem of learning from interval-valued data and develop a new algorithm to solve this problem. Then, this algorithm is applied to build a new framework for protecting data privacy. (to achieve RO 1)

3. We formalize the domain adaptation with imprecise observations problem and derive the theoretical bound of this problem. Based on the T-S fuzzy rule-based model, a new designed integral probability metric, and a deep clustering-based self-supervised pseudo-labeling strategy, we develop a new model to efficiently address this more realistic and challenging problem. (to achieve RO 2)

4. We extend the domain adaptation with imprecise observations problem into a more challenging scenario: multi-source domain adaptation scenario, where we aim to enhance the prediction performance on interval-valued target data by leveraging the knowledge derived from multiple source data with crisp-valued features. Two fuzzy technique-based frameworks are constructed to effectively address this problem. (to achieve RO 3)

5. We identify an unresolved issue in UniDA that the existing UniDA methods can not achieving satisfactory adaptation performance in the hard transfer categories. Then, we provide a theory to gain insights and propose a novel dynamic reweighted loss learning strategy to tackle this issue. (to achieve RO 4)

In addition, according to the achievements of this thesis, we get some significant insights of the advantages and limitations when combining fuzzy techniques with traditional machine learning algorithms. Here is a summary:

Advantages:

1. **Handling Uncertainty and Ambiguity:** Fuzzy techniques excel at handling uncertainty and imprecision, making them suitable for real-world problems where data may be imprecise or incomplete. Traditional machine learning algorithms typically assume precise and well-defined inputs; therefore, integrating fuzzy logic can enhance their robustness in uncertain environments.

2. **Improved Interpretability:** Fuzzy systems often provide more interpretable models through linguistic rules and membership functions. Traditional machine learning models, particularly complex ones like deep learning, can function as black-boxes. Combining fuzzy systems with traditional machine learning can yield models that are both powerful and interpretable.

3. **Enhanced Flexibility:** Fuzzy techniques can model complex, non-linear relationships using simple rule-based systems. Traditional machine learning algorithms can benefit from this flexibility, enabling them to better capture and model intricate patterns in data.

4. **Improved Generalization:** Fuzzy techniques can help create more generalized models that perform better on unseen data by incorporating the notion of partial truths and handling variations in data.

Limitations:

1. **Increased Complexity:** Integrating fuzzy techniques with traditional machine learning can result in more complex models. Consequently, computational over-

head may increase due to the additional complexity of processing fuzzy rules and membership functions.

2. **Scalability Issues:** Fuzzy systems may struggle with scalability, particularly when the number of rules and membership functions becomes large. Traditional machine learning algorithms are typically designed to handle large-scale data efficiently, and incorporating fuzzy components might hinder their scalability.

3. **Parameter Tuning:** Fuzzy systems require careful tuning of membership functions and rules, which can be a challenging and time-consuming task.

8.2 Future Study

This thesis identifies the following directions as future work:

1. *Heterogeneous transfer learning with imprecise data*

   Heterogeneous transfer learning addresses the challenge of transferring knowledge between domains with different feature spaces. However, most recent works in heterogeneous transfer learning primarily focus on crisp-valued data. In the future, we aim to develop and enhance methods for heterogeneous transfer learning where data from the source or target domain is imprecise.

2. *Consider other type of imprecise data*

   In this thesis, we decide to use fuzzy vectors to represent imprecise data, particularly interval-valued data. However, in real-world scenarios, there are various other types of imprecise data. In the future, we will explore methods to manage different types of imprecise data, such as noisy data and missing values. This work aims to develop comprehensive strategies for preprocessing and managing imprecise

data to maximize its utility in machine learning models, thereby improving overall model robustness and accuracy.

3. *OOD detection and generalization with imprecise data*

   Out-of-Distribution (OOD) detection and generalization are critical areas in machine learning that focus on identifying when a model encounters data that differs significantly from its training set and ensuring that models perform well even when faced with such unfamiliar data. Similarly, existing research on OOD detection and generalization often overlooks the issues posed by imprecise data. In the future, we aim to develop innovative strategies to train models that can generalize effectively to new, unseen domains with imprecise data while minimizing reliance on large source datasets.

4. *OOD detection and generalization with limited source data or without source data*

   Most existing methods still rely on utilizing large amounts of source data to enhance model generalization capabilities while effectively detecting unknown categories. To alleviate this constraint, in future research, we will propose a novel method that simultaneously addresses covariate shifts and category shifts in OOD generalization and detection with limited source data or without source data.

BIBLIOGRAPHY

[1] E. L. ALLWEIN, R. E. SCHAPIRE, AND Y. SINGER, *Reducing multiclass to binary: A unifying approach for margin classifiers*, Journal of Machine Learning Research, 1 (2000).

[2] P. P. ANGELOV AND D. P. FILEV, *An approach to online identification of takagi-sugeno fuzzy models*, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 34 (2004), pp. 484–498.

[3] R. J. AUMANN, *Integrals of set-valued functions*, Journal of mathematical analysis and applications, 12 (1965), pp. 1–12.

[4] F. R. BACH, G. R. LANCKRIET, AND M. I. JORDAN, *Multiple kernel learning, conic duality, and the smo algorithm*, in Proceedings of the 21st international conference on Machine learning, 2004, p. 6.

[5] V. BEHBOOD, J. LU, G. ZHANG, AND W. PEDRYCZ, *Multistep fuzzy bridged refinement domain adaptation algorithm and its application to bank failure prediction*, IEEE Transactions on Fuzzy Systems, 23 (2015), pp. 1917–1935.

[6] S. BEN-DAVID, J. BLITZER, K. CRAMMER, A. KULESZA, F. PEREIRA, AND J. W. VAUGHAN, *A theory of learning from different domains*, Machine learning, 79 (2010), pp. 151–175.

[7] S. BICKEL, M. BRÜCKNER, AND T. SCHEFFER, *Discriminative learning for differing training and test distributions*, in Proceedings of the 24th International Conference on Machine Learning, 2007, pp. 81–88.

[8] L. BILLARD AND E. DIDAY, *From the statistics of data to the statistics of knowledge: symbolic data analysis*, Journal of the American Statistical Association, 98 (2003), pp. 470–487.

[9] A. BLUM AND T. MITCHELL, *Combining labeled and unlabeled data with co-training*, in Proceedings of the 11th annual conference on Computational learning theory, 1998, pp. 92–100.

[10] L. BORZEMSKI AND G. STARCZEWSKI, *Application of transfer regression to tcp throughput prediction*, in Proceedings of the 2009 First Asian Conference on Intelligent Information and Database Systems, 2009, pp. 28–33.

[11] L. BOTTOU, *Stochastic gradient descent tricks*, in Neural Networks: Tricks of the Trade: Second Edition, Springer, 2012, pp. 421–436.

[12] K. H. BRODERSEN, C. S. ONG, K. E. STEPHAN, AND J. M. BUHMANN, *The balanced accuracy and its posterior distribution*, in Proceedings of the 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 3121–3124.

[13] S. BUCCI, M. R. LOGHMANI, AND T. TOMMASI, *On the effectiveness of image rotation for open set domain adaptation*, in Proceedings of the European Conference on Computer Vision, 2020, pp. 422–438.

[14] G. CAI, L. HE, M. ZHOU, H. ALHUMADE, AND D. HU, *Learning smooth representation for unsupervised domain adaptation*, IEEE Transactions on Neural Networks and Learning Systems, 34 (2021), pp. 4181–4195.

[15] Z. CAO, L. MA, M. LONG, AND J. WANG, *Partial adversarial domain adaptation*, in Proceedings of the 15th European Conference on Computer Vision, 2018, pp. 135–150.

[16] M. CARON, P. BOJANOWSKI, A. JOULIN, AND M. DOUZE, *Deep clustering for unsupervised learning of visual features*, in Proceedings of the 14th European conference on computer vision, 2018, pp. 132–149.

[17] W. CHANG, Y. SHI, H. D. TUAN, AND J. WANG, *Unified optimal transport framework for universal domain adaptation*, in Proceedings of the 36th International Conference on Neural Information Processing Systems, 2022, pp. 29512–29524.

[18] CHAPELLE, O., HAFFNER, P., VAPNIK, AND N. V., *Support vector machines for histogram-based image classification*, IEEE Transactions on Neural Networks, 10 (1999), pp. 1055–1064.

[19] L. CHEN, H. CHEN, Z. WEI, X. JIN, X. TAN, Y. JIN, AND E. CHEN, *Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 7181–7190.

[20] L. CHEN, Y. LOU, J. HE, T. BAI, AND M. DENG, *Evidential neighborhood contrastive learning for universal domain adaptation*, in Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022, pp. 6258–6267.

[21] L. CHEN, Y. LOU, J. HE, T. BAI, AND M. DENG, *Geometric anchor correspondence mining with uncertainty modeling for universal domain adaptation*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16134–16143.

[22] A. COLUBI, G. GONZÁLEZ-RODRÍGUEZ, M. Á. GIL, AND W. TRUTSCHNIG, *Non-parametric criteria for supervised classification of fuzzy data*, International journal of approximate reasoning, 52 (2011), pp. 1272–1282.

[23] C. CORTES, M. MOHRI, AND A. ROSTAMIZADEH, *Generalization bounds for learning kernels*, in Proceedings of the 27th International Conference on International Conference on Machine Learning, 2010, pp. 247–254.

[24] T. COUR, B. SAPP, AND B. TASKAR, *Learning from partial labels*, The Journal of Machine Learning Research, 12 (2011), pp. 1501–1536.

[25] E. CZOGALA, J. DREWNIAK, AND W. PEDRYCZ, *Fuzzy relation equations on a finite set*, Fuzzy Sets and systems, 7 (1982), pp. 89–101.

[26] W. DAI, Q. YANG, G.-R. XUE, AND Y. YU, *Self-taught clustering*, in Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 200–207.

[27] A. DANIELY AND S. SHALEV-SHWARTZ, *Optimal learners for multiclass problems*, in Proceedings of The 27th Conference on Learning Theory, 2014, pp. 287–316.

[28] F. D. A. DE CARVALHO AND Y. LECHEVALLIER, *Dynamic clustering of interval-valued data based on adaptive quadratic distances*, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 39 (2009), pp. 1295–1306.

[29] F. D. A. DE CARVALHO AND C. P. TENÓRIO, *Fuzzy k-means clustering algorithms for interval-valued data based on adaptive quadratic distances*, Fuzzy Sets and Systems, 161 (2010), pp. 2978–2999.

[30] M. DELGADO, M. A. VILA, AND W. VOXMAN, *On a canonical representation of fuzzy numbers*, Fuzzy Sets and Systems, 93 (1998), pp. 125–135.

[31] J. DENG, W. DONG, R. SOCHER, L.-J. LI, K. LI, AND L. FEI-FEI, *Imagenet: A large-scale hierarchical image database*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[32] Z. DENG, Y. JIANG, K.-S. CHOI, F.-L. CHUNG, AND S. WANG, *Knowledge-leverage-based TSK fuzzy system modeling*, IEEE Transactions on Neural Networks and Learning Systems, 24 (2013), pp. 1200–1212.

[33] Z. DENG, Y. JIANG, F.-L. CHUNG, H. ISHIBUCHI, AND S. WANG, *Knowledge-leverage-based fuzzy system and its modeling*, IEEE Transactions on Fuzzy Systems, 21 (2013), pp. 597–609.

[34] Z. DENG, Y. JIANG, H. ISHIBUCHI, K.-S. CHOI, AND S. WANG, *Enhanced knowledge-leverage-based TSK fuzzy system modeling for inductive transfer learning*, ACM Transactions on Intelligent Systems and Technology, 8 (2016), pp. 1–21.

[35] Z. DENG, P. XU, L. XIE, K.-S. CHOI, AND S. WANG, *Transductive joint-knowledge-transfer TSK FS for recognition of epileptic EEG signals*, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 26 (2018), pp. 1481–1494.

[36] T. G. DIETTERICH, P. DOMINGOS, L. GETOOR, S. MUGGLETON, AND P. TADEPALLI, *Structured machine learning: the next ten years*, Machine Learning, 73 (2008), p. 3.

[37] N. DING, Y. XU, Y. TANG, C. XU, Y. WANG, AND D. TAO, *Source-free domain adaptation via distribution estimation*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 7212–7222.

[38] J. DOMBI, *Membership function as an evaluation*, Fuzzy Sets and Systems, 35 (1990), pp. 1–21.

[39] J. DONG, Z. FANG, A. LIU, G. SUN, AND T. LIU, *Confident-anchor-induced multi-source-free domain adaptation*, in Proceedings of the 35th International Conference on Neural Information Processing Systems, 2021, pp. 2848–2860.

[40] L. DUAN, I. W. TSANG, AND D. XU, *Domain transfer multiple kernel learning*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34 (2012), pp. 465–479.

[41] D. DUBOIS AND H. PRADE, *Possibility theory: an approach to computerized processing of uncertainty*, Springer Science & Business Media, 2012.

[42] D. J. DUBOIS, *Fuzzy sets and systems: theory and applications*, vol. 144, Academic press, 1980.

[43] C. DWORK, A. ROTH, ET AL., *The algorithmic foundations of differential privacy*, Foundations and Trends® in Theoretical Computer Science, 9 (2014), pp. 211–407.

[44] Z. FANG, J. LU, F. LIU, J. XUAN, AND G. ZHANG, *Open set domain adaptation: Theoretical bound and algorithm*, IEEE transactions on neural networks and learning systems, 32 (2020), pp. 4309–4322.

[45] Z. FANG, J. LU, F. LIU, AND G. ZHANG, *Semi-supervised heterogeneous domain adaptation: Theory and algorithms*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 45 (2022), pp. 1087–1105.

[46] S. FRANCIS, J. VAN LANDEGHEM, AND M.-F. MOENS, *Transfer learning for named entity recognition in financial and biomedical documents*, Information, 10 (2019), p. 248.

[47] B. FU, Z. CAO, M. LONG, AND J. WANG, *Learning to detect open classes for universal domain adaptation*, in Proceedings of the 16th European Conference on Computer Vision, Springer, 2020, pp. 567–583.

[48] Y. GANIN AND V. LEMPITSKY, *Unsupervised domain adaptation by backpropagation*, in Proceedings of the 32nd International Conference on Machine Learning, 2015, pp. 1180–1189.

[49] R. GARGEES, J. M. KELLER, AND M. POPESCU, *Tlpcm: Transfer learning possibilistic c-means*, IEEE Transactions on Fuzzy Systems, 29 (2020), pp. 940–952.

[50] P. GERMAIN, A. HABRARD, F. LAVIOLETTE, AND E. MORVANT, *A pac-bayesian approach for domain adaptation with specialization to linear classifiers*, in Proceedings of the 30th International Conference on Machine Learning, PMLR, 2013, pp. 738–746.

[51] P. GERMAIN, A. HABRARD, F. LAVIOLETTE, AND E. MORVANT, *A new pac-bayesian perspective on domain adaptation*, in Proceedings of the 33rd International conference on machine learning, PMLR, 2016, pp. 859–868.

[52] R. GILAD-BACHRACH, N. DOWLIN, K. LAINE, K. LAUTER, M. NAEHRIG, AND J. WERNSING, *Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy*, in Proceedings of The 33rd International Conference on Machine Learning, PMLR, 2016, pp. 201–210.

[53] A. GRETTON, K. M. BORGWARDT, M. J. RASCH, B. SCHÖLKOPF, AND A. SMOLA, *A kernel two-sample test*, The Journal of Machine Learning Research, 13 (2012), pp. 723–773.

[54] X. GU, X. YU, Y. YANG, J. SUN, AND Z. XU, *Adversarial reweighting for partial domain adaptation*, in Proceedings of the 35th International Conference on Neural Information Processing Systems, 2021, pp. 14860–14872.

[55] H. HAN, Z. TANG, X. WU, H. YANG, AND J. QIAO, *Robust modeling for industrial process based on frequency reconstructed fuzzy neural network*, IEEE Transactions on Fuzzy Systems, (2023).

[56] P.-Y. HAO, *Interval regression analysis using support vector networks*, Fuzzy Sets and Systems, 160 (2009), pp. 2466–2485.

[57] M. HARDT, B. RECHT, AND Y. SINGER, *Train faster, generalize better: Stability of stochastic gradient descent*, in Proceedings of The 33rd International Conference on Machine Learning, 2016, pp. 1225–1234.

[58] T. HASTIE, R. TIBSHIRANI, J. H. FRIEDMAN, AND J. H. FRIEDMAN, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2, Springer, 2009.

[59] K. HE, X. ZHANG, S. REN, AND J. SUN, *Deep residual learning for image recognition*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[60] H. JEFFREYS, *The theory of probability*, OUP Oxford, 1998.

[61] G. JEONG AND H. Y. KIM, *Improving financial trading decisions using deep q-learning: Predicting the number of shares, action strategies, and transfer learning*, Expert Systems with Applications, 117 (2019), pp. 125–138.

[62] Y. JIANG, X. GU, D. JI, P. QIAN, J. XUE, Y. ZHANG, J. ZHU, K. XIA, AND S. WANG, *Smart diagnosis: A multiple-source transfer TSK fuzzy system for EEG seizure*

identification, ACM Transactions on Multimedia Computing, Communications, and Applications, 16 (2020), pp. 1–21.

[63] Y. JIANG, D. WU, Z. DENG, P. QIAN, J. WANG, G. WANG, F.-L. CHUNG, K.-S. CHOI, AND S. WANG, *Seizure classification from EEG signals using transfer learning, semi-supervised learning and TSK fuzzy system*, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 25 (2017), pp. 2270–2284.

[64] Y. JIANG, Y. ZHANG, C. LIN, D. WU, AND C.-T. LIN, *EEG-based driver drowsiness estimation using an online multi-view and transfer TSK fuzzy system*, IEEE Transactions on Intelligent Transportation Systems, (2021).

[65] M. I. JORDAN AND T. M. MITCHELL, *Machine learning: Trends, perspectives, and prospects*, Science, 349 (2015), pp. 255–260.

[66] Q. KANG, S. YAO, M. ZHOU, K. ZHANG, AND A. ABUSORRAH, *Effective visual domain adaptation via generative adversarial distribution matching*, IEEE Transactions on neural networks and learning systems, 32 (2020), pp. 3919–3929.

[67] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, in Proceedings of the International Conference on Learning Representations, 2015.

[68] G. KLIR AND B. YUAN, *Fuzzy sets and fuzzy logic*, vol. 4, Prentice hall New Jersey, 1995.

[69] M. KLOFT, U. BREFELD, S. SONNENBURG, AND A. ZIEN, *l_p-norm multiple kernel learning*, The Journal of Machine Learning Research, 12 (2011), pp. 953–997.

[70] V. KOLTCHINSKII AND D. PANCHENKO, *Empirical margin distributions and bounding the generalization error of combined classifiers*, Annals of Statistics, 30 (2002), pp. 1–50.

[71] S. KUMAR, A. K. SHUKLA, P. K. MUHURI, AND Q. D. LOHANI, *Transfer learning based GDP prediction from uncertain carbon emission data*, in Proceedings of the 2019 IEEE International Conference on Fuzzy Systems, 2019, pp. 1–6.

[72] J. N. KUNDU, N. VENKAT, R. V. BABU, ET AL., *Universal source-free domain adaptation*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4544–4553.

[73] H. KWAKERNAAK, *Fuzzy random variables - I. Definitions and theorems*, Information Sciences, 15 (1978), pp. 1–29.

[74] F. LAST, G. DOUZAS, AND F. BACAO, *Oversampling for imbalanced learning based on k-means and smote*, arXiv preprint arXiv:1711.00837, (2017).

[75] T. LEI, P. LIU, X. JIA, X. ZHANG, H. MENG, AND A. K. NANDI, *Automatic fuzzy clustering framework for image segmentation*, IEEE Transactions on Fuzzy Systems, 28 (2019), pp. 2078–2092.

[76] G. LI, G. KANG, Y. ZHU, Y. WEI, AND Y. YANG, *Domain consensus clustering for universal domain adaptation*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9757–9766.

[77] J. LI, E. CHEN, Z. DING, L. ZHU, K. LU, AND H. SHEN, *Maximum density divergence for domain adaptation.*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 43 (2021), pp. 3918–3930.

[78] J. LI, Y. LIU, R. YIN, H. ZHANG, L. DING, AND W. WANG, *Multi-class learning: from theory to algorithm*, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 1593–1602.

[79] K. LI, J. LU, H. ZUO, AND G. ZHANG, *Dynamic classifier alignment for unsupervised multi-source domain adaptation*, IEEE Transactions on Knowledge and Data Engineering, 35 (2022), pp. 4727–4740.

[80] K. LI, J. LU, H. ZUO, AND G. ZHANG, *Source-free multidomain adaptation with fuzzy rule-based deep neural networks*, IEEE Transactions on Fuzzy Systems, 31 (2023), pp. 4180–4194.

[81] R. LI, Q. JIAO, W. CAO, H.-S. WONG, AND S. WU, *Model adaptation: Unsupervised domain adaptation without source data*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 9641–9650.

[82] W. LI, H. ZHOU, W. XU, X.-Z. WANG, AND W. PEDRYCZ, *Interval dominance-based feature selection for interval-valued ordered data*, IEEE Transactions on Neural Networks and Learning Systems, 34 (2022), pp. 6898–6912.

[83] J. LIANG, D. HU, AND J. FENG, *Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation*, in Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 6028–6039.

[84] C. LIN, S. ZHAO, L. MENG, AND T.-S. CHUA, *Multi-source domain adaptation for visual sentiment classification*, in Proceedings of the 34th AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 2661–2668.

[85] T. Y. LIN AND N. CERCONE, *Rough sets and data mining: Analysis of imprecise data*, Springer Science & Business Media, 2012.

[86] F. LIU, J. LU, B. HAN, G. NIU, G. ZHANG, AND M. SUGIYAMA, *Butterfly: One-step approach towards wildly unsupervised domain adaptation*, arXiv preprint arXiv:1905.07720, (2020).

[87] F. LIU, J. LU, AND G. ZHANG, *Unsupervised heterogeneous domain adaptation via shared fuzzy equivalence relations*, IEEE Transactions on Fuzzy Systems, 26 (2018), pp. 3555–3568.

[88] F. LIU, G. ZHANG, AND J. LU, *A novel fuzzy neural network for unsupervised domain adaptation in heterogeneous scenarios*, in Proceedings of the 2019 IEEE International Conference on Fuzzy Systems, 2019, pp. 1–6.

[89] F. LIU, G. ZHANG, AND J. LU, *Heterogeneous domain adaptation: An unsupervised approach*, IEEE transactions on neural networks and learning systems, 31 (2020), pp. 5588–5602.

[90] F. LIU, G. ZHANG, AND J. LU, *Multisource heterogeneous unsupervised domain adaptation via fuzzy relation neural networks*, IEEE Transactions on Fuzzy Systems, 29 (2020), pp. 3308–3322.

[91] L. LIU AND T. DIETTERICH, *Learnability of the superset label learning problem*, in International conference on machine learning, PMLR, 2014, pp. 1629–1637.

[92] Q. LIU, X. LIAO, H. L. CARIN, J. R. STACK, AND L. CARIN, *Semisupervised multitask learning*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 31 (2009), pp. 1074–1086.

[93] M. LONG, Y. CAO, J. WANG, AND M. JORDAN, *Learning transferable features with deep adaptation networks*, in Proceedings of the 32nd International Conference on Machine Learning, PMLR, 2018, pp. 97–105.

[94] M. LONG, Z. CAO, J. WANG, AND M. I. JORDAN, *Conditional adversarial domain adaptation*, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 1647–1657.

[95] M. LONG, H. ZHU, J. WANG, AND M. I. JORDAN, *Deep transfer learning with joint adaptation networks*, in Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 2208–2217.

[96] Q. LOU AND L. JIANG, *Hemet: A homomorphic-encryption-friendly privacy-preserving mobile neural network architecture*, in Proceedings of the 38th International Conference on Machine Learning, PMLR, 2021, pp. 7102–7110.

[97] J. LU, H. ZUO, AND G. ZHANG, *Fuzzy multiple-source transfer learning*, IEEE Transactions on Fuzzy Systems, 28 (2020), pp. 3418–3431.

[98] G. MA, F. LIU, G. ZHANG, AND J. LU, *Learning from imprecise observations: An estimation error bound based on fuzzy random variables*, in Proceedings of the 2021 IEEE International Conference on Fuzzy Systems, 2021, pp. 1–8.

[99] G. MA, J. LU, F. LIU, Z. FANG, AND G. ZHANG, *Multiclass classification with fuzzy-feature observations: Theory and algorithms*, IEEE Transactions on Cybernetics, 54 (2022), pp. 1048–1061.

[100] Y. MANSOUR, M. MOHRI, AND A. ROSTAMIZADEH, *Domain adaptation: Learning bounds and algorithms*, in Proceedings of the 22nd Conference on Learning Theory, COLT, 2009.

[101] A. MAURER, *A vector-contraction inequality for rademacher complexities*, in Proceedings of the 27th International Conference on Algorithmic Learning Theory, Springer, 2016, pp. 3–17.

[102] Y. MAXIMOV, M.-R. AMINI, AND Z. HARCHAOUI, *Rademacher complexity bounds for a penalized multi-class semi-supervised algorithm*, Journal of Artificial Intelligence Research, 61 (2018), p. 761,Äì786.

[103] D. MCALLESTER, *A pac-bayesian tutorial with a dropout bound*, Computer Science, (2013).

[104] S. K. MEHER AND N. S. KOTHARI, *Interpretable rule-based fuzzy elm and domain adaptation for remote sensing image classification*, IEEE Transactions on Geoscience and Remote Sensing, 59 (2020), pp. 5907–5919.

[105] M. MOHRI, A. ROSTAMIZADEH, AND A. TALWALKAR, *Foundations of Machine Learning*, Massachusetts Institute of Technology, 2012.

[106] R. MÜLLER, S. KORNBLITH, AND G. HINTON, *When does label smoothing help?*, in Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, pp. 4694–4703.

[107] J. NA, H. JUNG, H. J. CHANG, AND W. HWANG, *Fixbi: Bridging domain spaces for unsupervised domain adaptation*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021.

[108] L. NANNI, S. GHIDONI, AND S. BRAHNAM, *Handcrafted vs. non-handcrafted features for computer vision classification*, Pattern Recognition, 71 (2017), pp. 158–172.

[109] M. OUSSALAH, *On the compatibility between defuzzification and fuzzy arithmetic operations*, Fuzzy Sets and Systems, 128 (2002), pp. 247–260.

[110] S. K. PAL AND A. GHOSH, *Fuzzy geometry in image analysis*, Fuzzy Sets and Systems, 48 (1992), pp. 23–40.

[111] F. PALUMBO AND C. N. LAURO, *A pca for interval-valued data based on midpoints and radii*, in New Developments in Psychometrics: Proceedings of the International Meeting of the Psychometric Society IMPS2001. Osaka, Japan, July 15–19, 2001, Springer, 2003, pp. 641–648.

[112] S. PAN AND Q. YANG, *A survey on transfer learning*, IEEE Transactions on Knowledge and Data Engineering, 22 (2010), pp. 1345–1359.

[113] S. J. PAN, J. T. KWOK, Q. YANG, ET AL., *Transfer learning via dimensionality reduction*, in Proceedings of the 23rd AAAI Conference on Artificial Intelligence, 2008, pp. 677–682.

[114] S. J. PAN, I. W. TSANG, J. T. KWOK, AND Q. YANG, *Domain adaptation via transfer component analysis*, IEEE transactions on neural networks, 22 (2010), pp. 199–210.

[115] N. PAPERNOT, S. SONG, I. MIRONOV, A. RAGHUNATHAN, K. TALWAR, AND U. ERLINGSSON, *Scalable private learning with pate*, in Proceedings of the 6th International Conference on Learning Representations, 2018.

[116] Z. PAWLAK, *Rough set theory and its applications to data analysis*, Cybernetics & Systems, 29 (1998), pp. 661–688.

[117] X. PENG, Q. BAI, X. XIA, Z. HUANG, K. SAENKO, AND B. WANG, *Moment matching for multi-source domain adaptation*, in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1406–1415.

[118] X. PENG, B. USMAN, N. KAUSHIK, J. HOFFMAN, D. WANG, AND K. SAENKO, *Visda: The visual domain adaptation challenge*, arXiv preprint arXiv:1710.06924, (2017).

[119] M. L. PURI AND D. A. RALESCU, *Fuzzy random variables*, Journal of Mathematical Analysis Applications, 114 (1986), pp. 409–422.

[120] X. QI, H. GUO, Z. ARTEM, AND W. WANG, *An interval-valued data classification method based on the unified representation frame*, IEEE Access, 8 (2020), pp. 17002–17012.

[121] X. QI, W. WANG, Y. SHI, H. QI, AND X. MU, *Agurf: An adaptive general unified representation frame for imbalanced interval-valued data*, Information Sciences, 641 (2023), p. 119089.

[122] A. B. RAMOS-GUAJARDO AND P. GRZEGORZEWSKI, *Distance-based linear discriminant analysis for interval-valued data*, Information Sciences, 372 (2016), pp. 591–607.

[123] A. RAMPONI AND B. PLANK, *Neural unsupervised domain adaptation in nlp,Äîa survey*, in Proceedings of the 28th International Conference on Computational Linguistics, Association for Computational Linguistics, 2020.

[124] C.-X. REN, Y.-H. LIU, X.-W. ZHANG, AND K.-K. HUANG, *Multi-source unsupervised domain adaptation via pseudo target domain*, IEEE Transactions on Image Processing, 31 (2022), pp. 2122–2135.

[125] J. A. RICE, *Mathematical statistics and data analysis*, Cengage Learning, 2006.

[126] Y. RONG, Z. WANG, P. A. HENG, AND K. S. LEUNG, *Classification of heterogeneous fuzzy data by choquet integral with fuzzy-valued integrand*, IEEE Transactions on Fuzzy Systems, 15 (2007), pp. 931–942.

[127] S. ROYCHOWDHURY AND W. PEDRYCZ, *A survey of defuzzification strategies*, International Journal of Intelligent Systems, 16 (2001), pp. 679–695.

[128] K. SAITO, D. KIM, S. SCLAROFF, AND K. SAENKO, *Universal domain adaptation through self-supervision*, in Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020, pp. 16282–16292.

[129] K. SAITO AND K. SAENKO, *Ovanet: One-vs-all network for universal domain adaptation*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9000–9009.

[130] K. SAITO, Y. USHIKU, AND T. HARADA, *Asymmetric tri-training for unsupervised domain adaptation*, in Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 2988–2997.

[131] P. SEEBÖCK, S. M. WALDSTEIN, S. KLIMSCHA, H. BOGUNOVIC, T. SCHLEGL, B. S. GERENDAS, R. DONNER, U. SCHMIDT-ERFURTH, AND G. LANGS, *Unsupervised identification of disease marker candidates in retinal oct imaging data*, IEEE transactions on medical imaging, 38 (2019), pp. 1037–1047.

[132] G. SHAFER, *Perspectives on the theory and practice of belief functions*, International Journal of Approximate Reasoning, 4 (1990), pp. 323–362.

[133] S. SHALEV-SHWARTZ AND S. BEN-DAVID, *Understanding machine learning: From theory to algorithms*, Cambridge university press, 2014.

[134] J. SHELL AND S. COUPLAND, *Fuzzy transfer learning: methodology and application*, Information Sciences, 293 (2015), pp. 59–79.

[135] J. SHEN, Y. QU, W. ZHANG, AND Y. YU, *Wasserstein distance guided representation learning for domain adaptation*, in Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 4058–4065.

[136] C. SHUI, R. PU, G. XU, J. WEN, F. ZHOU, C. GAGNÉ, C. X. LING, AND B. WANG, *Towards more general loss and setting in unsupervised domain adaptation*, IEEE Transactions on Knowledge and Data Engineering, (2023).

[137] A. K. SHUKLA, S. KUMAR, R. JAGDEV, P. K. MUHURI, AND Q. D. LOHANI, *Interval type-2 fuzzy weighted extreme learning machine for GDP prediction*, in Proceedings of the 2018 International Joint Conference on Neural Networks, 2018, pp. 1–8.

[138] B. SINOVA, M. Á. GIL, M. T. LÓPEZ, AND S. VAN AELST, *A parameterized l2 metric between fuzzy numbers and its parameter interpretation*, Fuzzy Sets and Systems, 245 (2014), pp. 101–115.

[139] S. SUN, J. YUN, H. LIN, N. ZHANG, A. ABRAHAM, AND H. LIU, *Granular transfer learning using type-2 fuzzy HMM for text sequence recognition*, Neurocomputing, 214 (2016), pp. 126–133.

[140] T. TAKAGI AND M. SUGENO, *Fuzzy identification of systems and its applications to modeling and control*, IEEE Transactions on Systems, Man, and Cybernetics, 15 (1985), pp. 116–132.

[141] M. E. TAYLOR, N. K. JONG, AND P. STONE, *Transferring instances for model-based reinforcement learning*, in Proceedings of the 2008th European Conference on Machine Learning and Knowledge Discovery in Databases-Volume Part II, 2008, pp. 488–505.

[142] S. TENG, Z. ZHENG, N. WU, L. TENG, AND W. ZHANG, *Adaptive graph embedding with consistency and specificity for domain adaptation*, IEEE/CAA Journal of Automatica Sinica, 10 (2023), pp. 2094–2107.

[143] C. V. THEODORIS, L. XIAO, A. CHOPRA, M. D. CHAFFIN, Z. R. AL SAYED, M. C. HILL, H. MANTINEO, E. M. BRYDON, Z. ZENG, X. S. LIU, ET AL., *Transfer learning enables predictions in network biology*, Nature, 618 (2023), pp. 616–624.

[144] L. TRAN AND L. DUCKSTEIN, *Comparison of fuzzy numbers using a fuzzy distance measure*, Fuzzy sets and systems, 130 (2002), pp. 331–341.

[145] L. UTKIN AND F. COOLEN, *Interval-valued regression and classification models in the framework of machine learning*, in Proceedings of the 7th International

Symposium on Imprecise Probability: Theories and Applications, 2011, pp. 371–380.

[146] W. VAN LEEKWIJCK AND E. E. KERRE, *Defuzzification: criteria and classification*, Fuzzy Sets and Systems, 108 (1999), pp. 159–178.

[147] H. VENKATESWARA, S. CHAKRABORTY, AND S. PANCHANATHAN, *Deep-learning systems for domain adaptation in computer vision: Learning transferable feature representations*, IEEE Signal Processing Magazine, 34 (2017), pp. 117–129.

[148] B. WANG, J. A. MENDEZ, M. B. CAI, AND E. EATON, *Transfer learning via minimizing the performance gap between domains*, in Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, pp. 10645–10655.

[149] G. WANG, T. ZHOU, K.-S. CHOI, AND J. LU, *A deep-ensemble-level-based interpretable takagi–sugeno–kang fuzzy classifier for imbalanced data*, IEEE transactions on cybernetics, 52 (2020), pp. 3805–3818.

[150] H. WANG, M. XU, B. NI, AND W. ZHANG, *Learning to combine: Knowledge aggregation for multi-source domain adaptation*, in Proceedings of the 16th European Conference on Computer Vision, 2020, pp. 727–744.

[151] R. WANG, C. LI, W. FU, AND G. TANG, *Deep learning method based on gated recurrent unit and variational mode decomposition for short-term wind power interval prediction*, IEEE transactions on neural networks and learning systems, 31 (2019), pp. 3814–3827.

[152] W. WANG, Z. ZHONG, W. WANG, X. CHEN, C. LING, B. WANG, AND N. SEBE, *Dynamically instance-guided adaptation: A backward-free approach for test-time domain adaptive semantic segmentation*, in Proceedings of the IEEE/CVF

Conference on Computer Vision and Pattern Recognition, 2023, pp. 24090–24099.

[153] L. WEN, X. LI, L. GAO, AND Y. ZHANG, *A new convolutional neural network-based data-driven fault diagnosis method*, IEEE Transactions on Industrial Electronics, 65 (2018), pp. 5990–5998.

[154] J. WESTON, C. WATKINS, ET AL., *Support vector machines for multi-class pattern recognition.*, in European Symposium on Artificial Neural Networks, vol. 99, 1999, pp. 219–224.

[155] F. WILCOXON, *Individual comparisons by ranking methods*, in Breakthroughs in Statistics, Springer, 1992, pp. 196–202.

[156] D. WU, V. J. LAWHERN, S. GORDON, B. J. LANCE, AND C.-T. LIN, *Driver drowsiness estimation from EEG signals using online weighted adaptation regularization for regression (OwARR)*, IEEE Transactions on Fuzzy Systems, 25 (2017), pp. 1522–1535.

[157] H. C. WU, *Probability density functions of fuzzy random variables*, Fuzzy Sets and Systems, 105 (1999), pp. 139–158.

[158] R. XIA, C. ZONG, X. HU, AND E. CAMBRIA, *Feature ensemble plus sample selection: domain adaptation for sentiment classification*, IEEE Intelligent Systems, 28 (2013), pp. 10–18.

[159] N. XIAO AND L. ZHANG, *Dynamic weighted learning for unsupervised domain adaptation*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 15242–15251.

[160] B. XIE, S. LI, F. LV, C. H. LIU, G. WANG, AND D. WU, *A collaborative alignment framework of transferable knowledge extraction for unsupervised domain adap-*

tation, IEEE Transactions on Knowledge and Data Engineering, 35 (2023), pp. 6518–6533.

[161] L. XIE, Z. DENG, P. XU, K.-S. CHOI, AND S. WANG, *Generalized hidden-mapping transductive transfer learning for recognition of epileptic electroencephalogram signals*, IEEE Transactions on Cybernetics, 49 (2018), pp. 2200–2214.

[162] C. XU, T. LIU, D. TAO, AND C. XU, *Local rademacher complexity for multi-label learning*, IEEE Transactions on Image Processing, 25 (2016), pp. 1495–1507.

[163] P. XU, Z. DENG, J. WANG, Q. ZHANG, K.-S. CHOI, AND S. WANG, *Transfer representation learning with tsk fuzzy system*, IEEE Transactions on Fuzzy Systems, 29 (2019), pp. 649–663.

[164] C. YANG, Z. DENG, K.-S. CHOI, AND S. WANG, *Takagi–Sugeno–Kang transfer learning fuzzy logic system for the adaptive recognition of epileptic electroencephalogram signals*, IEEE Transactions on Fuzzy Systems, 24 (2016), pp. 1079–1094.

[165] J. YANG, G. WANG, AND Q. ZHANG, *Knowledge distance measure in multigranulation spaces of fuzzy equivalence relations*, Information Sciences, 448 (2018), pp. 18–35.

[166] J. YANG, R. YAN, AND A. G. HAUPTMANN, *Cross-domain video concept detection using adaptive svms*, in Proceedings of the 15th ACM International Conference on Multimedia, 2007, pp. 188–197.

[167] L. YANG, Y. BALAJI, S.-N. LIM, AND A. SHRIVASTAVA, *Curriculum manager for source selection in multi-source domain adaptation*, in Proceedings of the 16th European Conference on Computer Vision, 2020, pp. 608–624.

[168] S. YANG, Y. WANG, J. VAN DE WEIJER, L. HERRANZ, AND S. JUI, *Generalized source-free domain adaptation*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8978–8987.

[169] X. YANG, Y. GU, K. WEI, AND C. DENG, *Exploring safety supervision for continual test-time domain adaptation*, in Proceedings of the 32nd International Joint Conference on Artificial Intelligence, 2023, pp. 1649–1657.

[170] X. YANG, G. ZHANG, J. LU, AND J. MA, *A kernel fuzzy c-means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises*, IEEE Transactions on Fuzzy Systems, 19 (2011), pp. 105–115.

[171] K. YOU, M. LONG, Z. CAO, J. WANG, AND M. I. JORDAN, *Universal domain adaptation*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2720–2729.

[172] C. ZHANG, E. ADELI, T. ZHOU, X. CHEN, AND D. SHEN, *Multi-layer multi-view classification for alzheimer's disease diagnosis*, in Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 4406–4413.

[173] Q. ZHANG, C. WANG, H. WU, C. XIN, AND T. V. PHUONG, *Gelu-net: a globally encrypted, locally unencrypted deep neural network for privacy-preserved learning*, in Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 3933–3939.

[174] X. ZHANG, X. ZHANG, H. LIU, AND X. LIU, *Multi-task clustering through instances transfer*, Neurocomputing, 251 (2017), pp. 145–155.

[175] Y. ZHANG, T. LIU, M. LONG, AND M. JORDAN, *Bridging theory and algorithm for domain adaptation*, in Proceedings of the 36th International Conference on Machine Learning, PMLR, 2019, pp. 7404–7413.

[176] S. ZHAO, B. LI, X. YUE, Y. GU, P. XU, R. HU, H. CHAI, AND K. KEUTZER, *Multi-source domain adaptation for semantic segmentation*, in Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, pp. 7287–7300.

[177] L. ZHONG, Z. FANG, F. LIU, J. LU, B. YUAN, AND G. ZHANG, *How does the combined risk affect the performance of unsupervised domain adaptation approaches?*, in Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 11079–11087.

[178] J. T. ZHOU, S. J. PAN, AND I. W. TSANG, *A deep learning framework for hybrid heterogeneous transfer learning*, Artificial Intelligence, 275 (2019), pp. 310–328.

[179] X. ZHU, H. SUK, S. LEE, AND D. SHEN, *Subspace regularized sparse multitask learning for multiclass neurodegenerative disease identification*, IEEE Transactions on Biomedical Engineering, 63 (2016), pp. 607–618.

[180] Y. ZHU, Y. CHEN, Z. LU, S. PAN, G.-R. XUE, Y. YU, AND Q. YANG, *Heterogeneous transfer learning for image classification*, in Proceedings of the 25th AAAI Conference on Artificial Intelligence, 2011, pp. 1304–1309.

[181] H. ZUO, J. LU, G. ZHANG, AND F. LIU, *Fuzzy transfer learning using an infinite gaussian mixture model and active learning*, IEEE Transactions on Fuzzy Systems, 27 (2018), pp. 291–303.

[182] H. ZUO, J. LU, G. ZHANG, AND W. PEDRYCZ, *Fuzzy rule-based domain adaptation in homogeneous and heterogeneous spaces*, IEEE Transactions on Fuzzy Systems, 27 (2019), pp. 348–361.

[183] H. ZUO, G. ZHANG, V. BEHBOOD, J. LU, AND X. MENG, *Transfer learning in hierarchical feature spaces*, in Proceedings of the 10th International Conference on Intelligent Systems and Knowledge Engineering, 2015, pp. 183–188.

[184] H. ZUO, G. ZHANG, J. LU, AND W. PEDRYCZ, *Fuzzy rule-based transfer learning for label space adaptation*, in Proceedings of the 2017 IEEE International Conference on Fuzzy Systems, 2017, pp. 1–6.

[185] H. ZUO, G. ZHANG, W. PEDRYCZ, V. BEHBOOD, AND J. LU, *Fuzzy regression transfer learning in Takagi–Sugeno fuzzy models*, IEEE Transactions on Fuzzy Systems, 25 (2017), pp. 1795–1807.

[186] H. ZUO, G. ZHANG, W. PEDRYCZ, V. BEHBOOD, AND J. LU, *Granular fuzzy regression domain adaptation in Takagi–Sugeno fuzzy models*, IEEE Transactions on Fuzzy Systems, 26 (2018), pp. 847–858.