ANFIS Interpolation and Its Application for Image Super Resolution

Jing Yang

Supervisors: Dr. Changjing Shang
Prof. Qiang Shen

Ph.D. Thesis

Department of Computer Science
Faculty of Business and Physical Sciences
Aberystwyth University

July 16, 2021

Declaration and Statement

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed ..... (candidate)

Date .....16/07/2021.....

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. Where **correction services**[1] have been used, the extent and nature of the correction is clearly marked in a footnote(s).

Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed ..... (candidate)

Date .....16/07/2021.....

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... (candidate)

Date .....16/07/2021.....

[1] This refers to the extent to which the text has been corrected by others.

Abstract

Fuzzy inference systems work through manipulating fuzzy if-then rules. Amongst the various existing fuzzy inference systems, the adaptive network based fuzzy inference system (ANFIS) has become one of the most powerful and popular tools for finding solutions to highly non-linear problems. Whilst promising for practical applications, most existing research related to ANFIS focuses on how to learn such an inference system with sufficient training data. However, in some real situations it is very hard or even impossible to get sufficient data for the required learning process. The shortage of training data significantly degrades the performance of such learned ANFIS models.

In light of this, the concept of ANFIS interpolation is proposed in the thesis, to deal with the problem of ANFIS construction with insufficient training data. It works by interpolating two well trained ANFISs in neighbouring domains where sufficient training data is available, in an effort to improve the performance of the constructed ANFIS in the current problem domain where there is only limited (or sparse) training data. Two types of ANFIS interpolation method are developed, including: 1) An initial method via group based rule interpolation, working through interpolating a group of fuzzy rules with the help of a rule dictionary; 2) An improved method via evolutionary process, in which the interpolated rules act as an initial population and are updated subsequently through the use of a genetic algorithm. Experimental results demonstrate the significantly improved performance of the proposed methods over the original non-interpolation ANFIS model, with the averaged root mean squared error (RMSE) reduced to 1.63 ± 0.39 from the original 3.71 ± 0.98.

Having recognised the capability of ANFIS in producing effective non-linear mappings, it is applied to address the problem of image super resolution in this thesis, in an effort to learn the non-linear mapping between a low resolution image and a high resolution one. Firstly, a natural image super resolution method with full training data is developed by learning multiple ANFIS mappings, showing the effectiveness of ANFIS model when sufficient training data is provided. A hyperspectral image super resolution method with sparse training data through the use of the proposed ANFIS interpolation methods is then developed, validating the efficacy of the interpolated ANFIS model when only sparse training data is available.

Acknowledgements

I would like to express my uttermost gratitude to my supervisors: Dr. Changjing Shang and Prof. Qiang Shen, for their motivation, supervision and kindness. Their patient guidance and warm encouragement go through all my Ph.D. and thesis writing. Without them, this Ph.D. and thesis would never have happened.

I am extremely grateful to all researchers from the Advanced Reasoning Group for their useful discussions and helpful suggestions. I am especially grateful to Dr. Ling Zheng, Dr. Tianhua Chen, Dr. Zhenpeng Li, Dr. Fangyi Li, Pu Zhang, and Muhanmude Ismail for their helpful advices.

I would like to thank all of the academic, administrative, technical, and support staff at the Department of Computer Science, Aberystwyth University, for their kind assistance throughout my entire study.

I would like to express the appreciation to my landlord lady Thelema for her tremendous help with my English. I would like to thank my housemates Fangyi Li, Xuejiao Sun, and all my dear friends in Aberystwyth.

My sincere gratitude goes to my entire family: my parents Yanwen Yang, Lijuan Yao, my husband Lu Chen and our lovely son Ruqian Chen. The completion of this Ph.D. would not have been possible without their kind support and encouragement.

Contents

Contents i

List of Figures iv

List of Tables vii

List of Algorithms ix

1 Introduction 1

1.1 Fuzzy Inference Systems(FIS) 3

1.2 Training Data Shortage Issues 5

1.3 Research Objectives and Contributions 6

1.4 Structure of Thesis 8

2 Background 13

2.1 Adaptive Network based Fuzzy Inference System (ANFIS) 13

2.1.1 Network architecture 14

2.1.2 Hybrid learning algorithm 17

2.2 Fuzzy Rule Interpolation (FRI) 19

2.2.1 The KH approach 21

2.2.2 The T-FRI approach 24

2.2.3 The TSK-type rule interpolation 29

2.3 Image Super Resolution 32

2.3.1 Problem description 32

2.3.2 Learning based image SR 35

2.4 Summary 38

3 ANFIS Interpolation - A Group Rule Interpolation Based Approach 39

3.1 The Framework for the Proposed Approach 40

3.2 Implementation of Proposed Approach 43
3.2.1 Rule dictionary generation 43
3.2.2 Intermediate ANFIS construction 44
3.2.3 ANFIS fine-tuning 49
3.3 Complexity Analysis 50
3.4 Experimentation 51
3.4.1 Experimental setup 51
3.4.2 Experiments on synthetic data 52
3.4.3 Experiments on real world benchmark data 61
3.4.4 Experiments with different settings 65
3.5 Summary 66

4 ANFIS Interpolation - An Evolutionary Approach 69
4.1 Genetic Algorithm 70
4.2 GA for ANFIS Interpolation 71
4.2.1 Chromosome representation: two strategies 72
4.2.2 Population initialisation 73
4.2.3 Crossover and mutation 77
4.2.4 Fitness function 80
4.2.5 Summary of GA-based ANFIS interpolation 81
4.2.6 ANFIS Interpolation: contrasting two algorithms 81
4.2.7 Fine-tuning of GA-learned ANFIS model 84
4.3 Complexity Analysis 84
4.3.1 Rule-based chromosome representation 85
4.3.2 ANFIS-based chromosome representation 86
4.4 Experimentation and Validation 87
4.4.1 Experimental setup 87
4.4.2 Experiments on synthetic data 88
4.4.3 Experiments on real data 96
4.5 Summary 103

5 Image Super Resolution with Full Training Data via Multiple Learned ANFIS Mappings 106
5.1 Framework of Proposed Approach 107
5.2 Implementation of the Proposed Approach 109
5.2.1 Learning of multiple ANFIS mappings 109

5.2.2 Image SR with learned ANFIS mappings 111
5.3 Experimentation and Validation 114
5.3.1 Experimental setup 115
5.3.2 Evaluation index 116
5.3.3 Results and discussion 116
5.4 Summary 122

6 Hyperspectral Image Super Resolution with Sparse Training Data via ANFIS Interpolation 124
6.1 Hyperspectral Image Super Resolution 125
6.1.1 Hyperspectral image (HSI) 125
6.1.2 HSI super resolution (HSI SR) 126
6.2 HSI SR with Sparse Data through ANFIS Interpolation 129
6.2.1 Training stage 130
6.2.2 Testing stage 134
6.3 Experimentation and Discussion 135
6.3.1 Experimental setup 135
6.3.2 Performance criteria 137
6.3.3 Results and discussion 138
6.4 Summary 145

7 Conclusion and Future Work 146
7.1 Summary of Thesis 146
7.1.1 ANFIS interpolation methods 147
7.1.2 Real world applications 148
7.2 Future Work 149
7.2.1 On ANFIS interpolation 149
7.2.2 On image super resolution 151

Appendix A Publications Arising from the Thesis 152
A.1 Journal Articles 152
A.2 Conference Papers 153

Appendix B List of Acronyms 154

Bibliography 156

List of Figures

1.1 General structure of a fuzzy inference system 3
1.2 Relationship among main thesis chapters 9

2.1 (a) TSK-type fuzzy reasoning. (b) Equivalent ANFIS structure representing two TSK rules 15
2.2 Dense and sparse rule-base 20
2.3 Lower and upper distances between two α-cuts. 22
2.4 Application examples of image super resolution reconstruction. Source images are obtained from [134, 147]. 32
2.5 Illustration of LR and HR images. Source images are obtained from [141]. 33
2.6 Single-frame and multi-frame SR. Source images are obtained from [141]. 33
2.7 Interpolation based image SR. Source images are obtained from [141]. . 34
2.8 Learning based image SR 36

3.1 Flowchart of proposed approach 42
3.2 Illustrative implementation of single rule interpolation: (a) Choosing K closest neighbours. (b) Calculating weights of chosen neighbours. (c) Generating new rule. 46
3.3 Integrating newly generated rules as an (intermediate) ANIFS 49
3.4 Illustration of data and source ANFISs. (a) Source data and target data used. (b) Membership functions in each source ANFIS. 53
3.5 Five folds taken from 5×5-fold cross validation for one-dimensional non-linear function approximation. 57
3.6 One fold of the two-dimensional function-approximation results: (a) Ground truth. (b) Result based on $\mathscr{A}_{ori}$. (c) Result based on $\mathscr{A}_{int}$. 59
3.7 Function approximation with more than two source domains 60
3.8 Function approximation with no target training data 61
3.9 Performance with no target training data under non-linear situation . . . 62
3.10 Performance vs. number of selected closest neighbours 66

3.11 Performance vs. amount of training data 67

4.1 Flowchart of proposed approach. (a) Overall flowchart. (b) Population initialization process. (c) Crossover process. (d) Mutation process. . . . 74
4.2 Individual generation from a seed. 76
4.3 Rule based chromosome representation used in implementation. 78
4.4 ANFIS based chromosome representation used in implementation. 79
4.5 Illustration of source data and target data. 89
4.6 Example for score table computation 94
4.7 Visual illustration of one-fold result by different ANFISs 96
4.8 Performance vs. percentage of training data used 103
4.9 Performance vs. crossover rate. 104
4.10 Performance vs. mutation rate. 104

5.1 Flowchart of the proposed approach 108
5.2 Example images used for training. 109
5.3 Basic idea of NLM – Similar pixel neighbourhoods lead to a large weight (w_1 or w_2), and different neighbourhoods result in a small weight (w_3). 114
5.4 Visual comparison of results from different methods on image 'Butterfly' with scale factor being 2 – From left to right: (a) HR image. (b) Bicubic. (c) Fuzzy Rule. (d) Sparse Representation. (e) ANFIS. (f) ANFIS & Post-processing. 117
5.5 Visual comparison of results from different methods on image 'Child' with scale factor being 3 – From left to right: (a) HR image. (b) Bicubic. (c) Fuzzy Rule. (d) Sparse Representation. (e) ANFIS. (f) ANFIS & Post-processing. 118
5.6 PSNR index of using different P (number of clusters in K-means algorithm) on Lena image with scale factor being 2. 122
5.7 Training time of using different P (number of clusters in K-means algorithm).123

6.1 Hyperspectral Image (HSI). 126
6.2 Band images of an HSI 127
6.3 Flowchart of the training process 130
6.4 Illustration of the 5 subsets 136
6.5 Super resolution results using different ANFISs on HSI 'toy' with scale factor being 2. 141

6.6 Super resolution results using different ANFISs on HSI ‘balloons’ with scale factor being 2. 142
6.7 Super-resolution results using different ANFISs on HSI ‘flowers’ with scale factor being 3. 143
6.8 Super-resolution results using different ANFISs on HSI ‘balloons’ with scale factor being 3. 144

List of Tables

2.1 Functionality of individual layers in ANFIS 17
2.2 Two passes in the hybrid learning procedure for ANFIS 19

3.1 Atoms in generated rule dictionary 54
3.2 Generated new rules for all cluster centroids 55
3.3 Rules in Interpolated ANFIS $\mathscr{A}_{int}$ 56
3.4 Experimental results on one-dimensional function 56
3.5 Experimental results on two-dimensional function 58
3.6 Public dataset used 62
3.7 Experimental results on real world datasets 64
3.8 p values of pairwise t-test results 65

4.1 Score table 81
4.2 Summary of complexity analysis 86
4.3 Functions used 88
4.4 Rule dictionary 90
4.5 Generating individuals for each cluster ($P = 5$) 91
4.6 Number of chromosomes in one (sub-)population during different processes within one evolutionary iteration 92
4.7 Accuracy (Mean ± Standard deviation) and running time (seconds) of different methods on synthetic data 95
4.8 Datasets used in experimental study 97
4.9 Experimental results using different ANFISs on real world datasets . . . 98
4.10 p values of pairwise t-test results 99
4.11 Accuracy (Mean ± Standard deviation) of different methods on real data 100
4.12 Average running time (seconds) of different methods on real data 102

5.1 SR results of different algorithms with scale factor = 2 119
5.2 SR results of different algorithms with scale factor = 3 120

6.1 HSI SR results of different algorithms with scale factor = 2 138
6.2 HSI SR results of different algorithms with scale factor = 3 139

List of Algorithms

3.1 ANFIS Interpolation - A group rule interpolation based approach. 41

4.1 Calculation of score table. 82
4.2 ANFIS Interpolation - An evolutionary approach. 83

5.1 Learning multiple ANFIS mappings. 110
5.2 Image super resolution via multiple ANFIS mappings. 115

6.1 Training stage for HSI SR with sparse data. 131
6.2 Testing stage for HSI SR with sparse data. 135

Chapter 1

Introduction

With the rapid development of computer hardware and software technologies, machine learning has become one of the most attractive topics in the field of computer science in recent years. There is no doubt that the world we live in today is full of all types of data (such as text data, image data, video data and so on), the fast growth of available data makes the management and analysis of data at hand ever more important and also ever more difficult. The machine learning theory is proposed aiming at extracting useful knowledge from those raw data instead of the human brain with higher speed and larger scale, which brings extreme convenience to human life. Different machine learning techniques, representative algorithms including the evolutionary computation, the neural networks, deep learning and fuzzy inference systems etc., are developed with different mechanisms and have different characteristics.

Evolutionary computation [9, 10] is one of the most popular machine learning methods, which offers a set of optimisation algorithms by analogy to natural evolution processes (e.g. crossover, mutation, and survival of the fitness evaluation for genetic algorithms). It provides an effective and efficient way for searching optimal solution in poorly understood and irregular problem spaces. Typically, evolutionary algorithms work with a population of individuals, in which each individual may be one or a set of potential solutions in the solution space. Whilst implemented in a stochastic manner, such algorithms perform a highly effective search in the problem hyperspace, iteratively directing the solution to promising regions. Typical evolutionary algorithms such as genetic algorithm (GA) [121], genetic programming

(GP) [1] and particle swarm optimizer (PSO) [175], have been widely employed for a variety of theoretical and practical applications [87, 143, 148, 161].

Another branch of the machine learning methods is the artificial neural network [54, 117]. The first neurocomputer, Dr. Robert Hecht-Nielsen, defined it as a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs. Although neural networks are an old topic, they have achieved significant breakthrough in recent decades. With the increase of storage capacity and calculation speed of computers, deep learning [40] technique has been developed as an extension of traditional artificial neural networks, with much more network layers and much more complex network structures. It can better extract and represent the features from data, especially for those multi-dimensional complex data (e.g. image data and voice data). Deep learning algorithms have been applied to various real word applications since proposed (such as classification, tracking, recognition, just to name a few), generating the state of the art results [53, 86, 127]. Different from traditional neural network methods, the layers of network in deep learning methods are much deeper with much more parameters. This gives the network more learning capacity, but increases the required storage space at the mean time.

Apart from the high accuracy of the deep learning algorithms, they suffer from having a huge amount of parameters in a network, which requires a large storage space and long running time in the network training process. Another argued issue in deep learning methods is the interpretability and transparency. Deep neural networks work as a black box, making the physical meaning of the resulting network structures not easy to explain. Different from deep learning, fuzzy inference systems provide an interpretable way for solving machine learning problems [115, 139]. They work by the use of a rule-base containing a set of fuzzy 'if-then' rules, in which each fuzzy set is described by a membership function. The interpretability and transparency of fuzzy inference systems is reflected in two aspects: 1) The fuzzy 'if-then' rules record the logical representation of the intuitive expert knowledge. 2) The fuzzy sets as well as the underlying membership functions preserve the linguistic meaning of the given label. Fuzzy inference systems have achieved significant successes in performing many real world tasks, such as classification, regression and prediction [4, 23, 48, 72, 114].

1.1 Fuzzy Inference Systems(FIS)

Fuzzy inference systems transform human knowledge to mathematical models (represented by a set of fuzzy rules), and conduct inference in a systematic manner. They are developed based on the fuzzy set theory [168, 177] and the fuzzy logic theory [167]. Based on the fact that the crisp description of the real world information is always imprecise and is easily affected by the subjective factors, fuzzy inference systems handle imprecision, a form of uncertainty to better describe the observations in the real world. The general framework of a fuzzy inference system is shown in Figure 1.1. There are four key components in such an FIS, including the fuzzification interface, the inference engine, the defuzzification interface and the knowledge base. These key components are introduced respectively as follows.

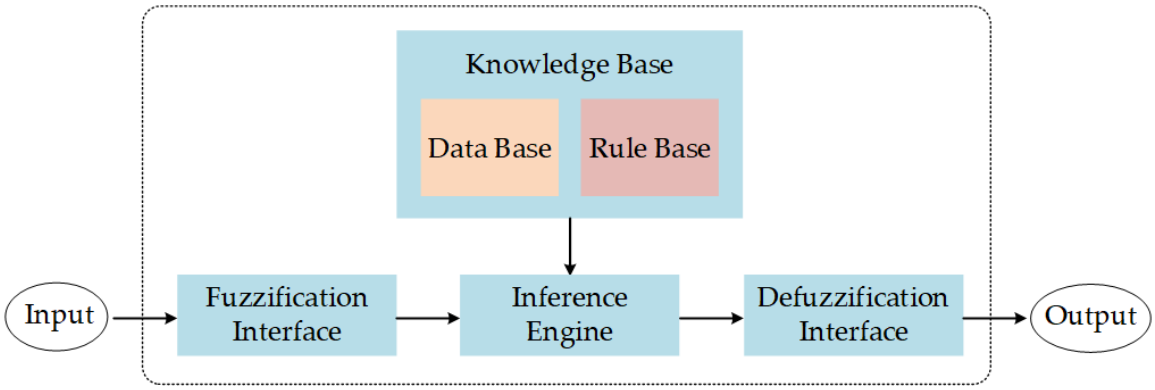


Figure 1.1: General structure of a fuzzy inference system

- The *fuzzification interface* transforms crisp values obtained from the real world measurements into fuzzy representations through the use of fuzzy sets. The definition of fuzzy set makes it possible to represent an input value with linguist meaning, which is implemented by the use of a membership function related to the fuzzy set. Take the representation of a person's height (say, 6 feet) as an example, if using the crisp representation, it is only a fixed value, while in terms of fuzzy representation, it can be interpreted as "This person is high with a high degree of truth". Theoretically any continuous or piecewise differentiable, convex and normal functions can be utilized as Membership Function (MF), commonly used MFs include triangular, trapezoidal, Gaussian and bell-shaped ones.

- The *inference engine* processes the incoming fuzzy values from the fuzzification interface and then makes the decision together with the knowledge base. The most commonly used inference engine is the compositional rule of inference [166]. It works by firing or matching rules in the rule-base, and the final decision is made by composing the consequences of all the fired rules.

- The *defuzzification interface* performs the inverse operation to the fuzzification interface. Usually in real world tasks, the output of a fuzzy inference system should be real values, keeping the same as its input, so the defuzzification interface is designed to map the fuzzy values obtained from the inference engine into crisp values. Generally, the choice of the defuzzification method depends on what kind of decision the fuzzy inference system is going to make. One of the most commonly used defuzzification methods is using the centre of the area, e.g. the centre of gravity (COG) of a triangular fuzzy set.

- The *knowledge base* contains two components, the rule-base, containing the collection of fuzzy 'if-then' rules, and the data base, containing the parameters in rules (e.g. membership functions of the fuzzy partitions associated to the linguistic variables).

  Fuzzy 'if-then' rules are the most important components in a fuzzy inference system. Different types of fuzzy rule have been developed in the literature. According to whether the consequence of a fuzzy rule is associated with its rule antecedent or not, the fuzzy rules can be classified into two categories, the mamdani-type rules and the Takagi-Sugeno-Kang (TSK)-type rules. In the mamdani-type rules [108], the input variables only appear in the antecedent parts, while in the TSK-type rules [131] those variables also appear in the consequent parts as a linear function of the input.

  Based on the two types of fuzzy rule, there are accordingly two forms of knowledge representation in fuzzy inference systems: Mandani-type models and TSK-type models. Mandani models are classical fuzzy systems which have been popular in many real world applications [19], while TSK representatives have also played an increasingly important role in such applications, including for example, stock market prediction [23] and EGG signals recognition [72]. The success of TSK-type fuzzy inference systems is largely owing to their capability of approximating complex non-linear functions [36].

Generally, there exist two ways to construct the knowledge base of a fuzzy inference system. One way is to translate expert knowledge into fuzzy rules directly, and the corresponding fuzzy inference systems are also called fuzzy expert systems [18, 103, 107]. It offers very good fuzzy representation of expert knowledge. But for large real-world problems, it may be very hard to ensure the efficiency and accuracy of the resulted knowledge base when such strategy is utilised. The other way for building an effective knowledge base is to use data-driven methods. In such FISs, the fuzzy rules are obtained from data by using machine learning techniques rather than obtained from expert knowledge directly [19, 30, 33, 34, 78].

1.2 Training Data Shortage Issues

Owing to the simplicity and explainability in representing human knowledge, fuzzy inference systems have become one of the most popular tools for finding solutions to various real world problems (e.g., [4, 30, 124]). They have been studied for many years, in both the design of the inference engine [136, 159, 166] and the induction of the rule-base [2, 163]. Amongst such research, how to construct an effective FIS from data is becoming a more and more attractive topic. For such data-driven based fuzzy inference systems, it is necessary to get sufficient training data for the required learning process in order to train a useful model. Most existing studies on fuzzy inference system construction assume that sufficient training data is available [116]. This assumption can hold for many cases in this age with sufficient well-distributed data, however, in some special situations, it is really hard or even impossible to obtain enough data for the required training procedure (such as some remote sensing tasks, or medical problems). In such cases, the shortage of data will severely restrict the potential of the learned fuzzy inference systems.

In the relevant literature, a typical way to solve this practical issue (of training data shortage) is to conduct the learning process through a so-called transfer learning procedure [41, 116, 128, 178, 179]. Such techniques exploit the knowledge accumulated from data in an auxiliary domain (termed source domain) to support predictive modelling in the problem domain at hand (termed target domain). Particularly, transfer learning usually works, by constructing non-linear mappings between the target and the source domain and transferring the data or the model parameters

between a certain source domain and a target domain via learned mappings. In so doing, it is feasible to utilise the knowledge regarding a given source model to approximately perform fuzzy inference in the target domain.

An alternative solution to this challenging problem is through the use of Fuzzy Rule Interpolation (FRI) technique [12, 84, 133]. Conventional fuzzy inference methods [136, 159, 166] must work with dense rule-base, in which the input region is completely covered. However, under the data shortage situations the resulting rule-base will be a sparse rule-base with 'gaps', therefore the conventional fuzzy inference methods cannot drive a conclusion. The FRI technique is developed to enable fuzzy inference to be performed on sparse rule-bases. In general, for a given observation that does not match any rule of a sparse rule-base, FRI techniques work by interpolating new rules from the existing ones. Typically, FRI interpolates a fuzzy rule by selecting and averaging given rules that are the closest to an unfired observation, thereby enabling fuzzy systems to perform inference even if no rules can be fired by pattern matching. Based on the general FRI idea, transformation based FRI (T-FRI) [58, 59] has been proposed for improved performance through scale and move transformations. This greatly increases the accuracy of the interpolated rules and has led to a number of advanced theoretical and applicational developments over the past decade (e.g., [24, 26, 27, 73, 93, 113, 155, 156]). However, the existing literature of FRI is mainly focussed on Mandani-type models, apart from limited initial attempt as reported in [95, 152, 172, 173], the research on FRI with TSK-type models is still rather rare.

1.3 Research Objectives and Contributions

Considering the data shortage problem mentioned above, the main goal of this thesis will focus on how to construct an effective fuzzy inference system with insufficient training data (termed as sparse data in this thesis). Among the various fuzzy inference systems, the ANFIS (Adaptive Network-based Fuzzy Inference System) which is implemented within the framework of a network structure, has become one of the most popular FISs due to its simplicity and effectiveness. Therefore, how to construct an ANFIS with insufficient training data (or sparse training data) is chosen as a key research objective in this thesis, although other fuzzy inference systems can also be studied in the same way.

In particular, the main research objective in the thesis can be achieved from the following three perspectives:

1. Inspired by the observation in transfer learning theory that knowledge accumulated from neighbouring domains (with sufficient data) may help the inference of the current problem domain of interest (with sparse data), in this thesis, the concept 'ANFIS interpolation' is proposed, which is defined as the construction of a new ANFIS in the target domain, by the assistance of two source ANFISs in the source domains. Reflecting the fact that an ANFIS is a set of fuzzy rules, an initial ANFIS interpolation method is proposed by interpolating a group of fuzzy rules, which is an extension of the popular fuzzy rule interpolation (FRI) technique.

2. The initial ANFIS interpolation method will terminate when a group of rules have been interpolated. How to further optimise these interpolated rules presents the second challenge that is addressed in the thesis. Evolutionary computation is introduced to iteratively update the interpolated rules by encoding those rules into specific chromosomes, in an effort to obtain improved performance.

3. Apart from the theoretical studies, another important goal of the thesis is to investigate how the proposed techniques may perform in dealing with real word problems. Particularly, the proposed methods are applied to the problem of image super resolution. Both full training data and sparse training data are used to evaluate the performance of the learned and the interpolated ANFIS models respectively.

Following these initial objectives, the main contributions of this thesis include ANFIS interpolation and its application to image super resolution problems, are listed as follows:

• An initial ANFIS interpolation technique via group rule interpolation is proposed, which is implemented by interpolating a group of fuzzy rules. Experimental results on benchmark data sets indicate that the proposed approach significantly improves the performance of ANFIS in data shortage situations. The proposed approach has been published in the *Proceedings of the 18th annual UK workshop on Computational Intelligence* and the journal of *IEEE Transactions on Cybernetics*.

• An improved ANFIS interpolation technique via evolutionary computation is proposed, which is implemented by using the interpolated rules as an initial population and updating the system via a genetic algorithm. This work is validated against a number of benchmark data sets, the results indicate that it further improves the performance of the initial ANFIS interpolation method. The work has been published in the journal of *IEEE transactions on Fuzzy Systems*.

• An image super resolution method via ANFIS is proposed, with full training data. It works by learning non-linear mappings from the low resolution image space to the high resolution image space using learned ANFIS models. Experiments indicate that ANFIS works very well for this practical image super resolution problem when sufficient training data is provided. This proposed approach has been published in the *Proceedings of the 2017 IEEE International Conference on Fuzzy Systems*.

• A hyperspectral image super resolution method via the proposed ANFIS interpolation technique is proposed, with sparse training data. Different from the natural RGB images, the hyperspectral images are special images which are much harder to obtain, so in such cases, the training data could be sparse. The experimental results of this real world application suggest that the proposed ANFIS interpolation techniques can not only significantly improve the poor performance caused by training data shortage on benchmark data sets, but also have promising application prospect. This work is currently being prepared for journal submission.

1.4 Structure of Thesis

This section outlines the structure of the remainder of this thesis. The relationship among the main chapters of this thesis is illustrated in Figure 1.2. In summary, Chapter 2 provides the background knowledge of ANFIS, image super resolution together with the review of typical fuzzy rule interpolation methods. Based on this, Chapters 3 and 4 present the theoretical part of this thesis and Chapters 5 and 6 give the application part of the thesis on image super resolution problems. Particularly, Chapter 3 presents an initial ANFIS interpolation method via group rule interpolation; Chapter 4 presents an improved ANFIS interpolation scheme via an evolutionary process; Chapter 5 provides the application of ANFIS on the image super resolution problem with full training data; Finally, Chapter 6 extends this application problem

using a special type of image (hyperspectral image), and applies the techniques developed in Chapters 3 and 4 for hyperspectral image super resolution with sparse training data.

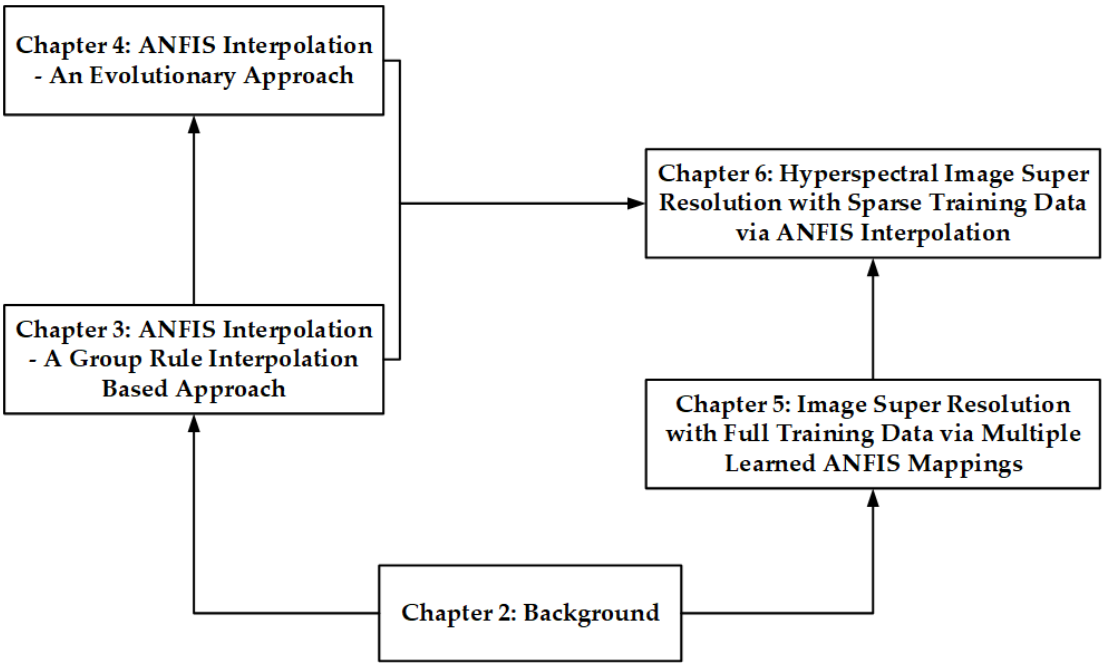


Figure 1.2: Relationship among main thesis chapters

Chapter 2: Background

This chapter provides the background knowledge that is relevant to the proposed work in this thesis. Primarily, a formal introduction of ANFIS including its network structure and the parameter learning strategy is presented, which forms a key basis upon which to develop the main research of the thesis. A brief introduction to the fuzzy rule interpolation technique is then followed in this chapter, which provides the fundamental knowledge for the proposed ANFIS interpolation. Finally, this chapter also introduces the basic knowledge of the image super resolution problem, forming the basis of the application part of the proposed work. The image super resolution (ISR) is a classical task in the field of image processing, and various classes of algorithms have been proposed in the literature, among which the most popular one in the recent decades is the learning based super resolution (SR) approaches, giving the state of the art results, on which the establishment of the proposed SR method in Chapter 5 is based.

Chapter 3: ANFIS Interpolation - A Group Rule Interpolation Based Approach

To deal with the data shortage situations in the construction of an ANFIS, this chapter proposes an initial ANFIS interpolation approach to improve the performance of the original ANFIS model trained with sparse training data using group based rule interpolation. In particular, it works by interpolating a group of fuzzy rules in a certain given problem domain with the assistance of existing ANFISs in its neighbouring domains. The construction process involves a number of computational mechanisms, including a rule dictionary which is created by extracting the rules from the existing ANFISs; a group of rules which are interpolated by exploiting the local linear embedding algorithm to build an intermediate ANFIS; and a fine-tuning method which refines the resulting intermediate ANFIS. The experimental evaluation on both synthetic and real world data sets is reported, demonstrating that when facing the data shortage situations, the proposed approach helps significantly improve the performance of the original ANFIS modeling mechanism.

Chapter 4: ANFIS Interpolation - An Evolutionary Approach

This chapter presents an extension of the method in Chapter 3, improving the interpolated ANFIS model through the use of evolutionary computation. The proposed approach takes the interpolated rules as an initial population, and updates these candidate rules via a genetic process. The crossover and mutation operations over these candidate rules are then executed in an effort to attain candidates of improved performance. When this genetic learning process terminates the chromosomes in the final population either collectively form or each individually represent a learned ANFIS, depending on whether a single fuzzy rule or a set of fuzzy rules representing an entire ANFIS is implemented with a chromosome within the evolving population. Comparative experimental evaluations on both synthetic and real world data sets are carried out, demonstrating that in spite of data shortage, the proposed interpolation approach is able to produce ANFIS models that significantly outperform those trained using existing learning mechanisms.

Chapter 5: Image Super Resolution with Full Training Data via Multiple Learned ANFIS Mappings

This chapter provides an application of ANFIS on the image super resolution problem. The proposed approach aims to generate a high resolution image using a low resolution one, using multiple learned ANFIS mappings. It presents an implemented learning system that captures the relationship between a high resolution image space and a low resolution one using an external image database. In particular, a collected large number of low resolution and high resolution image patch pairs are divided into different groups with a clustering method. For each clustered group of the training samples, an ANFIS mapping is learned for super resolution. The post processing techniques (including a non-local means filter and an iterative back projection operator) are subsequently employed to suppress the displeasing artefacts of the resulting reconstructed high resolution image. The proposed approach is evaluated using a range of natural images and compared with a number of existing popular ISR algorithms, demonstrating its effectiveness.

Chapter 6: Hyperspectral Image Super Resolution with Sparse Training Data via ANFIS Interpolation

Given the promising results achieved with ANFIS interpolation generated from Chapters 3 and 4 on benchmark data sets, in this chapter, the proposed ANFIS interpolation techniques are applied to a real-world hyperspectral image super resolution problem. Hyperspectral images are a special image type, in which there are dozens or hundreds of image bands while some of the image bands are very hard to obtain. This characteristic means that the training data in some image bands are sparse. The proposed super resolution approach uses the learned ANFIS models over the image bands with sufficient training data as source ANFISs, to assist the construction of the target ANFIS models over the image bands with sparse training data. Experimental results show that compared with the original ANFIS model, the interpolated ANFIS model gives remarkable improvements over the data shortage bands.

Chapter 7: Conclusion

This chapter summarises this thesis and lists the key contributions, together with a discussion of topics which set up the basis for future research.

Appendices

Appendix A lists the publications arising from the work presented in this thesis, including the published journal papers and conference papers. Appendix B summarises the abbreviations employed throughout the thesis.

Chapter 2

Background

FUZZY inference system enables the tolerance of imprecision when dealing with real life inference problems, which is closer to human reasoning. Among the various types of existing fuzzy inference systems, those implemented in an adaptive network structure (ANFIS) have gained more and more attention in the recent decades, owing to their simplicity and effectiveness.

This chapter reviews the background knowledge relevant to the work to be presented in the later part of this thesis, the remainder of this chapter is structured as follows. ANFIS, which is the main research topic of the thesis, is firstly introduced in Section 2.1, with its network structure and associated parameter learning method. Section 2.2 introduces the basic idea and the typical algorithms of the fuzzy rule interpolation method, which will form the base of the interpolation of ANFISs. A brief introduction to image super resolution and the most popular representative method (the learning based image super resolution) is presented in Section 2.3. Finally Section 2.4 summarizes this chapter.

2.1 Adaptive Network based Fuzzy Inference System (ANFIS)

ANFIS [66] is a fuzzy inference system that implements approximate reasoning within the general framework of an adaptive network. It works by equivalently extracting

useful (interpretable) knowledge in terms of a set of fuzzy rules directly from training data, and has proven to be very suitable for highly non-linear problems. Thanks to its simplicity and effectiveness, it has been widely applied to various kinds of real world problems (e.g., [11, 47, 135, 144]). The following gives a formal introduction to the basic concepts of ANFIS, including an illustrative network structure and the associated data-driven process for learning the parameters of such networks.

2.1.1 Network architecture

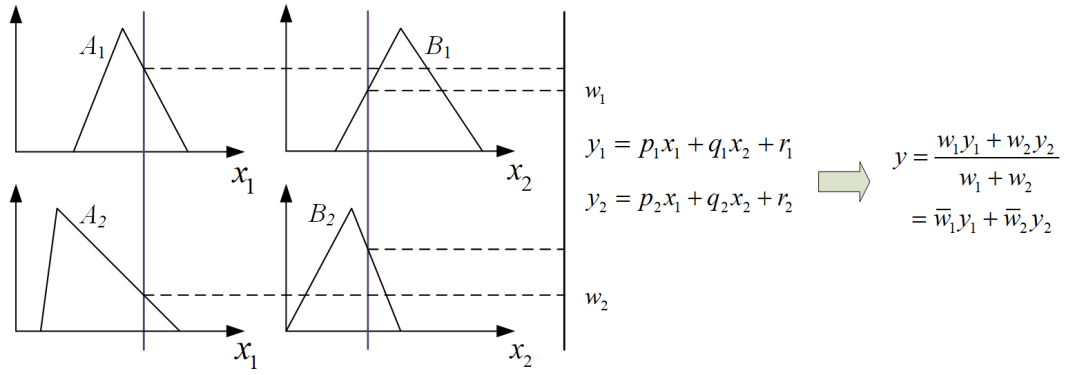
For easy understanding, a simple ANFIS with two input variables (x_1 and x_2) and one output variable (y) is used here for illustration (whilst more complex structures can be readily expanded from this basic form). In particular, suppose that there are two fuzzy if-then rules of Takagi-Sugeno-Kang's type [130] in the rule base of this example ANFIS, as follows:

Rule 1: If x_1 is A_1 and x_2 is B_1, then $y_1 = p_1 x_1 + q_1 x_2 + r_1$

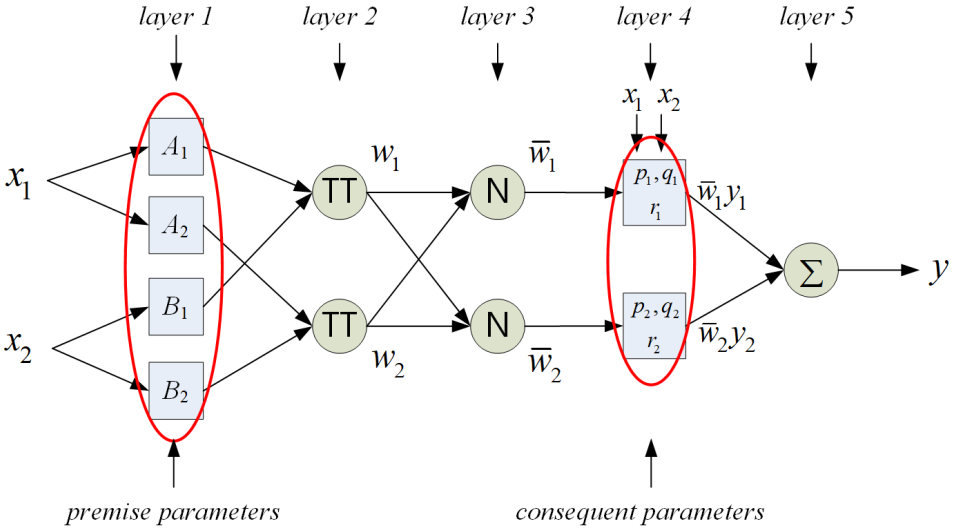
Rule 2: If x_1 is A_2 and x_2 is B_2, then $y_2 = p_2 x_1 + q_2 x_2 + r_2$

Figure 2.1 shows the corresponding TSK-type fuzzy reasoning process (a) and the equivalent ANFIS structure (b). As shown in Figure 2.1(a), in the TSK-type fuzzy reasoning, the output of each rule is a linear combination of all the input variables, and the overall output is the weighted average ($\bar{w}_1 y_1 + \bar{w}_2 y_2$) of each rule's output. The network structure of a general ANFIS contains five layers of computing elements, as shown in Figure 2.1(b), In this figure, the square nodes stand for adaptive computing units with modifiable parameters, and the circle nodes represent those fixed units without parameters. Further details of the individual layers within this ANFIS are outlined below.

Layer 1: Each node i in this layer is a square unit defined by a fuzzy set O_{ij} of the membership function: $\mu_{O_{ij}}(x_i)$, $i, j \in \{1, 2\}$, where x_i denotes the input variable to node i, and $O_{i1} \in \{A_i | i = 1, 2\}$ and $O_{i2} \in \{B_i | i = 1, 2\}$ denote the fuzzy sets defined on the domains of x_1 and x_2, respectively. Such a membership function can be specified as any continuous or piecewise differentiable, convex and normal functions such as trapezoidal, triangular, bell-shaped ones. For example, a triangular fuzzy set is defined as follows in general terms,



(a)



(b)

Figure 2.1: (a) TSK-type fuzzy reasoning. (b) Equivalent ANFIS structure representing two TSK rules

$$\mu_O(x) = \begin{cases} k_1x + b_1 & a_0 \leq x \leq a_1 \\ k_2x + b_2 & a_1 \leq x \leq a_2 \\ 0 & otherwise \end{cases} \tag{2.1}$$

where a_0 and a_2 are the two extreme points delimiting the fuzzy set with a membership value of 0, and a_1 stands for the normal point of the set whose membership value is 1. $k_1 = 1/(a_1 - a_0)$, $b_1 = -a_0/(a_1 - a_0)$, $k_2 = 1/(a_1 - a_2)$, and $b_2 = -a_2/(a_1 - a_2)$. In ANFIS terms, k_1, k_2, b_1, b_2 are called premise parameters as they are associated

with the underlying input variable (which appears in the antecedent part of a rule). Incidentally, the notion of representative value that is often used in FRI for such a triangular-shaped fuzzy set $O(a_0, a_1, a_2)$ can be simply defined as:

$$Rep(O) = (a_0 + a_1 + a_2)/3 \tag{2.2}$$

The Gaussian-shaped and bell-shaped functions are also commonly used membership functions, which are defined in Equation (2.3) and (2.4) respectively:

$$\mu_O(x) = exp[-(\frac{x-m}{\sigma})^2] \tag{2.3}$$

$$\mu_O(x) = \frac{1}{1 + [(\frac{x-c}{a})^2]^b} \tag{2.4}$$

where m, σ, a, b, c are the corresponding premise parameters.

Layer 2: Each node in this layer is a circle unit which multiplies the incoming membership of each attribute and gives the product as its (local) output, denoted by w_i. w_i is also the firing strength of the ith rule ($i = 1, 2$):

$$w_i = \mu_{A_i}(x_1) \times \mu_{B_i}(x_2) \tag{2.5}$$

Layer 3: Each node in this layer is also a circle unit, calculating the relative proportion of the ith rule's firing strength to the total of both rules' firing strengths:

$$\bar{w_i} = \frac{w_i}{w_1 + w_2} \tag{2.6}$$

where again, $i = 1, 2$. To reflect the underlying semantics, the outputs of this layer are termed as normalised firing strengths.

Layer 4: Each node i in this layer is a square unit implementing the following linear function:

$$\bar{w_i}(p_i x_1 + q_i x_2 + r_i) = \bar{w_i} y_i \tag{2.7}$$

where $\bar{w_i}$ is the output of the previous layer, and p_i, q_i, r_i are the parameters associated with the rule consequents and hence, are termed consequent parameters hereafter.

Layer 5: This layer is the (global network) output layer, which consists of a single circle unit, computing the overall output in response to the current input to the network, defined as the summation of all its incoming values from the previous layer, namely,

$$y = \sum_i \bar{w}_i y_i = \bar{w}_1 y_1 + \bar{w}_2 y_2 \tag{2.8}$$

Each layer within a general ANFIS is summarised in Table 2.1.

Table 2.1: Functionality of individual layers in ANFIS

Layer	Activity	Parameter
1	Fuzzifying crisp value	Premise parameter
2	Computing firing strength of each rule	No
3	Normalising firing strength	No
4	Computing local output	Consequent parameter
5	Computing global output	No

2.1.2 Hybrid learning algorithm

Given the network structure, the only uncertain part of an ANFIS concerns the parameters in the first layer (premise parameters describing the fuzzy sets) and the fourth layer (consequent parameters specifying the linear functions). In the original work reported in [66], these parameters are trained using a hybrid learning method [67] combining gradient descent and Least Square Estimation (LSE).

Considering a simple adaptive network structure in which there is only one output variable:

$$output = F(\vec{I}, S) \tag{2.9}$$

where $F(\cdot,\cdot)$ denotes the function of the fuzzy inference system, $\vec{I}$ denotes the set of the input variables, and S denotes the set of parameters. If there exists a function H such that the composite function $H \circ F$ is linear in a part of the parameters of S, then these parameters can be identified by the LSE. Suppose that S can be decomposed into two sets S_1 and S_2:

$$S = S_1 \oplus S_2 \tag{2.10}$$

where $\oplus$ denotes direct sum, and $H \circ F$ is linear in the elements of S_2. Applying H to Equation (2.9), it can be written as:

$$H(output) = H \circ F(\vec{I}, S) = H \circ F(\vec{I}, S_1 \oplus S_2) \tag{2.11}$$

Since $H \circ F$ is linear for S_2, when giving the values of parameters in S_1, and specifying the training data into Equation (2.11), a linear matrix equation $AX = B$ can be obtained, where X denotes the unknown vector whose elements are the parameters in S_2. By minimizing the squared error $||AX - B||^2$ using LSE, the solution of $AX = B$ (denoted by X^*) can be expressed as:

$$X^* = (A^T A)^{-1} A^T B \tag{2.12}$$

where $(A^T A)^{-1} A^T$ is the pseudo-inverse of A. Since Equation (2.12) is a concise solution, it is computational expensive for doing the matrix inverse. So instead of solving Equation (2.12) directly, a sequential formula [8] is employed to iteratively compute the LSE of X:

$$\begin{cases} X_{i+1} = X_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T X_i) \\ S_{i+1} = S_i - \dfrac{S_i a_{i+1} a_{i+1}^T S_i}{1 + a_{i+1}^T S_i a_{i+1}} \end{cases} \tag{2.13}$$

where $i = 0, 1, \cdots, P-1$, P denotes the number of training data pairs, a_i^T is the i th row vector of A, b_i^T is the i th element of B, S_i is called the covariance matrix with $S_0 = \gamma I$ (γ is a positive large number and I is the identity matrix), X_0 is set to be $X_0 = 0$, and the least square estimate X^* is equal to X_P.

Based on the above analysis, the parameters in S_2 can be calculated by Equation (2.13), then the gradient descent and LSE can be combined to update the parameters. Each loop of the hybrid learning algorithm is composed of a forward pass and a backward pass. In the forward pass, the input training data go through each node of the network until matrices A and B are obtained, then the parameters in S_2 can be identified using the sequential least square formulas in Equation (2.13). After that, the function signals keep going forward to get the estimated output of the current loop, which will be used to calculate the output error together with the desired output. Then, in the backward pass, with the identified parameters in S_2, the parameters in S_1 is computed by the gradient descent algorithm.

From the above ANFIS architecture (see Figure 2.1(b)), it is observed that given the values of premise parameters, the overall output can be expressed as a linear

combinations of the consequent parameters. More precisely, the output y in Figure 2.1(b) can be rewritten as:

$$\begin{aligned} y =& \frac{w_1}{w_1+w_2}y_1 + \frac{w_2}{w_1+w_2}y_2 \\ =& \bar{w_1}y_1 + \bar{w_2}y_2 \\ =& (\bar{w_1}x_1)p_1 + (\bar{w_1}x_2)q_1 + (\bar{w_1})r_1 \\ &+ (\bar{w_2}x_1)p_2 + (\bar{w_2}x_2)q_2 + (\bar{w_2})r_2 \end{aligned} \tag{2.14}$$

which is linear in the consequent parameters (p_1, q_1, r_1, p_2, q_2 and r_2). Define S, S_1, S_2 in Equation (2.10) as follows:

$$\begin{cases} S \rightarrow set\ of\ total\ parameters \\ S_1 \rightarrow set\ of\ premise\ parameters \\ S_2 \rightarrow set\ of\ consequent\ parameters \end{cases} \tag{2.15}$$

and define $H(\cdot)$ in Equation (2.11) as the identify function, then the composite function $H \circ F$ satisfies the condition that $H \circ F$ is linear in a part of parameters in S (i.e. the parameters in S_2), therefore the hybrid learning algorithm (the combination use of LSE method and gradient descent method) can be applied directly. More specifically, in the forward pass, the input values of the training samples are fed forward to compute the output with the premise parameters being fixed, and the consequent parameters are optimised by LSE. Then, in the backward pass, the consequent parameters are set to be fixed, while the error rates between the computed output and the expected ones (as part of the given training samples) are propagated backward, with the premise parameters updated using the gradient descent method. Table 2.2 summarizes the activities in each pass.

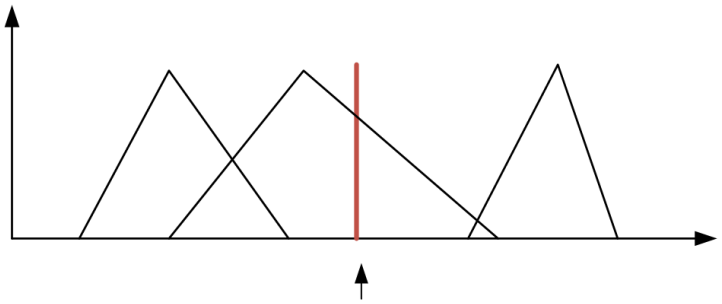
Table 2.2: Two passes in the hybrid learning procedure for ANFIS

-	Forward pass	Backward pass
Premise parameters	Fixed	Gradient descent
Consequent parameters	LSE	Fixed
Signals	Node Outputs	Error Rates

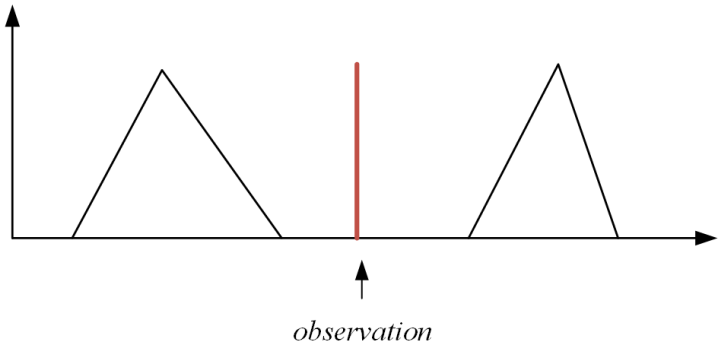
2.2 Fuzzy Rule Interpolation (FRI)

Fuzzy systems utilise a fuzzy rule-base to perform the inference. For most fuzzy inference systems, it is assumed that the rule-bases are complete. The complete

rule-base (or dense rule-base) ensures that for any possible input, there exists at least one fired fuzzy rule whose antecedent part overlaps the input data. Therefore, the conclusion can be drawn by combining the consequences of all the fired rules. However, in certain problems, the rule-base can be incomplete (or sparse) in which some rules are missed. For a sparse rule-base, the input space is not fully covered, there may exist an empty space between the membership functions of two rule antecedents. In this case, for the observations that fall into the 'empty space', the conventional fuzzy reasoning methods are highly likely to conclude nothing. Figure 2.2 shows the difference between a dense rule-base and a sparse rule-base. Note that there is a subtle nuance between the concepts of 'sparse data' and 'sparse rule-base'. A sparse rule-base is generated from sparse data, but sparse data does not necessarily lead to a sparse rule-base. Even sparse data can result in a dense rule-base with badly generated rules.



(a) Dense rule-base



(b) Sparse rule-base

Figure 2.2: Dense and sparse rule-base

Fuzzy rule interpolation (FRI) techniques are proposed to overcome this challaging issue, which enable a fuzzy inference system with a sparse rule-base to perform

inference [85, 94]. The basic idea of FRI is to interpolate a rule at the place where no rule can be fired with the given observation, by combining the information of the neighbouring rules. A number of FRI methods have been developed in the literature [75, 76, 120], which can be approximately classified into two categories, the single step rule-based interpolation and the intermediate rule based interpolation. The most representative approaches for these two categories are the KH (Kóczy and Hirota) method and the T-FRI (transformation based FRI) method respectively, which will be reviewed in the following sections. Most existing FRI techniques in the literatures are with Mamdani-type rules, while the development of TSK-type FRI is still rather rare. Some initial work has been reported most recently [31, 95, 173], which will be reviewed later.

2.2.1 The KH approach

The KH (Koczy and Hirota) method [83, 84] is the very first and classic FRI technique, which is based on the α-cuts theory. The α-cut A_α of a fuzzy set A is defined as:

$$A_\alpha = \{x | A(x) \geqslant \alpha, \alpha \in [0,1]\} \tag{2.16}$$

where $A(x)$ is the membership function of fuzzy set A. Theoretically, every fuzzy set can be approximated by the use of the family of its α-cuts.

For a given observation A^* that does not match with any rule in the rule-base, the KH method selects a few (usually two) neighbouring rules and then conducts the linear combination to obtain the inference result. For simplicity, suppose that both of the two selected flanking rules (R_1 and R_2) are with only one input variable, formatted as follows:

R_1 : $If\ x\ is\ A_1,\ then\ y\ is\ B_1$

R_2 : $If\ x\ is\ A_2,\ then\ y\ is\ B_2$

The KH method produces the conclusion using the assumption that the ratio of distances between the conclusion and the rule consequents should be identical with that of distances between the observation and the rule antecedents. The required conclusion B^* is determined by the following equation:

$$\frac{d(A^*,A_1)}{d(A^*,A_2)} = \frac{d(B^*,B_1)}{d(B^*,B_2)} \tag{2.17}$$

where $d(.,.)$ denotes the fuzzy distance between the two fuzzy sets. According to the resolution principle [125, 165], every fuzzy set can be represented as a family of its α-cuts. Therefore the Equation (2.17) can be written as:

$$\frac{d(A^*_\alpha,A_{1\alpha})}{d(A^*_\alpha,A_{2\alpha})} = \frac{d(B^*_\alpha,B_{1\alpha})}{d(B^*_\alpha,B_{2\alpha})} \tag{2.18}$$

where $\alpha \in [0,1]$. Define the lower and upper distances between α-cuts A^*_α and $A_{1\alpha}$ as follows:

$$d_L(A^*_\alpha,A_{1\alpha}) = d(inf\{A^*_\alpha\}, inf\{A_{1\alpha}\}) \tag{2.19}$$

$$d_U(A^*_\alpha,A_{1\alpha}) = d(sup\{A^*_\alpha\}, sup\{A_{1\alpha}\}) \tag{2.20}$$

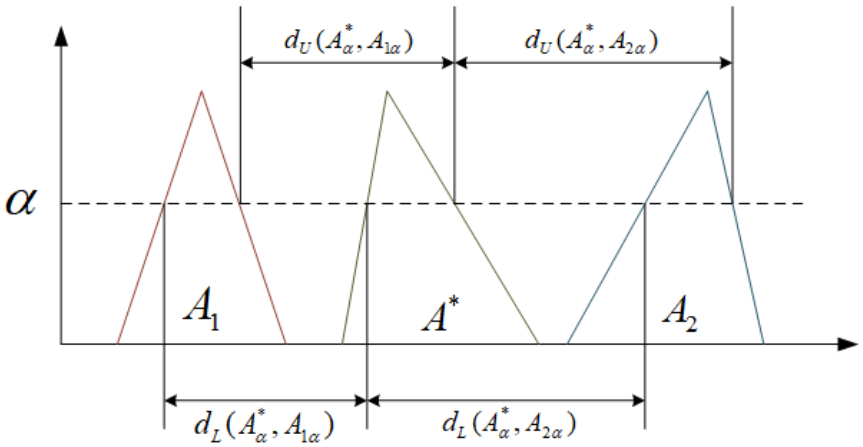


Figure 2.3: Lower and upper distances between two α-cuts.

The lower and upper distances are illustrated in Figure 2.3. Using the lower and upper distances, Equation (2.18) is transformed into:

$$\frac{d_L(A^*_\alpha,A_{1\alpha})}{d_L(A^*_\alpha,A_{2\alpha})} = \frac{d_L(B^*_\alpha,B_{1\alpha})}{d_L(B^*_\alpha,B_{2\alpha})} \tag{2.21}$$

$$\frac{d_U(A^*_\alpha,A_{1\alpha})}{d_U(A^*_\alpha,A_{2\alpha})} = \frac{d_U(B^*_\alpha,B_{1\alpha})}{d_U(B^*_\alpha,B_{2\alpha})} \tag{2.22}$$

From Equation (2.19), Equation (2.21) can be written as:

$$\begin{aligned}\frac{d_L(A^*_\alpha, A_{1\alpha})}{d_L(A^*_\alpha, A_{2\alpha})} &= \frac{d_L(B^*_\alpha, B_{1\alpha})}{d_L(B^*_\alpha, B_{2\alpha})} \\ &= \frac{d(inf\{B^*_\alpha\}, inf\{B_{1\alpha}\})}{d(inf\{B^*_\alpha\}, inf\{B_{2\alpha}\})} \\ &= \frac{inf\{B^*_\alpha\} - inf\{B_{1\alpha}\}}{inf\{B_{2\alpha}\} - inf\{B^*_\alpha\}}\end{aligned} \tag{2.23}$$

Then it can be solved as follows:

$$\begin{aligned}inf\{B^*_\alpha\} &= \frac{inf\{B_{1\alpha}\}d_L(A^*_\alpha, A_{2\alpha}) + inf\{B_{2\alpha}\}d_L(A^*_\alpha, A_{1\alpha})}{d_L(A^*_\alpha, A_{1\alpha}) + d_L(A^*_\alpha, A_{2\alpha})} \\ &= \frac{\frac{inf\{B_{1\alpha}\}}{d_L(A^*_\alpha, A_{1\alpha})} + \frac{inf\{B_{2\alpha}\}}{d_L(A^*_\alpha, A_{2\alpha})}}{\frac{1}{d_L(A^*_\alpha, A_{1\alpha})} + \frac{1}{d_L(A^*_\alpha, A_{2\alpha})}}\end{aligned} \tag{2.24}$$

$sup\{B^*_\alpha\}$ can be calculated using the same way by combining Equation (2.20) and (2.22), resulting in

$$inf\{B^*_\alpha\} = \frac{\frac{inf\{B_{1\alpha}\}}{w_{1L}} + \frac{inf\{B_{2\alpha}\}}{w_{2L}}}{w_{1L} + w_{2L}} \tag{2.25}$$

$$sup\{B^*_\alpha\} = \frac{\frac{sup\{B_{1\alpha}\}}{w_{1U}} + \frac{sup\{B_{2\alpha}\}}{w_{2U}}}{w_{1U} + w_{2U}} \tag{2.26}$$

where

$$w_{1L} = \frac{1}{d_L(A^*_\alpha, A_{1\alpha})}, \quad w_{2L} = \frac{1}{d_L(A^*_\alpha, A_{2\alpha})} \tag{2.27}$$

$$w_{1U} = \frac{1}{d_U(A^*_\alpha, A_{1\alpha})}, \quad w_{2U} = \frac{1}{d_U(A^*_\alpha, A_{2\alpha})} \tag{2.28}$$

From this, B^*_α can be obtained by $B^*_\alpha = [inf\{B^*_\alpha\}, sup\{B^*_\alpha\}]$. Then the conclusion fuzzy set B^* can be constructed by the resolution principle of fuzzy set, as follows:

$$B^* = \bigcup_{\alpha \in [0,1]} B^*_\alpha \tag{2.29}$$

Theoretically, all possible α-cuts (which is an infinite number) should be considered to interpolate an approximate conclusion. However, in order to reduce the

complexity in practice, only a finite number (usually 2, 3 or 4) of α-cuts are taken into consideration. The KH approach forms the base of the FRI theory, based on this fundamental KH method, many extended versions have been proposed [132, 137, 146]. For instance, the modified α-cut based interpolation method [132] solves the 'abnormal fuzzy set problem' in the original KH approach through coordinate modification, and reference [146] extends the classical KH approach to more complex version involving multiple rules with multiple antecedent variables.

2.2.2 The T-FRI approach

Apart from the earliest linear FRI method (KH method), a number of FRI methods have been proposed to improve the performance of interpolative reasoning. The most popular group will be the transformation-based approaches, termed as scale and move transformation-based fuzzy rule interpolation (T-FRI) [58, 59]. It can handle both interpolation and extrapolation, and guarantees the normality as well as convexity of the interpolated fuzzy sets. The T-FRI is valid for multiple variables involved in multiple fuzzy rules with various types of membership functions (e.g. triangular, trapezoidal, Gaussian and bell-shaped MFs). Due to the simplicity and popularity of the triangular-shaped membership function, here only the T-FRI approach with triangular-shaped MF is introduced as a representative.

2.2.2.1 Representative value

An important concept in the T-FRI is the representative value (Rep) of a fuzzy set that must be defined first. This value captures the important information (such as the overall location) of a fuzzy set, and will guide the scale and move transformations later. For an arbitrary polynomial fuzzy set with n odd points $A = (a_0, \ldots, a_{n-1})$, its general Rep is defined as below:

$$Rep(A) = \sum_{i=0}^{n-1} w_i a_i \tag{2.30}$$

with w_i is the weight assigned to point a_i. Different calculation methods of these weights can result in different representative value. The simplest way to specify the

weights is called the 'average Rep' method, where all points take the same weight value:

$$Rep(A) = \frac{1}{n}\sum_{i=0}^{n-1} a_i \tag{2.31}$$

For example, if A denotes a triangular-shaped fuzzy set with three points a_0, a_1 and a_2, the average Rep is computed as: $Rep(A) = \frac{(a_0 + a_1 + a_2)}{3}$.

2.2.2.2 Main steps in T-FRI

Based on the definition of Rep value, the T-FRI approach can be implemented as three key steps. For a given unfired observation, firstly, K closest rules is selected from the sparse rule-base. Then an intermediate rule is generated by weighted averaging the chosen closest rules. Finally, the scale and move transformation is applied on the intermediate rule to obtain the final result. These three steps will be described in detail as follows.

A. K closest rules selection

Without losing generality, a fuzzy rule R_i in the sparse rule-base and an unfired observation O^* is represented as follows:

$$Rule\ R_i : if\ X_1\ is\ A_{i1}\ and \ldots and\ X_m\ is\ A_{im},\ then\ Y\ is\ B_i \tag{2.32}$$

$$O^* : X_1\ is\ A_1^*\ and \ldots and\ X_m\ is\ A_m^* \tag{2.33}$$

where A_{ij} denotes the jth antecedent fuzzy set in rule R_i, B_i is the consequent fuzzy set of rule R_i, A_j^* denotes the jth fuzzy set in observation O^*, and m is the number of input variables.

The fuzzy distance d_{ij} between two fuzzy sets is defined as the distance of their corresponding representative vales:

$$d_{ij} = d(A_{ij}, A_j^*) = d(Rep(A_{ij}), Rep(A_j^*)) \tag{2.34}$$

Since different variables may have different ranges, each distance measure related to one variable is normalised to [0, 1], as follows:

$$d'_j = \frac{d(A_{ij}, A^*_j)}{max_j - min_j} = \frac{d(Rep(A_{ij}), Rep(A^*_j))}{max_j - min_j} \tag{2.35}$$

where max_j and min_j denote the maximal and minimal values of variable X_j respectively. Then the distance d_i between rule R_i and observation O^* is computed as:

$$d_i = \sqrt{\sum_{j=1}^{m} (d'_j)^2} \tag{2.36}$$

After calculating the distances $\{d_i\}$ between a given observation and all the fuzzy rules in the sparse rule-base, the K rules that have the minimal distances will be chosen as the K closest rules, and will be used for constructing the intermediate rule subsequently.

B. Intermediate rule construction

By combining the K closest rules selected from the sparse rule-base, the intermediate rule will result, which has the following format:

$$if\ X_1\ is\ A'_1\ and \ldots and\ X_m\ is\ A'_m, then\ Y\ is\ B' \tag{2.37}$$

where $A'_j, j = 1, 2, \ldots, m$ is calculated as:

$$A'_j = A''_j + \delta_j (max_j - min_j) \tag{2.38}$$

A''_j is the weighted average of the K selected rules:

$$A''_j = \sum_{k=1}^{K} w'_{kj} A_{kj} \tag{2.39}$$

where each $w'_{kj} = w_{kj} / \sum_{k=1,\ldots,K} w_{kj}$ denotes the normalized weight, so that the sum of all the weights will equal to one. The weight w_{kj} is computed according to the fuzzy distance between O^*_j and A_{kj}, generally, a larger distance will result in a smaller weight. Usually w_{kj} can be simply defined as (2.40), although alternative non-increasing functions (such as $w_{kj} = exp^{-d(O^*_j, A_{kj})}$) can also be adopted to assign the weights.

$$w_{kj} = \frac{1}{d(O^*_j, A_{kj})} \tag{2.40}$$

In (2.38), δ_j is a constant defined by:

$$\delta_j = \frac{Rep(A_j^*) - Rep(A_j'')}{max_j - min_j} \tag{2.41}$$

In doing so, the Rep value of A_j' will keep the same with that value of A_j^*. By analogy to the antecedent part, the consequent part B' of the intermediate rule is obtained by:

$$B' = B'' + \delta_a(max - min) \tag{2.42}$$

where max, min are the maximal and minimal value of the output variable Y respectively, and B'' and δ_a are computed as:

$$B'' = \sum_{k=1}^{K} w'_{k_a} B_k \tag{2.43}$$

$$\delta_a = \frac{1}{m} \sum_{j=1}^{m} \delta_j \tag{2.44}$$

where K denotes the number of closest rules selected from the sparse rule-base, $w'_{k_a} = (1/m) \sum_{j=1,\ldots,m} w'_{kj}$ is the mean value of w'_{kj}, $j = 1, \ldots, m$. The subscript a in δ_a and w'_{k_a} means 'average'.

C. Scale and move transformation

The construction method of the intermediate rule can only insure that the Rep value of A_j' and O_j^* as close as possible, but the shapes of these two fuzzy sets can be rather different, so the scale and move transformation is then introduced to solve this problem. This process is implemented in two stages: scale transformation and move transformation.

- Scale transformation: For a given fuzzy set $A_j' = (a'_{j0}, a'_{j1}, a'_{j2})$ of the intermediate rule, the scale factor s_{A_j} is defined by:

$$s_{A_j} = \frac{a^*_{j2} - a^*_{j0}}{a'_{j2} - a'_{j0}} \tag{2.45}$$

  where a^*_{j0} and a^*_{j2} are the two extreme points of fuzzy set $A_j^* = (a^*_{j0}, a^*_{j1}, a^*_{j2})$. Then, the scaled fuzzy set $\bar{A}_j' = (\bar{a}'_{j0}, \bar{a}'_{j1}, \bar{a}'_{j2})$ can be calculated as follows:

$$\begin{cases} \bar{a}'_{j0} = \dfrac{(1+2s_{A_j})a'_{j0} + (1-s_{A_j})a'_{j1} + (1-s_{A_j})a'_{j2}}{3} \\ \bar{a}'_{j1} = \dfrac{(1-s_{A_j})a'_{j0} + (1+2s_{A_j})a'_{j1} + (1-s_{A_j})a'_{j2}}{3} \\ \bar{a}'_{j2} = \dfrac{(1-s_{A_j})a'_{j0} + (1-s_{A_j})a'_{j1} + (1+2s_{A_j})a'_{j2}}{3} \end{cases} \tag{2.46}$$

According to the intuition that a similar rule antecedent will lead to a similar rule consequent, the scaled consequent fuzzy set $\bar{B}' = (\bar{b}'_0, \bar{b}'_1, \bar{b}'_2)$ can be obtained by using the same scale factor:

$$\begin{cases} \bar{b}'_0 = \dfrac{(1+2s_B)b'_0 + (1-s_B)b'_1 + (1-s_B)b'_2}{3} \\ \bar{b}'_1 = \dfrac{(1-s_B)b'_0 + (1+2s_B)b'_1 + (1-s_B)b'_2}{3} \\ \bar{b}'_2 = \dfrac{(1-s_B)b'_0 + (1-s_B)b'_1 + (1+2s_B)b'_2}{3} \end{cases} \tag{2.47}$$

where $s_B = (1/m)\sum_{j=1,\ldots,m} s_{A_j}$ is the average value of all s_{A_j}.

- Move transformation: The move factor is defined as:

$$m_{A_j} = \begin{cases} \dfrac{3(a^*_{j0} - \bar{a}'_{j0})}{\bar{a}'_{j1} - \bar{a}'_{j0}} & a^*_{j0} \geq \bar{a}'_{j0} \\ \dfrac{3(a^*_{j0} - \bar{a}'_{j0})}{\bar{a}'_{j2} - \bar{a}'_{j1}} & a^*_{j0} \leq \bar{a}'_{j0} \end{cases} \tag{2.48}$$

Given m_{A_j}, the conclusion B^* can be estimated as:

$$\begin{cases} \begin{cases} b^*_0 = \bar{b}'_0 + m_B \dfrac{\bar{b}'_1 - \bar{b}'_0}{3} \\ b^*_1 = \bar{b}'_1 - 2m_B \dfrac{\bar{b}'_1 - \bar{b}'_0}{3} \\ b^*_2 = \bar{b}'_2 + m_B \dfrac{\bar{b}'_1 - \bar{b}'_0}{3} \end{cases} & if\ m_B \geq 0 \\ \begin{cases} b^*_0 = \bar{b}'_0 + m_B \dfrac{\bar{b}'_2 - \bar{b}'_1}{3} \\ b^*_1 = \bar{b}'_1 - 2m_B \dfrac{\bar{b}'_2 - \bar{b}'_1}{3} \\ b^*_2 = \bar{b}'_2 + m_B \dfrac{\bar{b}'_2 - \bar{b}'_1}{3} \end{cases} & otherwise \end{cases} \tag{2.49}$$

where $m_B = (1/m)\sum_{j=1,\ldots,m} m_{A_j}$.

Based on this fundamental T-FRI approach, a series of advanced improvements have been developed over the past decades [24, 73, 74, 112, 155, 156], including the backward fuzzy rule interpolation [73], the rough fuzzy rule interpolation [24], the dynamic fuzzy rule interpolation [112] and the adaptive fuzzy rule interpolation [155, 156], just to name a few.

2.2.3 The TSK-type rule interpolation

Most existing FRI techniques in the literatures are the Mamdani-type rule interpolation, while the development of TSK-type FRI is still rather rare, although some initial work has most recently been reported in [31, 95, 173]. As mentioned previously, different from the Mamdani-type rules in which the consequent part is a fuzzy set, in the TSK-type rules the consequent is usually a crisp polynomial function. A typical TSK-type rule R_i can be represented as follows:

$$R_i : if\ x_1\ is\ A_{i1}\ and \ldots and\ x_m\ is\ A_{im}, then\ y_i = \sum_{j=0}^{m} p_{ij} x_j \tag{2.50}$$

where $x_0 = 1$ is the constant term of the polynomial function. Therefore, the TSK-type fuzzy rule interpolation methods are also slightly different as compared with the classical Mamdani-type fuzzy rule interpolation methods. This section summarizes the main steps of the existing representative TSK-type FRI methods reported in [31, 95, 173].

For a given observation $O^*(if\ x_1\ is\ A_1^*\ and \ldots and\ x_m\ is\ A_m^*)$ and the rule R_i in the sparse rule-base represented as Equation (2.50), the TSK-type rule interpolation approaches (with triangular-shaped membership functions) are implemented as the following steps:

Step 1: Compute the similarity measure between the observation O^* and each rule R_i in the sparse rule-base;

The similarity measure can be defined using various ways. The simplest and also the most popular method to define the similarity measure is using the fuzzy distance between two fuzzy sets, which is wildly adopted in the literature (such as reference [31], and the T-FRI approaches [58, 59]). Alternatively in the TSK+ approach [95], a new similarity measurement between two triangular fuzzy sets is developed , taking both the similarity and the distance between the two fuzzy sets into consideration. Suppose that $A = (a_0, a_1, a_2)$ and $A' = (a_0', a_1', a_2')$ are two fuzzy sets in a normalized variable domain, their similarity degree $S(A, A')$ is defined as follows:

$$S(A, A') = (1 - \frac{\sum_{i=0}^{2} |a_i - a_i'|}{3}) \cdot d \tag{2.51}$$

where d is called the 'distance factor' that is calculated by:

$$d = \begin{cases} 1 & a_0 = a_1 = a_2 \ \& \ a_0' = a_1' = a_2' \\ 1 - \dfrac{1}{1 + e^{(-s \cdot ||A,A'|| + 5)}} & otherwise \end{cases} \tag{2.52}$$

where $||A,A'||$ denotes the fuzzy distance between A and A' (which is defined as the distance between their representative values), $s(s > 0)$ is a sensitive factor, where smaller s will result in the similarity degree $S(A,A')$ more sensitive to the distance between the two fuzzy sets. The constant 5 in the equation ensures that the distance factor d is close to one when the fuzzy distance between A and A' is zero.

The proposed similarity measure has the following properties:

1) A lager $S(A,A')$ indicates higher similarity degree between A and A';

2) A and A' are identical if and only if $S(A,A') = 1$;

3) $S(A,A') > 0$ unless ($a_0 = a_1 = a_2 = 0$ and $a_0' = a_1' = a_2' = 1$) or ($a_0 = a_1 = a_2 = 1$ and $a_0' = a_1' = a_2' = 0$).

Step 2: Choose K closest rules according to the similarity measure;

Similar with the Mamdani-type rule interpolation, the TSK-based approaches also choose a portion of closest rules to contribute to the interpolation result according to the calculated similarity measures. The choice of K can be various in different approaches: In the TSK+ method [95], all the rules in the sparse rule-base are utilized without selection; in reference [173], K is set to be a fixed small value to avoid the bad influence of the rules with low similarity; while in the weighted T-FRI approach [93], it has been proven that only two chosen rules can achieve the best performance.

Step 3: Construct an intermediate rule;

The intermediate rule is constructed by weighted averaging all the selected K closest rules. Suppose that the constructed intermediated rule has the following format:

$$R' : if \ x_1 \ is \ A_1' \ and \ldots and \ x_m \ is \ A_m', then \ y' = \sum_{j=0}^{m} p_j' x_j \tag{2.53}$$

where

$$A_j' = \sum_{k=1}^{K} w_k A_{kj}, \ j = 1, \ldots, m. \tag{2.54}$$

$$p'_j = \sum_{k=1}^{K} w_k p_{kj},\ j = 0, \ldots, m. \tag{2.55}$$

The weights w_k are calculated according to the similarity between the observation O^* and the K closest rules R_k. Generally, a higher similarity between O^* and R_k will result in larger weight w_k, although different methods can be designed to calculate the weights. For instance, in TSK+ [95] the weights are computed using the proposed similarity measure:

$$S(O_1^*, A_{k1}) \wedge S(O_2^*, A_{k2}) \wedge \cdots \wedge S(O_m^*, A_{km}) \tag{2.56}$$

where $S(O_j^*, A_{kj}), j = 1, \cdots, m$ is defined by Equation (2.51), $\wedge$ is a t-norm that is usually implemented by minimum operator. An alternative way to obtain the weights proposed in [31] is through solving a linear equation group with the constraint that $Rep(O_j^*) = Rep(A_{kj})$:

$$\begin{cases} Rep(A_1^*) = \sum_{k=1}^{K} w_k Rep(A_{k1}) \\ \vdots \\ Rep(A_j^*) = \sum_{k=1}^{K} w_k Rep(A_{kj}) \\ \vdots \\ Rep(A_m^*) = \sum_{k=1}^{K} w_k Rep(A_{km}) \end{cases} \tag{2.57}$$

In comparison with the possible alternatives that typically use the Euclidean norm and usually rely on an extra move operator when constructing the intermediate rule [58, 59], such constraints directly assess how much contribution of each selected rule may make to the resulting intermediate rule, avoiding more complicated computation.

Step 4: Inference using the intermediate rule.

Finally, the inference result will be obtained by using the constructed intermediate rule. In summary, the overall process of several existing TSK-type fuzzy rule interpolation is very similar with the intermediate rule based Mamdani-type FRI approaches, they all have the K closest rule selection and the intermediate rule construction steps. However, there are also some new techniques introduced to the TSK-type interpolation methods (e.g. the computation methods of the similarity measure, and different ways of identifying weights). Of course, the TSK-based FRI methods are still at their infancy with rare existing approaches, more promising methods will be developed in the future.

2.3 Image Super Resolution

High resolution (HR) images are desired in many real world applications, such as medical diagnosis, remote sensing problems and so on [164] (shown as Figure 2.4). However, due to hardware and communication channel limitations as well as budget constraints, images are sometimes obtained with low resolution (LR). To increase the spatial resolution of an image can be accomplished by either reducing the pixel size by sensor manufacturing techniques or increasing the chip size of the sensors, both of which are severely constrained by the physical limitation of imaging systems and also lead to huge budget burden. Therefore, image super resolution (SR) reconstruction algorithms [118] have been developed in order to efficiently improve the spatial resolution of a given image using computer software, and also to provide clear images for the subsequent image processing tasks (such as recognition, classification and detection).

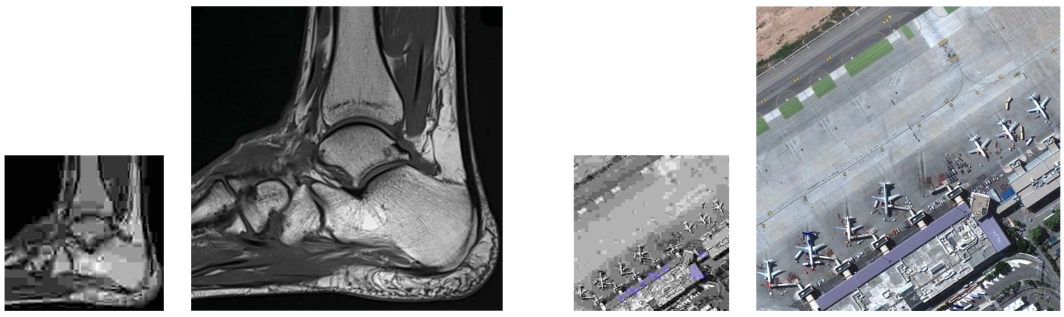
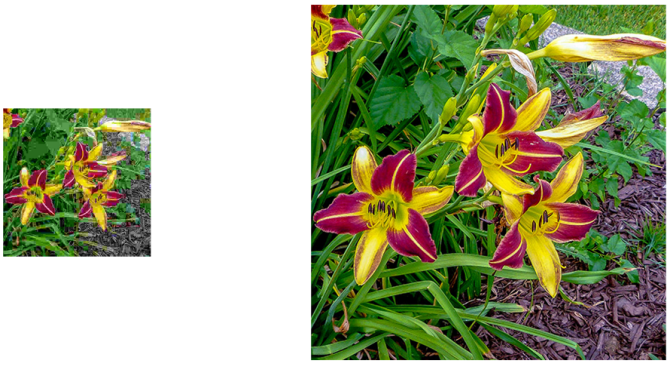


Figure 2.4: Application examples of image super resolution reconstruction. Source images are obtained from [134, 147].

Image super resolution techniques aim to generate an HR image from one or more given LR images. This section firstly gives a general description of the image super resolution problem, then the most popular super resolution algorithm (the learning based SR) is introduced in detail.

2.3.1 Problem description

Image resolution reflects how much detail is contained in an image, an image with higher resolution means that more details are captured. An illustration of HR image and LR image of the same scene is shown in Figure 2.5.



(a) LR image (b) HR image

Figure 2.5: Illustration of LR and HR images. Source images are obtained from [141].

Image super resolution techniques, also termed as image zooming or image enlargement, aim to generate an HR image using one or multiple input LR images. According to the number of the input LR images used, the existing SR approaches can be broadly categorized into two classes: 1) single-frame approaches using only one LR input image, and 2) multi-frame approaches using multiple LR input images of the same scene(shown as Figure 2.6).

Single-frame SR Multi-frame SR

Figure 2.6: Single-frame and multi-frame SR. Source images are obtained from [141].

Muti-frame images SR approaches [97, 99, 170] usually need geometrical registration of sub-pixel level over the multiple input LR images, and then the registered images are fused to obtain an HR image. The accurate sub-pixel level registration can be very time consuming, and inaccurate registration will seriously effect the super resolution results. Also, images of the same scene are sometimes very hard to obtain due to the weather changes, light limitation or the noise affect. Therefore, the single-frame SR algorithms are getting more and more popular as they do not

need to collect various LR images of the same scene as input. In single-frame SR problems, the observed LR image **X** is assumed to be the down sampled version of the desired HR image **Y**:

$$\mathbf{X} = H\mathbf{Y} \tag{2.58}$$

Here, H represents the down sampling operator. Single-frame SR is therefore, the inverse problem of the above image degradation equation. For a given LR input **X**, infinitely many HR images **Y** satisfy Equation (2.58), thus super resolution is an extremely ill-posed inverse problem. In order to handle such a problem, extra information is needed.

Existing single-frame SR methods [13, 14, 50, 150] typically adopt three different types of techniques: the interpolation based methods, the reconstruction based methods and the learning based methods. The interpolation based methods [56, 96] are the oldest image SR algorithms which are implemented by simply interpolating the image pixels. For example, as shown in Figure 2.7, images are seen as matrices of pixels in computer (the A, B, C denote image pixels). When an image is enlarged, the number of pixels will increase, so the task of image SR is to estimate the missing pixels. Interpolation based approaches aim to interpolate the existing pixels to fill in the blanks. The most commonly used interpolation methods include the nearest interpolation, bilinest interpolation and the bicubic interpolation, which are already proven techniques and have been wildly used in computer software or web pages for zooming pictures.

Figure 2.7: Interpolation based image SR. Source images are obtained from [141].

However, researchers have pointed out that large size is not the only meaning of high resolution, the term 'high resolution' also means more information and clearer details. Also, the interpolation based approaches usually utilise a base function that only considers the correlations of neighbouring pixels to approximate the missing pixels in HR images, therefore tending to cause blurred edges and generating overly

smoothed images. So the resulting image generated by the interpolation methods is still a low resolution image to some extent, and they are often be used as the initial step during the whole super resolution process in the improved approaches. After that, the interpolated large size image will be sequentially enhanced to get the real high resolution one, and this enhancement process is also what the recent super resolution researches focus on.

To further improve the image resolution, the reconstruction based methods [77, 98] are proposed, in which prior knowledge is introduced in an attempt to restraint the solution. For instance, the sparse representation base image SR [150] uses the image sparsity in the frequency domain as an extra constraint added in the reconstruction process, improving the quality of the enlarged image significantly; the similarity based image SR algorithms [22, 25] add the similarity constraint among image patches in an effort to eliminate the artefacts or noise in the reconstructed HR images; and in [35], the heavy-tail distribution property of image histogram is utilized as an extra prior knowledge to improve the super resolution accuracy. Although such strategies may offer much better performance than interpolation based techniques, important problems such as missing details remain.

More recently, the learning based methods have been proposed to tackle the challenging ISR problem via learning dictionaries [158] or relationships [122, 149, 171] from an external image database. Empirical results have shown that these approaches generally lead to improved visual outcomes, thereby forming the state of the art results in single frame ISR. The following section will give a brief introduction of the learning based ISR and review some typical literatures.

2.3.2 Learning based image SR

The learning based super resolution approaches collect a large amount of image pairs of both resolutions as the training data set, which will be subsequently utilised to learn the relationship between the high resolution image and the low resolution image. And once the relationship is learned, it can be applied on new testing images. So when it comes a new low resolution image, the corresponding high resolution image is estimated using the learned relationship '*R*', with patch-by-patch or pixel-by-pixel strategy. This is illustrated in Figure 2.8. Typical examples of learning based SR are listed in references [43, 122, 171]. In particular, the learning based SR methods

mainly include the learning of linear mappings, the learning of deep convolutional neural networks and the learning of fuzzy rules. These three representative types of learning based approaches are briefly introduced as follows.

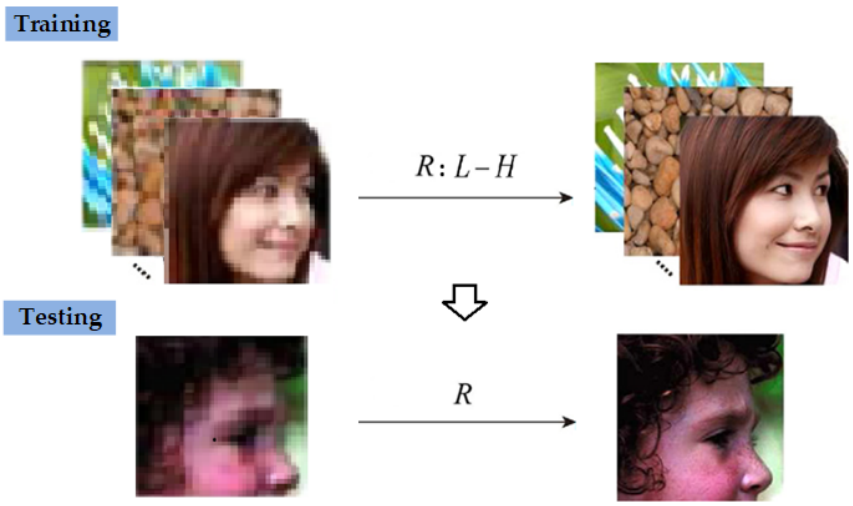


Figure 2.8: Learning based image SR

A. Learning of linear mappings

The type of relationships or mappings R can be various. Generally, they can be classified into two classes: the linear class, and the non-linear class. Linear mappings are first proposed and commonly used in the literatures due to their simplicity [14, 149, 171]. For example, the method proposed in [171] works by learning multiple linear mappings from the LR feature space onto the HR feature space, in order to perform single-frame image SR. And in [149], the feature space is split into numerous subspaces, and exemplars for each subspace are collected to learn the effective linear mapping functions.

But in more cases, linear mappings are too simple to describe the complex relationship between the low resolution image and the high resolution image, so the non-linear mappings are proposed to achieve better performances. Existing literatures mentioned two types of non-linear mappings: the deep convolutional neural network, and the fuzzy rule.

B. Learning of deep CNNs

As the rapid increasement of the computer storage capacity and the calculation speed, the traditional neural network can be designed much deeper than ever before,

which greatly improves the network ability in capturing accurate features and has been wildly utilized in a range of image processing tasks [53, 86, 101, 142]. The deep convolutional neural network (CNN) is one of the most powerful network structures in deep learning. For deep CNN based image super resolution problems, different CNNs are designed and utilised to extract image features and construct an end-to-end non-linear mapping between LR and HR images [43, 52, 88]. Owing to the powerful capacity in capture deep level image features, these approaches represent the state of the art techniques in image SR.

However, such a strategy usually requires a large amount of network parameters to work successfully, and may consume a lot of time for training despite their accuracy. A possible alternative is to explore fuzzy rule based approaches [122] which can produce non-linear mappings through the use of a set of fuzzy rules, resulting in lightweight models and making the training process faster.

C. Learning of fuzzy rules

As an LR image patch can be generated from any of the many possible HR image patches, this uncertainty of the image SR problem makes it natural to introduce fuzzy technique into the learning based approaches, which is well demonstrated in reference [122]. The image SR methods based on learning of fuzzy rules work by transferring the image pixels (or image features) of a given image into corresponding fuzzy representations and using fuzzy 'if-then' rules to describe the non-linear relationship between the LR and the HR images. Given a pixel (or an extracted image feature) in the low resolution image as an input, the corresponding pixel in the high resolution image will be the output of the fuzzy rule inference.

The work of [122] addresses the ISR problem through fuzzy clustering and fuzzy rule learning, updating the desired HR pixels as a linear combination of the input LR pixels and the relevant cluster centroid. In particular, the fuzzy clustering technique approximately classifies the given image pixels into different types firstly, assuming there exist a set of fuzzy rules concluding each type. Then, for each type of image pixels, a non-linear mapping consisting of a set of fuzzy rules is learned. A number of learning methods can be adopted for learning the relevant parameters in the fuzzy rules, such as the LSE, evolutionary algorithms. Amongst the various learning methods, the ANFIS has been proven to be a popular tool for generate a set of effective fuzzy rules through an adaptive neural network structure, this ANFIS based ISR approach is developed in this thesis and will be detailed described later in Chapter 5.

2.4 Summary

In this chapter, a number of background techniques have been introduced, forming the fundamental knowledge required to develop the work of this thesis. In particular, the basic concepts of ANFIS are reviewed in the first place. The typical fuzzy rule interpolation techniques are systemically outlined. These FRI techniques include the original KH approach and the popular T-FRI approach with Mamdani-type rules, and several existing TSK based rule interpolation approaches. Finally, the background knowledge of image super resolution is briefly introduced, and some recent popular image super resolution methods are reviewed.

Based on the background material presented in this chapter, Chapter 3 studies the problem of how to extend the traditional fuzzy rule interpolation to ANFIS interpolation, resulting in a group rule based interpolation approach.

Chapter 3

ANFIS Interpolation - A Group Rule Interpolation Based Approach

ALTOUGH ANFIS models have been widely used in various real world problems, the way to drive an effective ANFIS model with sparse training data still remains a challenging problem, similarly to many other algorithms. As reviewed previously in Section 1.2, the transfer learning and the FRI techniques may provide potential solutions to this important practical issue. Inspired by the basic ideas of transfer learning (that the knowledge obtained from the source domain can be transferred and used in the target domain) and fuzzy rule interpolation (that a new fuzzy rule can be constructed by interpolating two neighbouring rules), this chapter focusses on the introduction of a transfer-learning based approach to fuzzy rule interpolation for ANFIS construction. It combines the aforementioned ideas underlying the two distinct approaches to learn ANFISs for situations where there is only limited training data available in the given target domain. Particularly, to construct a new ANFIS in a target domain, the proposed approach seeks the assistance of two neighbouring ANFISs that have been generated from two source domains (where sufficient training data is available).

Because the proposed approach is ANFIS based, it has the following features:

1. Traditional FRI methods are concerned with individual rule interpolation within one fixed problem area, and are designed to interpolate a rule for an unmatched

observation, with just one rule interpolated using its closest neighbouring rules. While ANFIS interpolation focuses on the interpolation of an entire fuzzy inference system for a certain area, using ANFISs learned in the neighbourhood areas.

2. Traditional FRI methods work with a sparse rule-base, from which the closest rules are selected. But in ANFIS interpolation problem, there is no existing explicit sparse rule-base for use.

With these differences, how to extend the traditional rule interpolation to ANFIS interpolation? In order to cope with this problem, this chapter proposes a novel group rule interpolation based approach for ANFIS interpolation, established by exploiting the relationships between a target domain and its neighbouring source domains. This process is aided by a rule dictionary that acts in the similar role as a sparse rule-base, and a locally linear embedding (LLE) algorithm [126] that captures the hidden relationship between target domain and source domains in a certain rule antecedent space and then translates it into the rule consequent space. This group based work shows a radical departure from the conventional approach of FRI, offering a new FRI mechanism with improved abilities in non-linear adaptation and rapid learning.

The rest of this chapter is structured as follows. Section 3.1 gives an overall description of the ANFIS interpolation problem and the proposed method. In Section 3.2, the proposed approach for ANFIS construction is described in detail. The complexity for implementing the proposed approach is analysed in Section 3.3. Experimental results are presented and analysed in Section 3.4. Finally, Section 3.5 concludes this chapter.

3.1 The Framework for the Proposed Approach

In this section, the main idea underlying the proposed approach and the overall procedures implementing the idea are presented. At the highest level, this approach can be stated as follows. Without losing generality, suppose that sparse training data (not sufficient for learning an effective ANFIS) in the target domain T is given, expressed in a collection of input-output pairs $\{(\mathbf{x}, y)\}$, and that two ANFISs (denoted

as $\mathscr{A}_1$ and $\mathscr{A}_2$ respectively) are already trained over the source domains S_1 and S_2. Then, the goal of this work is to generate a new ANFIS $\mathscr{A}_{int}$ over T, through an innovative way of group-based rule interpolation. Note that here, $\mathbf{x}$ stands for the vector of all input variables.

Algorithm 3.1: ANFIS Interpolation - A group rule interpolation based approach.

Input:
Two source ANFISs in source domains: $\mathscr{A}_1, \mathscr{A}_2$;
Sparse training data in target domain: $\{(\mathbf{x}, y)\}$;
Number of closest atoms: K
`// Stage 1:Rule Dictionary Generation`
Extract fuzzy rules $\{R_i\}$ from $\mathscr{A}_1$ and $\mathscr{A}_2$;
Construct antecedent part of rule dictionary D_a via Equation (3.3);
Construct consequent part of rule dictionary D_c via Equation (3.4);
`// Stage 2:Intermediate ANFIS Construction`
Divide $\{(\mathbf{x}, y)\}$ into C clusters, C is decided by Equation (3.5);
for *each cluster centre* $c^{(k)}$ **do**
    Select K closest atoms in D_a;
    Compute weights $w^{(k)}$ for chosen atoms using Equation (3.8) and (3.9);
    Create rule antecedent using chosen atoms with index set $\mathscr{K}$ in D_a by Equation (3.11);
    Create rule consequent using atoms in D_c with the same index set $\mathscr{K}$ by Equation (3.12);
end
Integrate all interpolated rules to form the intermediate ANFIS $\mathscr{A}'$;
`// Stage 3:ANFIS Fine-tuning`
Use sparse training data $\{(\mathbf{x}, y)\}$ to fine tune $\mathscr{A}'$.
Output:
Interpolated ANFIS in target domain: $\mathscr{A}_{int}$.

To be concise, the overall algorithm implementing the proposed approach is shown in Figure 3.1, with detailed steps summarised in Algorithm 3.1. The entire process of constructing an effective ANFIS with sparse data over target domain involves three stages:

1. Rule dictionary generation,
2. Intermediate ANFIS construction,
3. ANFIS fine-tuning.

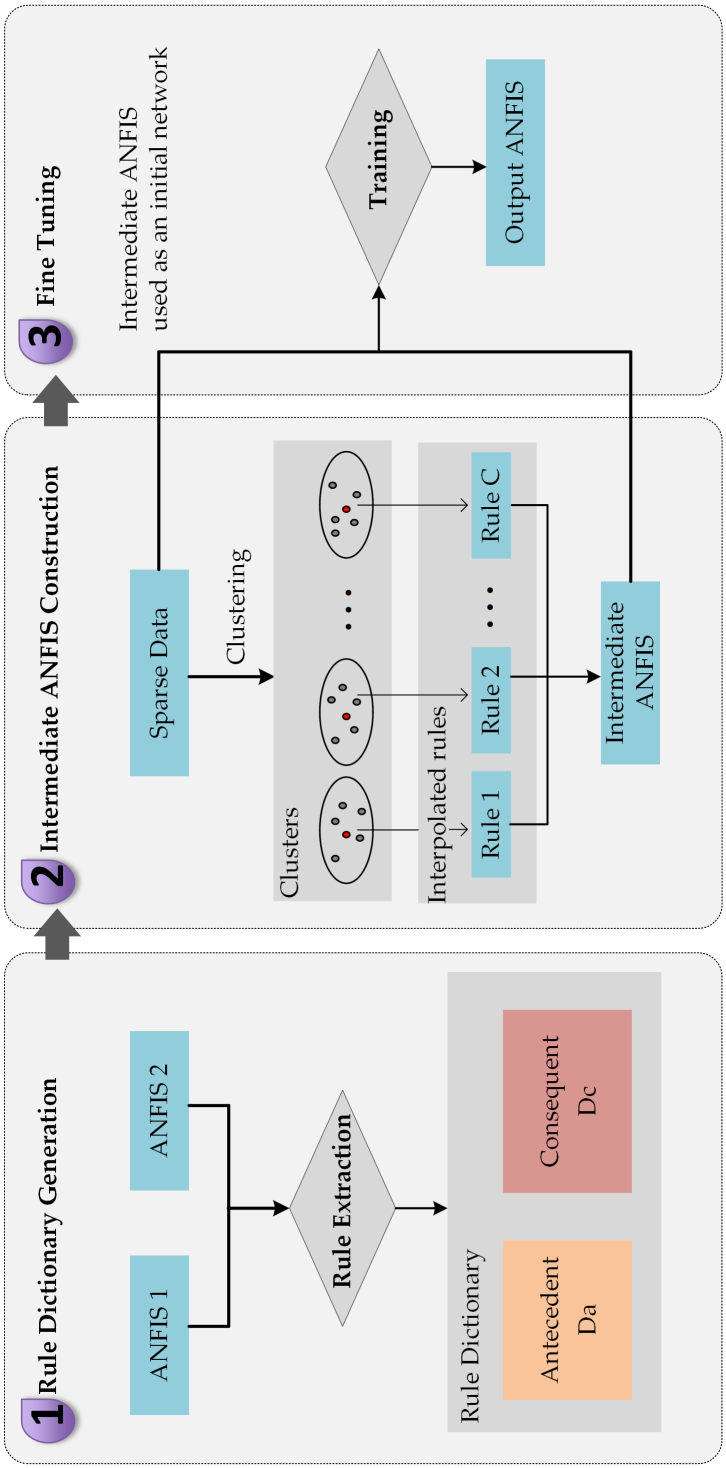


Figure 3.1: Flowchart of proposed approach

In stage 1, a rule dictionary is firstly generated by separating and reorganising rules extracted from source ANFISs $\mathscr{A}_1$ and $\mathscr{A}_2$. After that, in stage 2, an intermediate ANFIS is interpolated by the following procedure: a) clustering the sparse data of target domain into C clusters; b) interpolating a new rule for each cluster using LLE; and c) integrating all the newly generated rules in a network (to form the intermediate ANFIS). Finally in stage 3, the sparse training data is reused to refine the resulting intermediate ANFIS through retraining. The specifications for these three stages are further described in the following section.

3.2 Implementation of Proposed Approach

This section details the three stages that have been outlined in the framework of the proposed approach, including the rule dictionary generation stage, the intermediate ANFIS construction stage and the ANFIS fine-tuning stage.

3.2.1 Rule dictionary generation

To support interpolation of (groups of) fuzzy rules that are to be subsequently used for building an intermediate ANFIS, a rule dictionary (RD) is firstly constructed. Such a dictionary consists of an antecedent unit D_a and a consequent unit D_c, which are designed as two separate memories devised to respectively store collected rule antecedent parts and consequent parts that are extracted from given ANFISs. In general, it can be assumed that $\mathscr{A}_1$ and $\mathscr{A}_2$ consist of n_1 and n_2 rules respectively. Thus, the extracted rules can be expressed in the following format:

$$R_i^{\mathscr{A}_1} : if\ x_1\ is\ A_{i1}^{\mathscr{A}_1}\ and \ldots and\ x_m\ is\ A_{im}^{\mathscr{A}_1},\ \ then\ y_i = \sum_{j=0}^{m} p_{ij}^{\mathscr{A}_1} x_j \tag{3.1}$$

$$R_i^{\mathscr{A}_2} : if\ x_1\ is\ A_{i1}^{\mathscr{A}_2}\ and \ldots and\ x_m\ is\ A_{im}^{\mathscr{A}_2},\ \ then\ y_i = \sum_{j=0}^{m} p_{ij}^{\mathscr{A}_2} x_j \tag{3.2}$$

where m denotes the number of input variables, and $p_{i0}^{\mathscr{A}_t}, t \in \{1,2\}$ is a coefficient within the linear combination in a consequent part (with a set value of $x_0 = 1$ for the representation to meet the eye).

The rule dictionary $D = \{D_a, D_c\}$ is generated by separating and reorganising the rule antecedents and rule consequents of the aforementioned rules. In particular, $D_a \in R^{m \times n}$, consisting of all the rule antecedent parts:

$$\begin{aligned} D_a =& [d_1^a \; d_2^a \; \cdots \; d_n^a] \\ =& \begin{bmatrix} A_{1,1}^{\mathscr{A}_1} & A_{2,1}^{\mathscr{A}_1} & \cdots & A_{n_1,1}^{\mathscr{A}_1} & A_{n_1+1,1}^{\mathscr{A}_2} & \cdots & A_{n,1}^{\mathscr{A}_2} \\ A_{1,2}^{\mathscr{A}_1} & A_{2,2}^{\mathscr{A}_1} & \cdots & A_{n_1,2}^{\mathscr{A}_1} & A_{n_1+1,2}^{\mathscr{A}_2} & \cdots & A_{n,2}^{\mathscr{A}_2} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ A_{1,m}^{\mathscr{A}_1} & A_{2,m}^{\mathscr{A}_1} & \cdots & A_{n_1,m}^{\mathscr{A}_1} & A_{n_1+1,m}^{\mathscr{A}_2} & \cdots & A_{n,m}^{\mathscr{A}_2} \end{bmatrix} \end{aligned} \tag{3.3}$$

where each column $d_i^a = [A_{i,1}^{\mathscr{A}_t} \; A_{i,2}^{\mathscr{A}_t} \cdots A_{i,m}^{\mathscr{A}_t}]^T, t \in \{1,2\}$, (each $A_{i,j}^{\mathscr{A}_t}, j = 1,2,\ldots,m$, contains a fuzzy set value for a given input variable within a certain rule) forming an atom of the dictionary unit D_a. n_1 is the number of rules in $\mathscr{A}_1$, n_2 is the number of rules in $\mathscr{A}_2$, and $n = n_1 + n_2$ denotes the number of atoms in the rule dictionary. Similarly, the consequent unit $D_c \in R^{(m+1)\times n}$, consisting of the consequent parts of each and every rule (of the source domains), which can be expressed by

$$\begin{aligned} D_c =& [d_1^c \; d_2^c \; \cdots \; d_n^c] \\ =& \begin{bmatrix} p_{1,0}^{\mathscr{A}_1} & p_{2,0}^{\mathscr{A}_1} & \cdots & p_{n_1,0}^{\mathscr{A}_1} & p_{n_1+1,0}^{\mathscr{A}_2} & \cdots & p_{n,0}^{\mathscr{A}_2} \\ p_{1,1}^{\mathscr{A}_1} & p_{2,1}^{\mathscr{A}_1} & \cdots & p_{n_1,1}^{\mathscr{A}_1} & p_{n_1+1,1}^{\mathscr{A}_2} & \cdots & p_{n,1}^{\mathscr{A}_2} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ p_{1,m}^{\mathscr{A}_1} & p_{2,m}^{\mathscr{A}_1} & \cdots & p_{n_1,m}^{\mathscr{A}_1} & p_{n_1+1,m}^{\mathscr{A}_2} & \cdots & p_{n,m}^{\mathscr{A}_2} \end{bmatrix} \end{aligned} \tag{3.4}$$

with each atom $d_i^c = [p_{i,0}^{\mathscr{A}_t} \; p_{i,1}^{\mathscr{A}_t} \; p_{i,2}^{\mathscr{A}_t} \cdots p_{i,m}^{\mathscr{A}_t}]^T, t \in \{1,2\}$.

3.2.2 Intermediate ANFIS construction

An intermediate ANFIS is a set of new rules interpolated with the assistance of the rule dictionary created as above. The construction of an intermediate ANFIS is completed by three sub-steps: clustering sparse training data; interpolating rules; and integrating interpolated rules.

3.2.2.1 Clustering sparse training data

The sparse training data given in the form of $\{(\mathbf{x}, y)\}$ within the target domain are firstly partitioned into C clusters on a variable by variable basis by using K-means

clustering algorithm [105]. Clustering is utilised in order to minimise the derivation of any redundant rules, so that similar training data which belong to one cluster will (eventually) only lead to one rule. It is implemented in a 'variable by variable' fashion. That is, clustering is iteratively applied to all training data according to the first variable of the domain (with variables ordered in any preferable order), and then, for each resulting cluster, it is applied again to the data within the current cluster according to the second variable, etc. Following this 'variable by variable' clustering strategy, the resulting clusters will cover all possible regions of the input variables. The number of clusters C, which is also the expected number of rules in the intermediate ANFIS, is decided by Equation (3.5) below:

$$C = \prod_{j=1}^{m} \lfloor \frac{n_1^{(j)} + n_2^{(j)}}{2} \rfloor \tag{3.5}$$

where $n_1^{(j)}$ is the number of fuzzy sets of the jth variable in $\mathscr{A}_1$, $n_1 = \prod_{j=1}^{m} n_1^{(j)}$; and similarly, $n_2^{(j)}$ is that of $\mathscr{A}_2$, $n_2 = \prod_{j=1}^{m} n_2^{(j)}$. Note that here Equation (3.5) is not looking for the smallest C.

3.2.2.2 Interpolating rules

With respect to the resulting centroid of each cluster, a single newly interpolated rule is generated by exploiting the locally linear embedding algorithm (LLE) [126]. This is due to the recognition that LLE is one of the most classical and promising manifold learning methods, which is originally developed for tackling dimensionality reduction problems and has now been widely utilised in performing different types of machine learning task (e.g., [22, 70]). LLE generates a neighbourhood preserving mapping between a D-dimensional data space and a d-dimensional data space, assuming that the data of these two spaces lie on or near the same manifold. Without losing generality, suppose that each data point X_i in the D-dimensional data space is expected to be reconstructed by its K nearest neighbours $\{X_j\}$, and that the reconstruction error is measured by the following cost function:

$$\varepsilon(w) = |X_i - \sum_j w_j X_j|^2 \tag{3.6}$$

where the weight w_j modifies the contribution of the jth neighbour to the current data point X_i.

For each data point X_i in the D-dimensional data space, LLE involves the following implementation steps in an effort to construct the corresponding data point Y_i in the d-dimensional data space (thereby reducing the data dimensionality). Firstly, find the K nearest neighbours $\{X_j\}$ to X_i in D-dimensional data space. Secondly, compute the weights $\{w_j\}$ of the selected neighbours by minimising the cost function Equation (3.6) subject to: 1) that X_i is reconstructed only from its neighbours, while setting $w_j = 0$ if X_j does not belong to the set of neighbours; 2) that $\sum_j w_j = 1$. Finally, compute the corresponding data point Y_i regarding X_i in d-dimensional data space using weights $\{w_j\}$ and corresponding neighbours $\{Y_j\}$ regarding X_j such that $Y_i = \sum_j w_j Y_j$. Further details regarding this algorithm can be found in [126].

Similar to the LLE algorithm, the rule antecedent space and the rule consequent space are seen as the two spaces lying on the same manifold, and a new rule is interpolated for each cluster following the three steps, as outlined in Figure 3.2. Here, the first two steps are conducted in the antecedent RD space, while the third step is conducted in the consequent RD space. These three steps are described as follows.

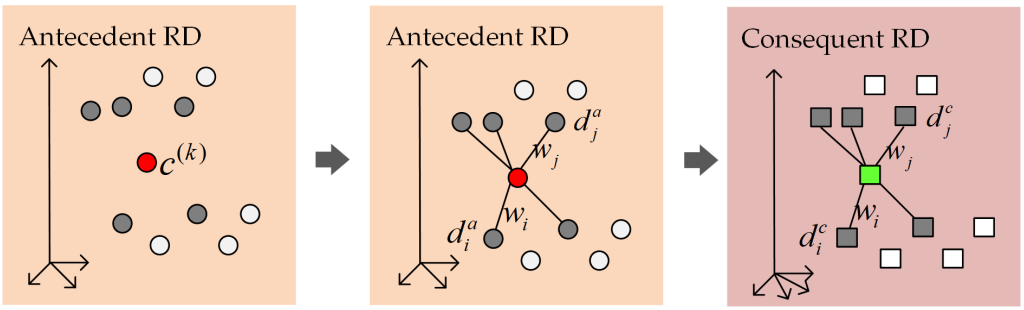


Figure 3.2: Illustrative implementation of single rule interpolation: (a) Choosing K closest neighbours. (b) Calculating weights of chosen neighbours. (c) Generating new rule.

A. Choosing the K closest atoms

To interpolate a new rule for a given cluster, the first step is choosing K closest atoms in D_a. For each cluster C_k, its centroid regarding the m antecedent attributes is denoted by $c^{(k)} = (c_1, c_2, \cdots, c_m)^T$. With the previously obtained antecedent rule dictionary D_a, the K closest atoms (a column in D_a denotes an atom) to $c^{(k)}$ are selected using the Euclidean distance metric (though any other distance metrics can be utilised as an alternative if preferred):

$$d_i = d(d_i^a, c^{(k)}) = \sqrt{\sum_{j=1}^{m} d(A_{ij}^{\mathscr{A}_t}, c_j)^2} \tag{3.7}$$

where d_i is the distance between d_i^a and $c^{(k)}$, $d(A_{ij}^{\mathscr{A}_t}, c_j) = |Rep(A_{ij}^{\mathscr{A}_t}) - c_j|, t \in \{1,2\}$. $Rep(A_{ij}^{\mathscr{A}_t})$ stands for the representative value of the fuzzy set $A_{ij}^{\mathscr{A}_t}$. m is the number of input variables. The K atoms $\{d_i^a\}$ in D_a which have the smallest distances are chosen as the closest neighbours, whose index set is denoted by $\mathscr{K}$. That is, $\forall i \in \mathscr{K}$, $max_i\{d(\{d_i^a\}, c^{(k)})\} < d(\{d_j^a\}, c^{(k)}), j \notin \mathscr{K}$ and $|\mathscr{K}| = K$.

B. Calculating the construction weights

Based on the obtained closest atoms $\{d_i^a | i \in \mathscr{K}\}$, the aim of step 2 is to find the best construction weights that indicate the relative significance of each selected atom. This process can be seen as the reconstruction of $c^{(k)}$ using $\{d_i^a\}$. Thus, the weights required for rule interpolation can be obtained by minimising the following reconstruction error:

$$w^{(k)} = \min_{w^{(k)}} ||c^{(k)} - \sum_{i \in \mathscr{K}} Rep(d_i^a) w_i^{(k)}||^2, \; s.t. \; \sum_{i \in \mathscr{K}} w_i^{(k)} = 1 \tag{3.8}$$

where $w_i^{(k)}$ is the relative weighting of d_i^a as compared to the rest, $Rep(d_i^a) = [Rep(A_{i,1}^{\mathscr{A}_t}) \; Rep(A_{i,2}^{\mathscr{A}_t}) \cdots Rep(A_{i,m}^{\mathscr{A}_t})]^T$, $t \in \{1,2\}$. Essentially it is a constrained least square optimisation problem which has the following solution:

$$w^{(k)} = \frac{G^{-1}\mathbf{1}}{\mathbf{1}^T G^{-1} \mathbf{1}} \tag{3.9}$$

where $G = (c^{(k)}\mathbf{1}^T - X)^T (c^{(k)}\mathbf{1}^T - X)$ is a Gram matrix, $\mathbf{1}$ denotes a column vector of ones, and X denotes an $m \times K$ matrix whose columns are the chosen atoms $\{d_i^a | i \in \mathscr{K}\}$.

C. Generating the new rule

Fundamentally speaking, FRI techniques are developed by performing similarity-based analytic reasoning, assuming that if the rule conditions are similar, then the consequents should also be similar. Reflecting this intuitive presumption in the view of LLE is that the underlying assumption for generating a new rule is: its antecedent part and consequent part lie on the same manifold. That is, if the centroid $c^{(k)}$ can be represented as a linear combination of the K selected atoms indexed by $\mathscr{K}$ in the antecedent rule dictionary D_a, then the corresponding consequent can be expressed

as the linear combination of the atoms (whose locations are also indexed by $\mathscr{K}$) in the rule consequent dictionary D_c. Based on this (practically working) presumption, a newly interpolated rule $R_k^{\mathscr{A}'}$ in response to cluster C_k is generated by applying the weight $w^{(k)}$ on both the antecedent and the consequent part, with the following format:

$$R_k^{\mathscr{A}'} : if\ x_1\ is\ A_{k1}^{\mathscr{A}'}\ and \ldots and\ x_m\ is\ A_{km}^{\mathscr{A}'},\ \ then\ y_k = \sum_{j=0}^{m} p_{kj}^{\mathscr{A}'} x_j \tag{3.10}$$

where the antecedent part is obtained by

$$A_{kj}^{\mathscr{A}'} = \sum_{i \in \mathscr{K}} w_i^{(k)} A_{ij}^{\mathscr{A}_t},\ j = 1, 2, \cdots, m,\ k = 1, 2, \cdots, C. \tag{3.11}$$

$t \in \{1, 2\}$, and similarly the consequent part is obtained by

$$p_{kj}^{\mathscr{A}'} = \sum_{i \in \mathscr{K}} w_i^{(k)} p_{ij}^{\mathscr{A}_t},\ j = 0, 1, 2, \cdots, m,\ k = 1, 2, \cdots, C. \tag{3.12}$$

3.2.2.3 Integrating the rules

The above completes the process of producing a newly interpolated rule. By integrating all such interpolated rules, the intermediate ANFIS $\mathscr{A}'$ results. This is implemented by putting all the newly generated rules into an ANFIS network, which is an inverse operation of extracting rules from a given network. A simple and generic example is given here to show this procedure.

Suppose that there are two input variables x_1 and x_2, and that a small number of training data associated with each variable is divided into two clusters. Thus, the sparse training data can be divided into $2 \times 2 = 4$ clusters as follows:

Cluster 1: $\{(x_1)_1, (x_2)_1\}$
Cluster 2: $\{(x_1)_1, (x_2)_2\}$
Cluster 3: $\{(x_1)_2, (x_2)_1\}$
Cluster 4: $\{(x_1)_2, (x_2)_2\}$

where $\{(x_1)_u, (x_2)_v\}$ represents the cluster including the u-th portion of x_1 and the v-th portion of x_2, with $u, v \in \{1st, 2nd\}$, e.g., $\{(x_1)_1, (x_2)_1\}$ represents the cluster

including the first portion of x_1 and the first portion of x_2. From these clusters, the following four rules are newly generated according to the above three steps:

Rule 1: If x_1 is A_1, x_2 is B_1, then $y = p_{11}x_1 + p_{12}x_2 + p_{10}$
Rule 2: If x_1 is A_1, x_2 is B_2, then $y = p_{21}x_1 + p_{22}x_2 + p_{20}$
Rule 3: If x_1 is A_2, x_2 is B_1, then $y = p_{31}x_1 + p_{32}x_2 + p_{30}$
Rule 4: If x_1 is A_2, x_2 is B_2, then $y = p_{41}x_1 + p_{42}x_2 + p_{40}$

These rules form the specification for a new (intermediate) ANFIS network to be constructed, by assigning the premise parameters $\{A_q, B_r\}$, $q, r \in \{1, 2\}$ in Layer 1, and the consequent parameters $\{p_{ij}\}$ in Layer 4, as shown in Figure 3.3.

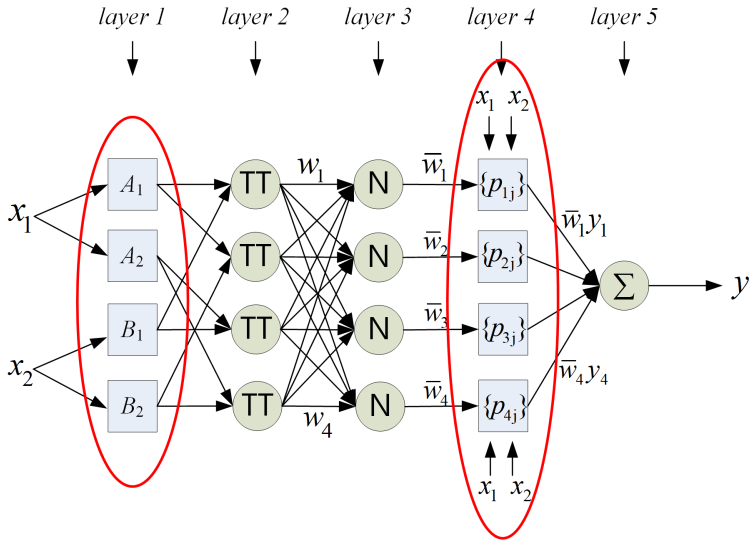


Figure 3.3: Integrating newly generated rules as an (intermediate) ANIFS

3.2.3 ANFIS fine-tuning

In the final stage, the intermediate ANFIS $\mathscr{A}'$ is employed as an initial network here to train the final interpolated ANFIS $\mathscr{A}_{int}$, using the sparse training data provided. The training procedure is basically the same as the traditional ANFIS training algorithm summarised in Section 2.1, with just one exception as described below.

The traditional procedure for ANFIS learning takes 'zeros' or 'random values' as the initial parameters of the network. This is practically acceptable for problems where

a large amount of training data is available. However, this is a rather inefficient strategy under the data shortage situations as for the present consideration. In general, a good initial setting is very important to the performance of a learning algorithm. Thus, instead of using random initials, the parameters embedded in the interpolated intermediate ANFIS are herein utilised to populate the initial network setting. In so doing, the final interpolated ANFIS in the target domain is one that is obtained through a fine-tuning procedure over the intermediate ANFIS that is derived from source domains. As such, the entire ANFIS construction can be performed with little training data.

3.3 Complexity Analysis

The time complexity of Algorithm 3.1 is estimated here. There are totally three stages listed in Algorithm 3.1. As Stage 3 is the same as that for conducting traditional ANFIS training, only the complexity of the procedures regarding rule interpolation (i.e., that of Stages 1 and 2) is addressed. As indicated before, for computational simplicity, triangular fuzzy sets with three characteristic points each are used for implementation. Note that the following notations are employed for the complexity analysis:

m : number of antecedent attributes

n : number of fuzzy rules in rule dictionary

K : number of chosen closest rules

C : number of clusters in training data

N : number of sparse training data points

In Stage 1 (rule dictionary generation, lines 1-3), the main task is to extract all the parameters from the given ANFIS networks (in the source domains). All the parameters of an ANFIS appear in either the first layer or the forth layer, with the premise parameters in layer one and the consequent parameters in layer four. Premise parameters are fuzzy sets, each of which contains three sub-parameters (say, a_0, a_1, and a_2), so the number of premise parameters is $3m$. Similarly, the

number of consequent parameters is $m+1$. Thus, the time complexity for extracting one rule is $O(4m+1)$, and the time complexity for computing the entire Stage 1 is $n \times O(4m+1) = O(4mn+n)$.

In Stage 2 (intermediate ANFIS construction, lines 4-11), line 4 for clustering using K-means algorithm takes $O(NCm)$. The sequence of lines 6-9 repeats C times. Particularly, line 6 involves two operations: 1) computing the Euclidean distances, of a complexity $O(n)$; and 2) sorting the distances, of a complexity $O(n^2)$. Then, in line 7, the weight is calculated once for each chosen rule, thereby being of a complexity $O(K)$. Finally, lines 8-9 take a complexity of $O(4m+1)$, in which the weighted average is calculated once for each parameter. Thus, the time complexity for Stage 2 is estimated to be $O(NCm) + C \times [O(n) + O(n^2) + O(K) + O(4m+1)] = O(NCm) + O(Cn^2)$.

Together, the overall time complexity for group rule interpolation bar that required by Stage 3 (for running the standard procedure to perform ANFIS fine-tuning with a small number of training data) is estimated to be $O(4mn+n) + O(NCm) + O(Cn^2) = O(NCm) + O(Cn^2)$. Compared with the standard ANFIS learning method, the complexity of ANFIS interpolation is slightly larger, but it is practically doable, since usually m and C are fixed in advance and both n and N are not a very large number.

3.4 Experimentation

Both synthetic and real world data are considered in the experiments to qualitatively and quantitatively evaluate the proposed ANFIS interpolation approach. Section 3.4.2 validates the approach by looking into two synthetic function modelling cases, and in Section 3.4.3, the effectiveness of the proposed approach in dealing with real world situations is shown. Section 3.4.4 discusses the model parameters, and examines the robustness of the proposed approach in response to the use of different amounts of training data.

3.4.1 Experimental setup

For all experiments carried out, triangular membership functions are employed due to their popularity and simplicity. The number of selected closest atoms is

empirically set to 3 unless otherwise stated (and a further investigation into the potential impact of different settings for this value will be reported towards the end of this section). To reflect the capability of the proposed approach in handling different data, both normalised and unnormalised data are investigated. In particular, for the experiments on synthetic data, the original data without normalisation is used despite that the data involves significantly skewed distributions over different magnitudes. In the experiments on the problem involving real world data, the input variables are normalised to $[0, 1]$.

The RMSE (Root-Mean-Squared Error) index is chosen to evaluate the performance of different ANFISs on both the source and the target data. Particularly, the RMSE measured from the source ANFIS $\mathscr{A}_1$ on the source domain data S_1 is denoted by $E_{\mathscr{A}_1(S_1)}$; that from $\mathscr{A}_2$ on S_2 by $E_{\mathscr{A}_2(S_2)}$; and that from $\mathscr{A}_1$ and $\mathscr{A}_2$ on the target domain data T by $E_{\mathscr{A}_1(T)}$ and $E_{\mathscr{A}_2(T)}$ respectively. These RMSEs are computed as below:

$$E_{\mathscr{A}_t(*)} = \sqrt{\frac{\sum_{k=1}^{N_*}(g_k^* - \mathscr{A}_t(x_k^*))^2}{N_*}} \tag{3.13}$$

where N_* is the number of the testing data points $\{x_k^*\}$ in the domain S_1, S_2 or T; g_k^* is the relevant ground truth of the kth data point; $\mathscr{A}_t(x_k^*), t \in \{1, 2, ori, int\}$ stands for the output value of an ANFIS on the data point x_k^*. Obviously, smaller RMSEs indicate better performance.

An original ANFIS developed using only the sparse training data in target domain [66] is also generated here for comparison, denoted by $\mathscr{A}_{ori}$. The RMSE of $\mathscr{A}_{ori}$ on the testing data T is denoted as $E_{\mathscr{A}_{ori}(T)}$, and the RMSE of the interpolated ANFIS $\mathscr{A}_{int}$ using the proposed approach on T is denoted by $E_{\mathscr{A}_{int}(T)}$.

3.4.2 Experiments on synthetic data

Two numerical functions are utilised here to test the proposed approach for working on highly non-linear one and two dimensional data. Further experiments are also included to show the case where absolutely no training samples are available for the target domain.

3.4.2.1 One dimensional input

As the first illustrative experimentation, a one-dimensional input function is used, which is generated by sampling the non-linear function as given below:

$$y = \frac{sin(2x)}{e^{\frac{x}{5}}} \tag{3.14}$$

where $x \in [0, 9]$, the shape of this function is plotted in Figure 3.4(a).

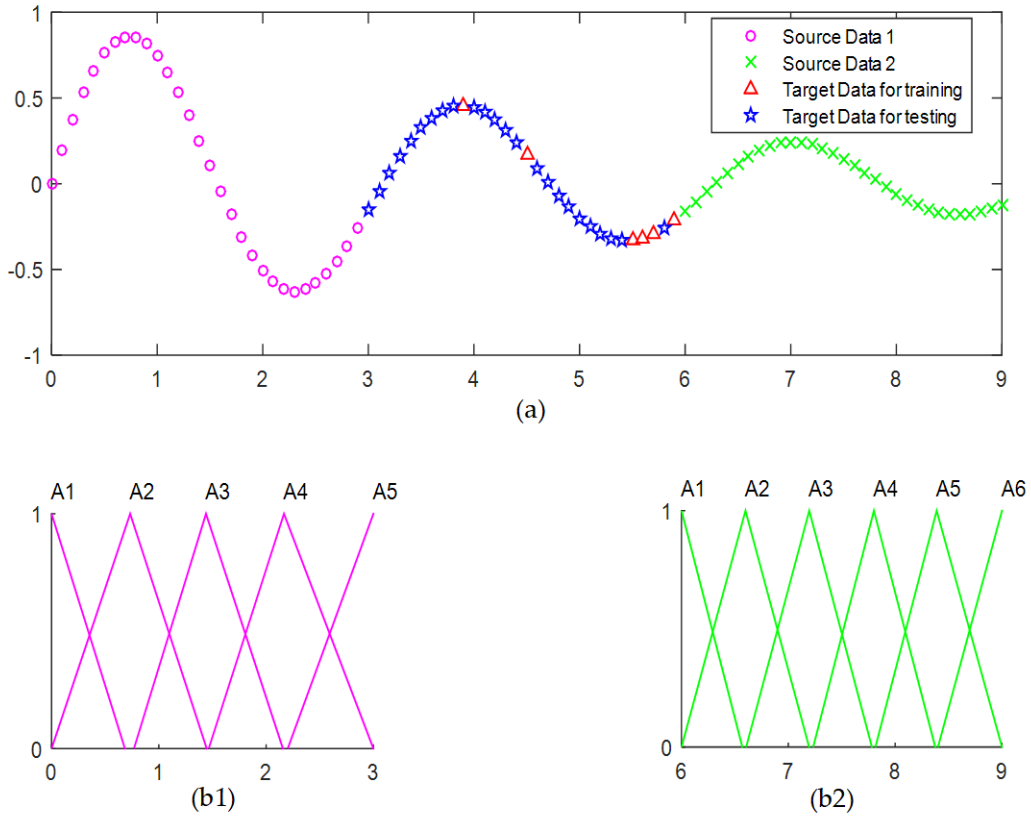


Figure 3.4: Illustration of data and source ANFISs. (a) Source data and target data used. (b) Membership functions in each source ANFIS.

A. Illustrative example

In this experimentation, to illustrate the proposed approach, the input domain $x \in [0, 9]$ is divided into three parts, representing two source domains and one target domain. As indicated previously, samples read off the underlying non-linear function are treated as the data as depicted in Figure 3.4(a). In particular, the left

part $x \in [0,3]$ is used as the first source domain for training the first source (or given) ANFIS $\mathscr{A}_1$, and the right part $x \in [6,9]$ is used as the second source domain for training the second source ANFIS $\mathscr{A}_2$. The middle part data $x \in [3,6]$ forms the target domain, which is divided into two sub-parts, with a small portion (20% randomly taken from the target domain, one fold training data is shown in triangles as an example for illustration) as the sparse training data (which is to artificially simulate the situation where little target domain data is available), with the rest used as testing data (80%, shown in pentagrams).

The two source ANFISs are trained using the standard method for ANFIS model learning as outlined in Section 2.1. The membership functions involved are illustrated in Figure 3.4(b). Here, the numbers of membership functions are set to 5 and 6 for $\mathscr{A}_1$ and $\mathscr{A}_2$, respectively. This is to demonstrate a case that is more complex than the usual (where quantity spaces tend to be defined with an equal number of fuzzy sets), making the experiments more challenging.

Rule dictionary generation: The rules embedded within the trained source ANFISs $\mathscr{A}_1$ and $\mathscr{A}_2$, form the atoms of the rule dictionary. Table 3.1 lists the rule dictionary constructed by extracting rules from trained ANFISs $\mathscr{A}_1$ and $\mathscr{A}_2$. There are totally 11 rules in the rule dictionary, with 5 from $\mathscr{A}_1$ and 6 from $\mathscr{A}_2$.

Table 3.1: Atoms in generated rule dictionary

Rule	Source ANFIS	Antecedent (a_0, a_1, a_2)	Consequent
1	$\mathscr{A}_1$	$(-0.725, 0.045, 0.684)$	$1.263x - 0.044$
2	$\mathscr{A}_1$	$(-0.040, 0.733, 1.446)$	$0.141x + 0.760$
3	$\mathscr{A}_1$	$(0.768, 1.440, 2.157)$	$-0.609x + 1.057$
4	$\mathscr{A}_1$	$(1.465, 2.166, 2.924)$	$-0.075x - 0.451$
5	$\mathscr{A}_1$	$(2.201, 2.874, 3.625)$	$0.658x - 2.151$
6	$\mathscr{A}_2$	$(5.400, 6.002, 6.570)$	$0.250x - 1.665$
7	$\mathscr{A}_2$	$(5.999, 6.599, 7.194)$	$0.195x - 1.127$
8	$\mathscr{A}_2$	$(6.605, 7.196, 7.782)$	$-0.098x + 0.936$
9	$\mathscr{A}_2$	$(7.234, 7.801, 8.375)$	$-0.223x + 1.765$
10	$\mathscr{A}_2$	$(7.815, 8.389, 9.012)$	$-0.083x + 0.531$
11	$\mathscr{A}_2$	$(8.401, 8.983, 9.600)$	$0.120x - 1.205$

Intermediate ANFIS construction: A group of new rules are interpolated in the target domain to form the intermediate ANFIS. The sparse training data in target domain is firstly clustered into C clusters. The number of clusters C (also the number of rules in the new ANFIS) is decided by Equation (3.5). In this illustrative example, $m = 1, n_1^{(1)} = 5, n_2^{(1)} = 6$, so $C = \lfloor (5+6)/2 \rfloor = 5$. The first column of Table 3.2 shows the 5 resulting centroids. Take the first centroid ($c = 3.90$) for example, a new rule is generated following the three steps (shown in the first row of Table 3.2): 1) choosing K closest neighbours with $K = 3$, the 5th, 4th and 6th atoms of the rule dictionary are chosen as shown in Table 3.2; 2) calculating the relevant weights for chosen atoms, resulting in weights $w_1 = 0.313, w_2 = 0.295, w_3 = 0.392$; 3) generating a new rule by averaging the chosen closest rules, $R' = w_1 R_5 + w_2 R_4 + w_3 R_6$, the antecedent and the consequent parts of the newly generated rule are shown in the last two columns of Table 3.2. Repeating this process for all the 5 centroids, the intermediate ANFIS results.

Table 3.2: Generated new rules for all cluster centroids

Centroids	Choosen atoms	weights	new rules	
			Ant	Con
3.90	(5, 4, 6)	0.313, 0.295, 0.392	(3.237, 3.891, 4.572)	$0.282x - 1.459$
4.50	(6, 5, 7)	0.264, 0.524, 0.212	(3.851, 4.489, 5.159)	$0.452x - 1.805$
5.55	(6, 7, 5)	0.374, 0.404, 0.222	(4.932, 5.549, 6.169)	$0.318x - 1.555$
5.70	(6, 7, 5)	0.389, 0.431, 0.179	(5.086, 5.700, 6.312)	$0.299x - 1.519$
5.90	(6, 7, 5)	0.411, 0.468, 0.121	(5.293, 5.902, 6.505)	$0.273x - 1.472$

ANFIS fine-tuning: By combining all the new rules, an intermediate ANFIS is constructed with 5 fuzzy rules. From this, the final output ANFIS is obtained by retraining it using the given data. The resultant interpolated ANFIS is shown in Table 3.3.

B. Experimental results

5×5-fold cross validation (5-FCV) is utilised to evaluate the performance of different approaches. Note that conventional 10×10-fold cross validation is not adopted here due to the extremely small number of samples for training, especially for the target domain. Note also that here the 5-FCV takes one piece of data for training while the other four pieces of data for testing, which is slightly different

Table 3.3: Rules in Interpolated ANFIS $\mathscr{A}_{int}$

Rule Number	Antecedent (a_0, a_1, a_2)	Consequent
1	$(3.237, 3.891, 4.572)$	$0.117x + 0.029$
2	$(3.852, 4.490, 5.159)$	$0.027x + 0.006$
3	$(4.934, 5.579, 6.167)$	$-0.081x - 0.012$
4	$(5.085, 5.696, 6.313)$	$-0.062x - 0.009$
5	$(5.288, 5.903, 6.504)$	$-0.0038x - 2.94 \times 10^{-5}$

with the standard 5-FCV where one piece of data is for testing and four pieces of data is for training. The mean and standard deviation values of different methods compared are listed in Table 3.4, and the visual results of 5 folds within the total 5×5 folds are shown in Figure 3.5.

Table 3.4: Experimental results on one-dimensional function

	Mean ± Standard deviation
$E_{\mathscr{A}_1(S_1)}$	0.003 ± 0.000
$E_{\mathscr{A}_2(S_2)}$	0.0006 ± 0.000
$E_{\mathscr{A}_1(T)}$	0.287 ± 0.013
$E_{\mathscr{A}_2(T)}$	0.255 ± 0.014
$E_{\mathscr{A}_{ori}(T)}$	0.121 ± 0.067
$E_{\mathscr{A}_{int}(T)}$	0.089 ± 0.039

By examining the experimental results in both Figure 3.5 and Table 3.4, it can be seen that the two source ANFISs $\mathscr{A}_1$ and $\mathscr{A}_2$ perform quite well in their corresponding source domains (as $E_{\mathscr{A}_1(S_1)}$ and $E_{\mathscr{A}_2(S_2)}$ are quite small). However, these two ANFISs do not work in the target domain (as $E_{\mathscr{A}_1(T)}$ and $E_{\mathscr{A}_2(T)}$ are rather large). This is of course, not surprising since they have been trained using the data for the source domains in the first place. Yet, if the standard training method is used to build an ANFIS for the target domain, it does not work well either, as the performance of the original ANFIS $\mathscr{A}_{ori}$ is poor. Again, this may be expected due to data shortage of the target domain. Fortunately, with the assistance of its two neighbouring source ANFISs, the interpolated ANFIS $\mathscr{A}_{int}$ improves the results significantly. As shown in the middle part of Figure 3.5(a)-(e), its outcome is much closer to the ground truth.

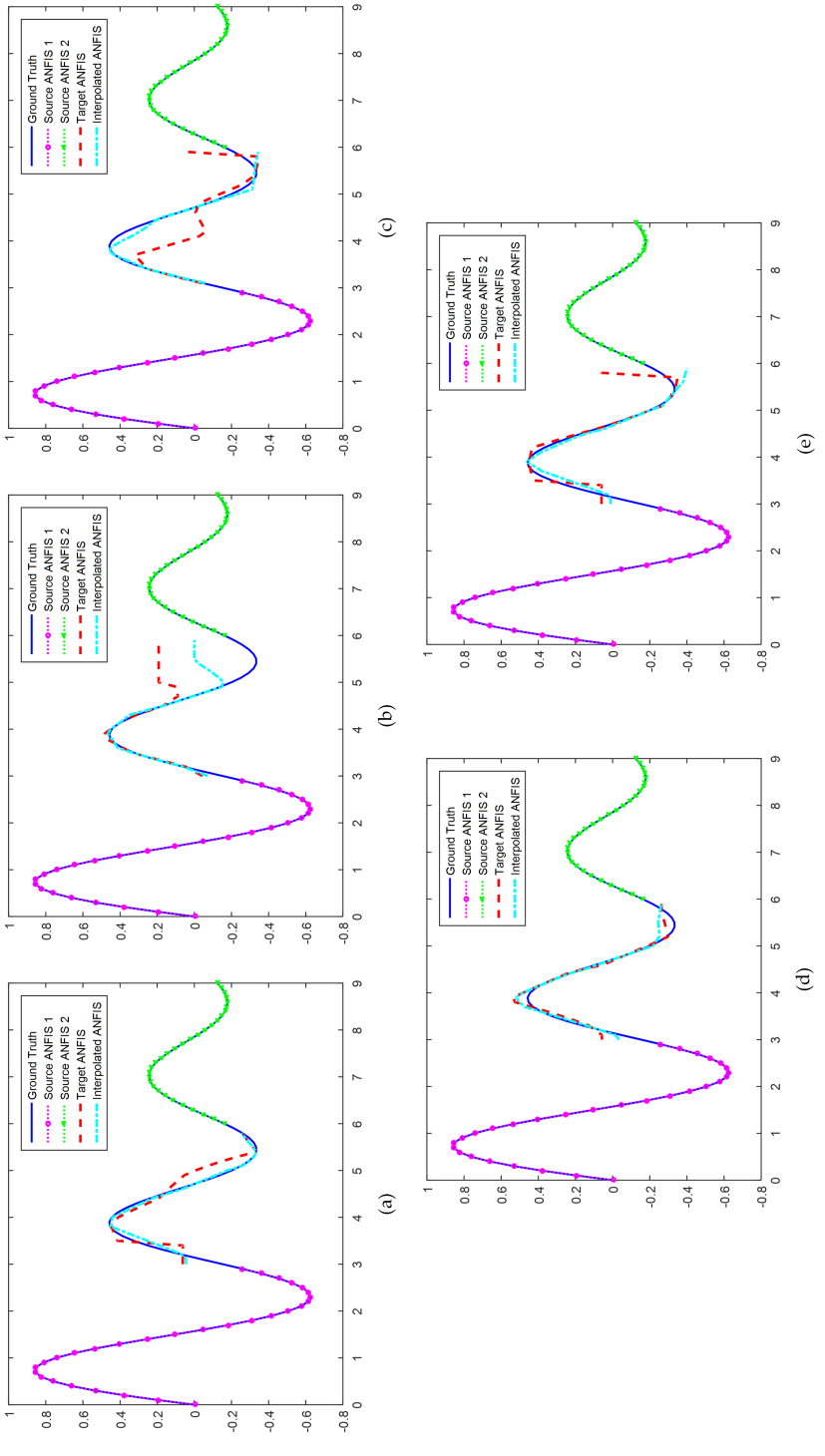


Figure 3.5: Five folds taken from 5×5-fold cross validation for one-dimensional non-linear function approximation.

3.4.2.2 Two dimensional input

The two-dimensional synthetic data used in this experimentation is sampled from the following function:

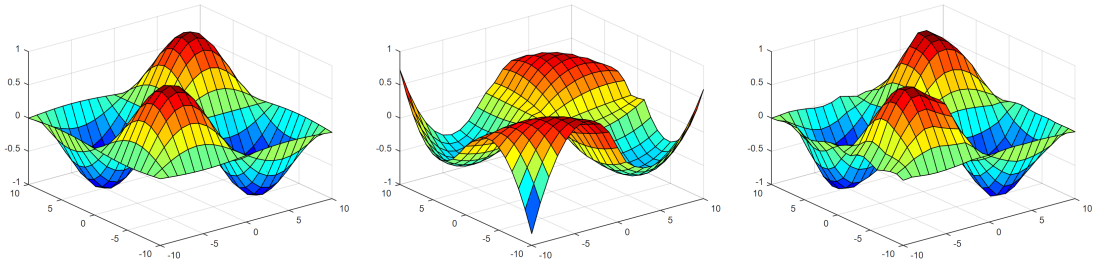
$$y = sin(\frac{x_1}{\pi})sin(\frac{x_2}{\pi}) \tag{3.15}$$

where $x_1 \in [-30, 30], x_2 \in [-10, 10]$. Figure 3.6(a) displays the shape of this function in one period. Similar to the previous experiments on one-dimensional data, the entire underlying domain is divided into 3 parts. Without any particular bias, this is implemented with respect to the first variable x_1. That is, the region covered by $x_1 \in [-30, -10], x_2 \in [-10, 10]$ (with $step = 1$ in each dimension, totally 441 data points sampled from the function) forms the first source domain, the region by $x_1 \in [10, 30], x_2 \in [-10, 10]$ (the amount of sampled data is the same as the first source domain, 441 data points in total) forms the second source domain, and the region covered by $x_1 \in [-10, 10], x_2 \in [-10, 10]$ forms the target domain. There are also 441 data points in the target domain, in which 88 data points (20%) are used for training while 353 points (80%) are used for testing. Experiments on splitting the data with respect to the second variable are also carried out, with similar results achieved as to be reported later.

Table 3.5: Experimental results on two-dimensional function

	Mean ± Standard deviation	
	splitting x_1	splitting x_2
$E_{\mathscr{A}_1(S_1)}$	0.024 ± 0.000	0.023 ± 0.000
$E_{\mathscr{A}_2(S_2)}$	0.023 ± 0.000	0.023 ± 0.000
$E_{\mathscr{A}_1(T)}$	1.176 ± 0.029	1.079 ± 0.030
$E_{\mathscr{A}_2(T)}$	1.331 ± 0.022	1.586 ± 0.035
$E_{\mathscr{A}_{ori}(T)}$	0.372 ± 0.063	0.489 ± 0.264
$E_{\mathscr{A}_{int}(T)}$	$\mathbf{0.070} \pm 0.016$	$\mathbf{0.103} \pm 0.087$

The middle column of Table 3.5 lists the results of 5-FCV for this two dimensional function approximation problem using different ANFISs. As an example, Figure 3.6 shows the visual result of one randomly picked fold, where Figure 3.6(a) displays the ground truth view of this two-dimensional function in target domain; Figure 3.6(b)



(a) (b) (c)

Figure 3.6: One fold of the two-dimensional function-approximation results: (a) Ground truth. (b) Result based on $\mathscr{A}_{ori}$. (c) Result based on $\mathscr{A}_{int}$.

illustrates the result based on the original ANFIS directly learned from the sparse training data; and Figure 3.6(c) is that of the interpolated ANFIS.

As reflected by these experimental outcomes, the result of the interpolated ANFIS is much more similar to the real view of the underlying highly non-linear two-dimensional function. Without the assistance of group rule interpolation, the outcome from running the ANFIS directly trained by the limited samples is rather different from the ground truth. Again, these results indicate that the ANFIS learned by the proposed interpolation method performs much better than the original ANFIS under the situations where only highly restricted training data is provided.

Note that the above results are obtained using source domains and target domain defined by splitting the domain of the first input variable x_1. However, similar results can also be obtained via specifying the source domains and target domain with regard to the second variable x_2 of this function, as listed in the third, i.e., the right-most column of Table 3.5.

Consider a more general situation where both input variables x_1 and x_2 are divided into three regions. As such there are totally nine subregions, as shown in Figure 3.7. This gives rise to a general case in which there may be more than two source ANFISs for use to interpolate an ANFIS within a given target domain. For instance, if insufficient training data appears in the central subregion of Figure 3.7, whilst an ANFIS is required to be constructed, then this will be a problem of interpolating multiple ANFISs from given ANFISs within the neighbouring source domains. It is straightforward to extend the proposed work to such a situation as the only difference is regarding rule dictionary generation, where the rules are extracted

from multiple source ANFISs. For this particular example, the result of taking the central subregion as target domain and the other 8 subregions as source domains can be computed, such that the average and standard deviation values of RMSE (over 5-FCV) for the original ANIFS $E_{\mathscr{A}_{ori}(T)}$ is 0.215 ± 0.178, whilst that for the interpolated ANFIS is $E_{\mathscr{A}_{int}(T)} = 0.177 \pm 0.081$, showing a remarkable improvement.

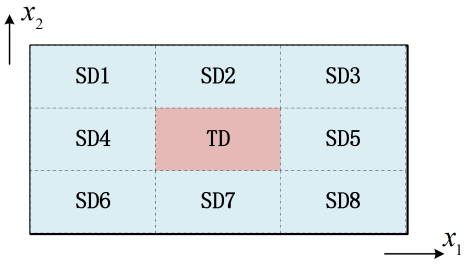


Figure 3.7: Function approximation with more than two source domains

3.4.2.3 Situation with no target training data

Consider a further situation where no target training data is available in the target domain, using the problem of approximating the function as expressed in Equation (3.14). In the area where $x \in [0.5, 2]$, this one dimensional non-linear function may be interpreted approximately as linear, as shown in Figure 3.8. Suppose that the data covered within the left part delimited by $x \in [0.5, 1]$ is used to train the first source ANFIS $\mathscr{A}_1$, and that the data within the right part $x \in [1.5, 2]$ is used to train the second source ANFIS $\mathscr{A}_2$. The data of the middle part $x \in [1, 1.5]$ is fully reserved for testing and hence, no training samples are available over the target domain. Since no target training data is provided, each test data point is seen as a centroid, and the new rule interpolated for this data point forms a special intermediate ANFIS containing just one rule, which is then directly used for inference without retraining. The visual result is also shown in Figure 3.8.

From the results shown, it can be seen that the proposed approach performs very well over this particular problem, even though no target training data is available. However, this should not be overly generalised since for the same underlying function, if the source domains are [1.5,2] and [2.5,3], and no information is present in support of the training of the target domain delimited by [2,2.5], the proposed approach will only produce an approximate model as depicted in Figure 3.9. This does not seem to

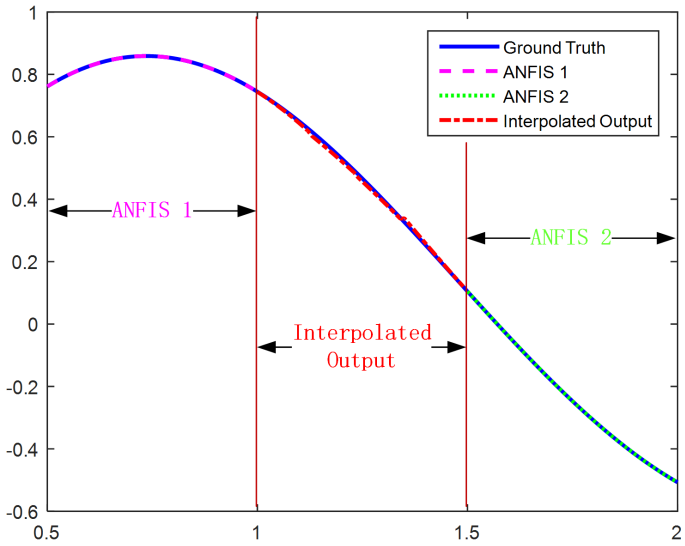


Figure 3.8: Function approximation with no target training data

perform well, due to the high non-linearity of the function to be approximated within this region. As shown in the target domain of this figure, the interpolated outcome is far away from the ground truth in the bottom area surrounding the minimum point, around which the function shape of target domain and source domains are totally different. Quantitatively, the mean value of RMSE is 0.0948, which is fairly large for this problem. Nevertheless, as previously demonstrated, once there are a small number of training samples provided for the target domain, the approach can result in an accurate ANFIS.

3.4.3 Experiments on real world benchmark data

Twelve benchmark datasets taken from the KEEL data repository [5] are used here to evaluate the performance of the proposed ANFIS construction algorithm on real world problems. The datasets used in this experimentation are summarised in Table 3.6.

The generation of the source domains and target domain is similar to the cases where the synthetic data are used, by dividing each dataset into three parts according to one of the input variables. For example, the Quake Dataset has three input variables: 'Longitude', 'Latitude' and 'Depth'. This dataset is divided into three sub-datasets using the values of the variable 'Longitude'. In particular, those instances

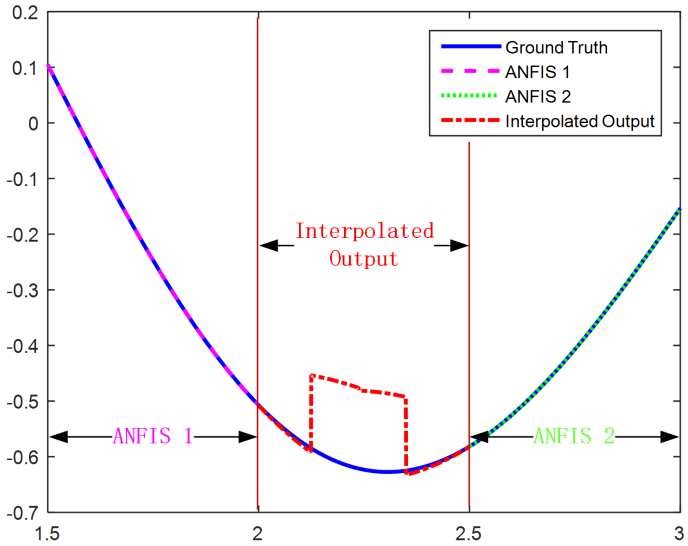


Figure 3.9: Performance with no target training data under non-linear situation

Table 3.6: Public dataset used

Dataset	**No.(Attributes)**	**No.(Instances)**
Diabetes	2	43
Plastic	2	1650
Quake	3	2178
Laser	4	993
AutoMPG6	5	392
Delta-ail	5	7129
Friedman	5	1200
Dee	6	365
Delta-elv	6	9517
AutoMPG8	7	392
Concrete	8	1030
Stock	9	950

with this variable value being smaller than -40 jointly form the first source domain S_1 (of a subtotal of 593 instances), instances with 'Longitude' value larger than 92 form the second source domain S_2 (1253 instances), and instances with 'Longitude' value between $[-40, 92]$ form the target domain T (332 instances). Similar to the experiments on synthetic data reported earlier, 20% (66 instances) of the data in the

target domain are used as the training data, with the remaining 80% (265 instances) used as the testing data. This is very different from the common choice that usually takes a majority of data (often 80% or 90%) for training, and the remaining minority of data for testing. Such a strategy makes the ANFIS construction a challenging problem as only very limited data is available for training. Thus, the proposed interpolation technique is introduced in an effort to generate an effective ANFIS model.

5×5-fold cross validation results of this experimentation are shown in Table 3.7. The average values of all twelve datasets are shown in the bottom row, with the best results shown in bold. As reflected by these results as per Table 3.7, a similar conclusion to what is previously learned from the experiments on synthetic data can be drawn. Both $E_{\mathscr{A}_1(T)}$ and $E_{\mathscr{A}_2(T)}$ are very large, indicating that the ANFISs trained in the source domains are not suitable for the regression problem in the target domain. Again, this is not surprising. However, those target domain models $\mathscr{A}_{ori}$ trained over the limited amounts of data perform significantly better than their counterparts. Since the the number of training data is so small for each of the twelve cases, the resultant ANFIS $\mathscr{A}_{ori}$ is still not very stable (as indicated by the large standard deviation value in Table 3.7), though its mean RMSE is significantly better. Despite this limitation, the interpolated ANFISs following the proposed approach remarkably minimise the inference error caused by data shortage, as the mean values and the standard deviation values of $E_{\mathscr{A}_{int}(T)}$ are much smaller than those of $E_{\mathscr{A}_{ori}(T)}$.

Apart from the RMSE values, it is also important and necessary to test the statistic significance of the interpolated ANFIS over the original ANFIS. The pairwise t-test ($p = 0.05$) is employed here for such statistic analysis. The p-values of t-test of the interpolated ANFIS ($\mathscr{A}_{int}$) over the original ANFIS ($\mathscr{A}_{ori}$) are listed in Table 3.8, with $p < 0.05$ representing that $\mathscr{A}_{int}$ performs significantly better than $\mathscr{A}_{ori}$. In this table, signs (v) and (*) denote statistically better or not respectively. As can be seen from the table, the p values are smaller than 0.05 for almost all datasets, indicating the statistical improvements made by the interpolated ANFIS. In t-test, smaller p value means better result. It can be noticed that the p value of the 'AutoMPG6' dataset is quite small, demonstrating the superior performance of the interpolated model.

Table 3.7: Experimental results on real world datasets

Datasets	Mean ± Standard deviation of ANFISs			
	$E_{\mathscr{A}_1(T)}$	$E_{\mathscr{A}_2(T)}$	$E_{\mathscr{A}_{ori}(T)}$	$E_{\mathscr{A}_{int}(T)}$
Diabetes	7.045±0.514	10.166±2.045	1.527±0.829	1.058±0.497
Plastic	5.824±0.128	8.654±0.173	2.003±0.246	1.843±0.105
Quake	0.563±0.014	1.624±0.063	1.264±0.641	0.542±0.248
Laser	71.142±0.687	18.514±0.438	12.087±4.739	3.757±2.726
AutoMPG6	17.672±1.082	18.781±0.886	12.770±1.431	5.120±1.038
Delta-ail	$2.36\times10^{-4}\pm1.41\times10^{-5}$	$4.76\times10^{-4}\pm5.52\times10^{-4}$	$3.53\times10^{-4}\pm1.09\times10^{-4}$	$1.78\times10^{-4}\pm1.06\times10^{-5}$
Friedman	5.583±0.120	4.571±0.091	5.177±0.633	3.004±0.409
Dee	3.059±0.080	5.614±0.201	1.017±0.339	0.870±0.225
Delta-elv	$0.0023\pm3.011\times10^{-5}$	$0.0074\pm7.229\times10^{-5}$	$0.0034\pm6.195\times10^{-4}$	$0.0020\pm1.690\times10^{-4}$
AutoMPG8	13.175±0.412	20.823±0.726	5.801±1.707	5.695±1.199
Concrete	85.764±2.633	28.918±0.778	27.228±10.894	14.058±2.813
Stock	37.357±0.797	26.129±0.484	3.843±2.080	2.314±0.803
Average	20.598±0.539	11.984±0.490	6.060±1.962	**3.189**±0.838

Table 3.8: p values of pairwise t-test results

Dataset	$\mathscr{A}_{int}$ v.s. $\mathscr{A}_{ori}$
Diabetes	0.0216 (v)
Plastic	0.0037 (v)
Quake	1.12×10^{-5} (v)
Laser	1.34×10^{-7} (v)
AutoMPG6	6.41×10^{-18} (v)
Delta-ail	1.90×10^{-8} (v)
Friedman	7.08×10^{-15} (v)
Dee	0.0065 (v)
Delta-elv	5.92×10^{-12} (v)
AutoMPG8	0.7832 (*)
Concrete	2.53×10^{-6} (v)
Stock	0.0013 (v)

3.4.4 Experiments with different settings

The parameter K controls the number of selected closest neighbours in running the LLE algorithm. In the experimental results reported above, this number is empirically set to be 3. In order to investigate the relationship between the parameter K and the experimental outcome, the performance over different K using the 'Quake' dataset is given, as shown in Figure 3.10. The conclusion is that for most cases, a smaller K will lead to better result, this finding is similar to what is established in [92] (though that piece of existing work is concerned with a weighted approach to FRI).

It may be expected that if the number of training data becomes smaller, the performance of the trained ANFINs will generally become worse. To verify this hypothesis with experimental investigation, Figure 3.11 shows the outomes under the condition where a different percentage of training data is employed over the 'Quake' dataset.

It can be induced from examining this figure that: 1) independent of what percentages of training data used, interpolated ANFISs outperform the original ANFISs; 2) the less training data is involved the more improvement the interpolated ANFIS makes over its counterpart which is trained just by the use of the sparse set of

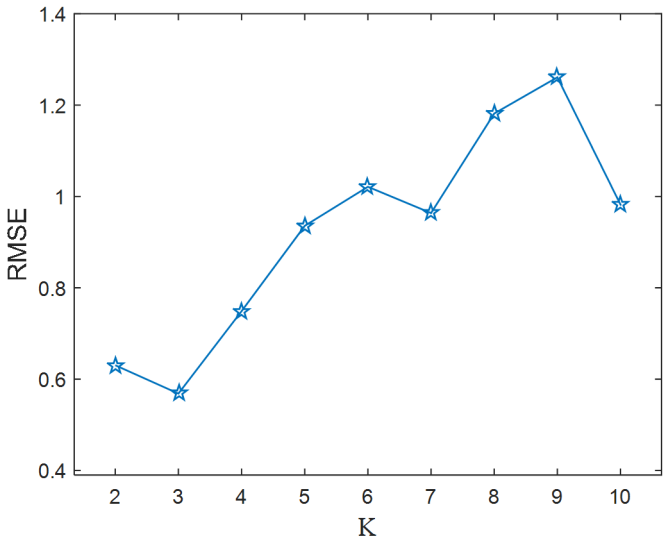


Figure 3.10: Performance vs. number of selected closest neighbours

training samples. It can be observed from the figure that when the percentage of training data reaches 90%, the performance of the two algorithms are close. This can be expected, because when sufficient training data is available, the superiority of the interpolated ANFIS is not obvious. An interesting observation is that as reflected by the triangular-marked line (which is the plot for interpolated ANFISs), the performance using 30% data is very close to that using 90% data. This is very different from the trend of the results attained using the ANFISs trained without the aid of interpolation. In summary, through the use of the proposed approach, a much improved inference outcome can be achieved while requiring much less training data.

3.5 Summary

How to construct an effective fuzzy inference system with insufficient training data is a practically important and challenging issue. This chapter has proposed a new ANFIS construction method through the use of group rule interpolation. To the best of our knowledge, this is the first time that the interpolation of ANFIS models is proposed (using two source ANFISs to assist the construction of the target one). It significantly differs from the general transfer learning methods in the literature where only one source domain is involved; in this work there are at least two source

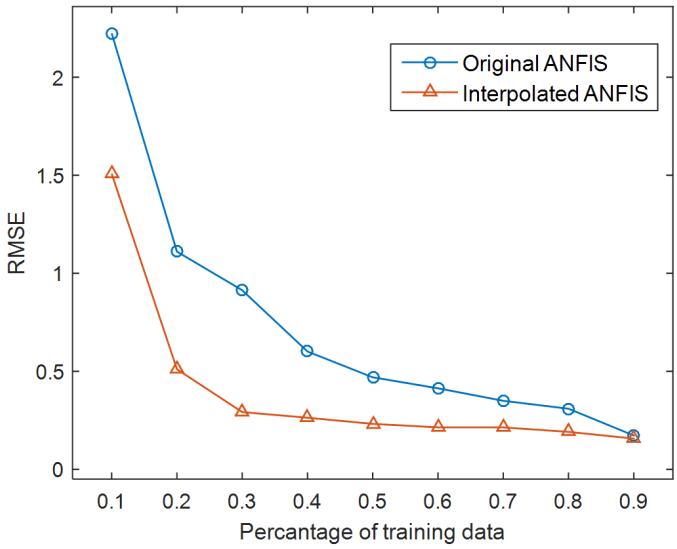


Figure 3.11: Performance vs. amount of training data

domains. The work effectively resolves the data shortage problem for training ANFISs in the target domain. Experiments on both synthetic data and real world data have been carried out, including variations of the experimental background settings. The results have consistently demonstrated that the proposed approach greatly improves the performance in learning ANFIS models for problems where only sparse training data is available.

In comparison with existing work on FRI involving the use of TSK-type representation, this wok exhibits the following distinct innovations:

1. As reflected above, rule interpolation is done at the level of a group of rules, instead of at the individual rule level as per the existing techniques. That is, unlike the existing FRI methods where only one intermediate rule is produced at a time, here by one run of the algorithm a group of rules are interpolated. This is because an ANFIS generally represents a set of TSK rules.

2. Learning is accomplished with the use of a dictionary purposefully introduced to facilitate ANFIS interpolation. In existing FRI techniques, there exists a sparse rule-base from which individual closest rules are directly selected for interpolation, but this does not apply to interpolation of a group of rules in a target domain. The rule dictionary is therefore designed for extracting rules from source ANFISs, acting as a sparse rule-base for the selection of the closest rules in the target domain.

3. Weights required for performing interpolation are computed differently. For instance, the active set algorithm [60] is employed in the existing work [31], which is an iterative method; whilst the LLE algorithm is utilised here, which is a one-step method without involving iterations. Different from traditional techniques that rely on the use of Euclidian distance measures to work, the proposed approach provides a theoretically well-formed and systematic method for calculating the weights that are required to construct an intermediate ANFIS during the interpolation process.

In this initial ANFIS interpolation approach, a group of fuzzy rules is interpolated. Following this, Chapter 4 will further optimise these interpolated rules with an evolutionary algorithm.

Chapter 4

ANFIS Interpolation - An Evolutionary Approach

Based on the aforementioned initial ANFIS interpolation approach, an improved ANFIS learning method is proposed in this chapter to deal with the problem of training data shortage. The proposal is to interpolate a number of candidate fuzzy rules first, forming an initial population which is modified via an evolutionary optimisation process subsequently.

In general, evolutionary computation [9, 10] offers a range of optimisation algorithms by analogy to natural evolution processes. Such algorithms perform optimisation with a set of chromosomes, which are able to search the problem hyperspace efficiently and effectively. Applying an evolutionary computation method to aid in building a fuzzy inference system injects learning capacity into the underlying fuzzy systems, which is commonly referred to as evolutionary fuzzy systems (EFS) in the literature (e.g., [7, 38, 39, 51, 129]). Considering genetic algorithms (GAs) being the most popular technique for use in developing such systems, the present work utilises a GA to implement the evolutionary process.

One of the key points of the proposed work is therefore, to examine how an ANFIS may be encoded using a specific chromosome representation. Without prejudgement, two alternative forms of representing fuzzy rules are considered in this chapter, using either individual rule based representation or group of rules (equivalently an entire

ANFIS) based representation to construct an initial population for further evolution. The initial population is iteratively updated through crossover and mutation operations, subject to the use of a fitness function, in order to determine whether a chromosome may enter into the next loop. The chromosomes of the highest fitness, namely rules or ANFISs with the best performance, will then be returned when the iterative process terminates. If the chromosomes represent individual rules, they collectively form the required ANFIS; otherwise, each returned chromosome is a learned ANFIS. Such a resulting ANFIS will be fine tuned to obtain the final improved ANFIS model, appropriate to deal with approximate inference in the target domain concerned. Both of the two chromosome representations are systemically evaluated and compared. Experimental results demonstrate that the evolutionary approach further improves the performance of the interpolated ANFIS while being competitive to those popular machine learning methods.

The remainder of this chapter is organised as follows. Section 4.1 reviews the basic knowledge of GA. Section 4.2 details the proposed evolutionary ANFIS interpolation approach. Section 4.3 discusses the complexity for implementing the proposed approach. Experimental results are discussed in Section 4.4. Finally, Section 4.5 concludes this chapter.

4.1 Genetic Algorithm

Inspired by observing natural evolution processes, evolutionary computation [9, 10] is in general, proposed to provide an effective and efficient way for searching optimal solution in poorly understood and irregular problem spaces. Typically, evolutionary algorithms work with a population of individuals, in which each individual may be one or a set of potential solutions in the solution space. Many evolutionary algorithms have been proposed, including genetic algorithms (GAs) [121, 145], genetic programming [1], and particle swarm optimisers [175]. Among these, GAs may be the most commonly adopted and hence, are utilised in this work also.

Mainly owing to the conceptual simplicity and computational effectiveness, GA becomes one of the most popular evolutionary algorithms. In a GA implementation, each individual of the population is encoded as a chromosome, which may be subsequently modified through crossover and mutation subject to a certain probability.

A fitness function is utilised to evaluate the performance of every chromosome within a population. The general implementation of a GA can be summarised in the following steps:

(1) Initialize the population;

(2) Perform crossover and mutation on the population;

(3) Calculate the fitness of each chromosome;

(4) Select a portion of chromosomes to construct a new population;

(5) Loop to step (2) until a certain stopping criterion is met.

Applying GAs for fuzzy systems modelling leads to techniques for building a form of genetic fuzzy learning systems. Typical approaches involve the development of procedures for: 1) Parameter tuning – By assuming that the structure of a fuzzy inference system is pre-defined, this procedure adapts the system's parameters [55] (such as those defining membership functions or the coefficients specifying the consequent functions in a TSK model) with respect to changes in the model input. 2) Rule selection – By encoding a rule-base as a fixed-length chromosome, this procedure aims to control the complexity of a fuzzy inference system, leaving less room for redundant, incorrect or badly defined rules to exist [64]. 3) Rule-base construction – By encoding both the parameters and the structure of a fuzzy system within each chromosome, this procedure performs parameter estimation and structure identification at the same time [37, 106] (although often at the cost of significantly increasing the problem complexity).

4.2 GA for ANFIS Interpolation

This section presents the proposed novel ANFIS interpolation approach, which is supported by a GA. The problem addressed in this chapter is the same with that in chapter 3, which can be outlined as follows: With only a small number of training data in the target domain T, expressed in the form of input-output pairs $\{(\mathbf{x}, y)\}$, ANFIS interpolation is to construct an effective ANFIS $\mathscr{A}_{int}$ over T, by interpolating two neighbouring ANFISs, $\mathscr{A}_1$ and $\mathscr{A}_2$, defined on two source domains S_1 and S_2, respectively. Figure 4.1 summarises the entire interpolation process, consisting of

three main stages: i) population initialisation via observation-guided interpolation of rules embedded in the source ANFISs, based on the method in Chapter 3; ii) interpolation via a GA over the initial population; and iii) ANFIS fine-tuning via the standard ANFIS learning method as described in Section 2.1 (using the given small number of training data in the target domain). The entire evolutionary process will of course depend upon how individuals are to be encoded in a population of chromosomes and how each chromosome may be evaluated. Details of this process are described below.

4.2.1 Chromosome representation: two strategies

How to encode a potential solution with a chromosome is a critical point in GAs. To reflect the fact that an ANFIS is essentially a set of fuzzy rules, there are two strategies that may be taken to represent an ANFIS model within a GA: 1) Encode each underlying rule of the ANFIS as a chromosome, and 2) Encode the entire ANFIS as a chromosome. Both types of chromosomes are introduced here.

4.2.1.1 Rule-based chromosome representation

This representation strategy encodes each rule of an ANFIS as a vector of rule parameters, in the form of (antecedent parameters, consequent parameters). For example, suppose that a rule involving m antecedent variables that take triangular fuzzy sets as values is expressed by:

$$\begin{aligned} &If\ x_1\ is\ A_1\ and\ x_2\ is\ A_2 \ldots and\ x_m\ is\ A_m, \\ &then\ y = p_0 + p_1 x_1 + p_2 x_2 + \cdots + p_m x_m \end{aligned} \tag{4.1}$$

where each fuzzy set A_i $(i = 1, 2, \cdots, m)$ contains three parameters (a_{i0}, a_{i1}, a_{i2}) that respectively denote the three vertices of the triangle. Then, the rule based chromosome can be expressed such that

$$((a_{10}, a_{11}, a_{12}), (a_{20}, a_{21}, a_{22}), \cdots, (a_{m0}, a_{m1}, a_{m2}),\ p_0, p_1, \cdots, p_m) \tag{4.2}$$

In general, using triangular fuzzy sets, if a rule contains m antecedent variables, there will be $3m$ premise parameters and $m+1$ consequent parameters. Thus, the length of the corresponding chromosome will be $3m + m + 1 = 4m + 1$.

4.2.1.2 ANFIS-based chromosome representation

In this representation, an entire ANFIS containing C rules is encoded as one chromosome. Suppose that a certain ANFIS contains m input variables $x_1, x_2, \cdots, x_m$, and that the ith variable may take one of the c^i triangular fuzzy sets $A_{i1}, A_{i2}, \cdots, A_{ic^i}$ as its value. Then, totally there will be $C = c^1 \times c^2 \times \cdots \times c^m$ rules captured in this ANFIS, and all these rules are collectively encoded as one vector like the following:

$$(rule\ 1, rule\ 2, \cdots, rule\ C) \tag{4.3}$$

where each single rule is coded exactly the same as Equation (4.2). In so doing, the length of an ANFIS based chromosome will be $C \times (4m+1) = \prod_{i=1}^{m} c^i(4m+1)$, which is much longer than that of a rule based one.

4.2.2 Population initialisation

An initial population, composed of a number of interpolated rules in the target domain, needs to be generated first, in order to start the evolutionary process. This is accomplished by adopting the group-based FRI mechanism of Chapter 3, over the given (sparse) training data, with the assistance of a rule dictionary constructed by extracting rules from the two source domain ANFISs $\mathscr{A}_1$ and $\mathscr{A}_2$ (as shown in Figure 4.1(b)). Details are as follows.

A rule dictionary $D = \{D_a, D_c\}$ with an antecedent part D_a and a consequent part D_c is generated firstly, to separately store collected rule antecedents and consequents, by extracting and then storing rules from the given source ANFISs. This process is the same with the rule dictionary generation procedure described in Section 3.2.1, so is omitted here.

4.2.2.1 Interpolating candidate rules

In order to form an initial population, a number of rules in the target domain are required. Traditional means for generating the initial population in a GA usually use randomly set parameters in the rules. Whilst this is practical for situations with sufficient training data it becomes a significant challenge for the current case, where

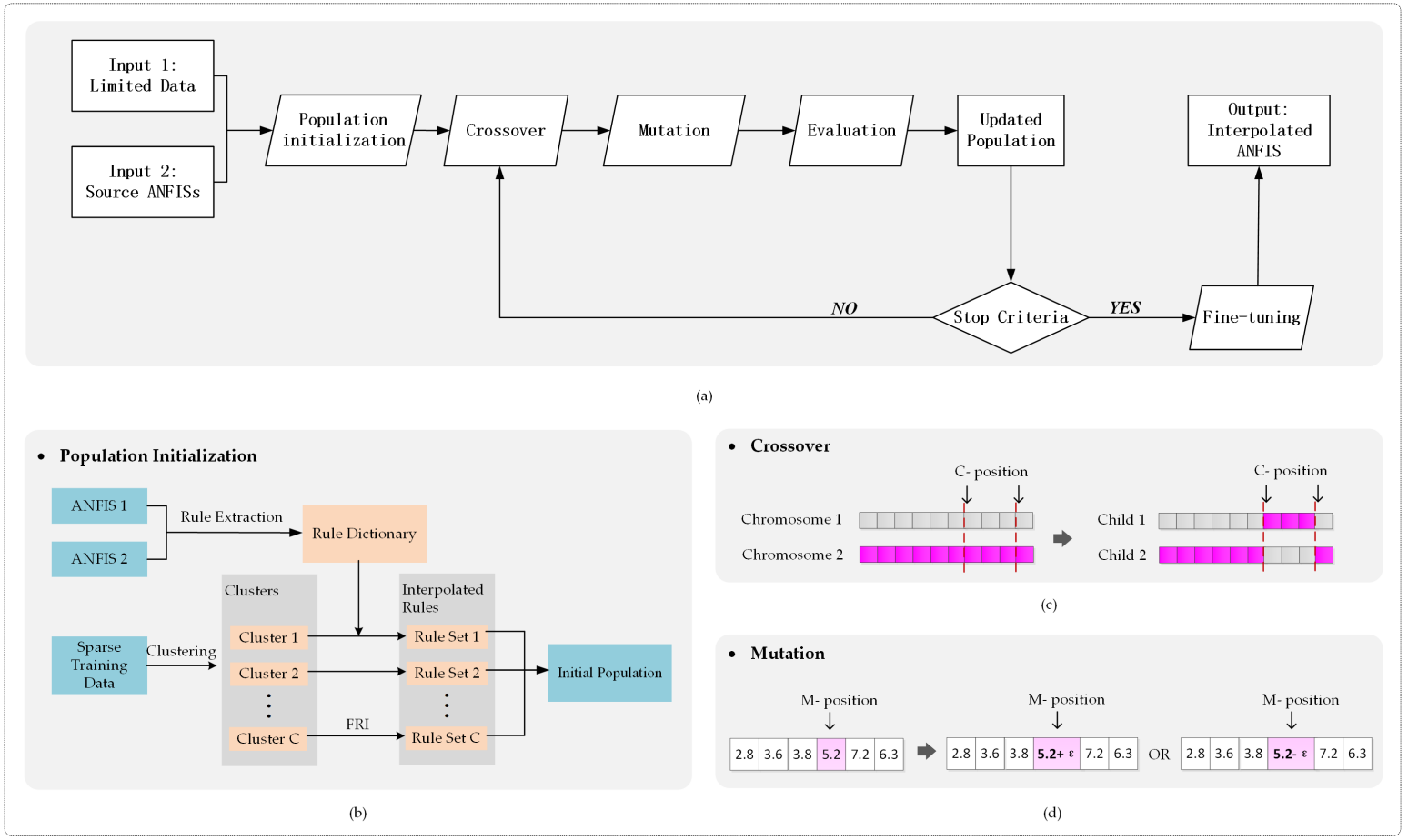


Figure 4.1: Flowchart of proposed approach. (a) Overall flowchart. (b) Population initialization process. (c) Crossover process. (d) Mutation process.

only highly restricted sparse data is available. To address this challenge, FRI is employed to interpolate rules in target domain, in an effort to set up an improved initial network, so as to be able to produce a more effective model with less training data. Given the above rule dictionary, a number of candidate rules in the target domain can indeed be obtained through interpolation by running the following procedure.

To start with, the given training data $\{(\mathbf{x}, y)\}$ in the target domain is divided into C clusters. C stands for the number of the rules in the ANFIS to be interpolated and is determined by Equation (3.5).

Having obtained the clusters, for each cluster, a set of candidate rules are interpolated to create the initial population. However, the number of training data in each cluster may be very different, and in certain clusters there may be just one datum. For extreme cases where the number of training samples is smaller than C, certain clusters are simply empty, covering no data at all, though such situations may be rare. Therefore, instead of just utilising the raw training data contained within the clusters, individual instances are also artificially generated in an effort to enrich the original sparse training data. For this, the centre of each cluster is used as the seed to generate more individuals. In particular, for those clusters without any training data, the seeds are set to be the same as their neighbouring clusters (of course, such initially identical settings will become different through the evolutionary process).

For each cluster C_k, the centroid is denoted by $c^{(k)} = (c_1, c_2, \cdots, c_m)^T$ with regard to the m attributes. Use $c^{(k)}$ as a seed, a number of artificial data (denoted by $\{O = (o_1, o_2, \cdots, o_m)^T\}$) can be generated by adding Gaussian white noise with the seed itself being the mean and a small value (δ) being the standard deviation for each attribute. This method borrows practical ideas often adopted in the field of electrical engineering. The number of the individual instances in each cluster is set subject to the constraint in which the sum of cluster sizes will be the size of the required initial population. For implementational simplicity, in this work, all clusters are set to be of an equal size. Such an individual generation process can be illustrated as per Figure 4.2.

From the resulting data enriched clusters, for each individual instance within a given cluster, a candidate rule can be generated through interpolation that involves the same steps with Section 3.2.2.2. Firstly, selecting K closest rules in rule dictionary

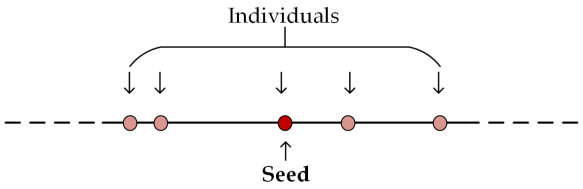


Figure 4.2: Individual generation from a seed.

by computing the Euclidian distance as defined by Equation (4.4), between the current individual O and every column of D_a:

$$d_i = d(d_i^a, O) = \sqrt{\sum_{j=1}^{m} d(A_{ij}^{\mathscr{A}_t}, o_j)^2} \tag{4.4}$$

where $d(A_{ij}^{\mathscr{A}_t}, o_j) = |Rep(A_{ij}^{\mathscr{A}_t}) - o_j|, t \in \{1,2\}$. The index set associated with the K selected columns $\{d_i^a\}$ in D_a is denoted by $\mathscr{K}$.

Next, with the obtained closest columns $\{d_i^a;\ i \in \mathscr{K}\}$, the candidate rule for the current individual can be interpolated as the weighted average of the selected rules. The weights required for rule interpolation are obtained by solving the optimization problem described as: $w = \min_w ||O - \sum_{i\in\mathscr{K}} Rep(d_i^a) w_i||^2,\ s.t.\ \sum_{i\in\mathscr{K}} w_i = 1$, where $Rep(d_i^a) = [Rep(A_{i,1}^{\mathscr{A}_t})\ Rep(A_{i,2}^{\mathscr{A}_t}) \cdots Rep(A_{i,m}^{\mathscr{A}_t})]^T$, $t \in \{1,2\}$, and w_i denotes the relative weighting of the column d_i^a. As mentioned previously in Section 3.2.2.2, the solution of this constraint optimisation problem is: $w^{(k)} = (G^{-1}\mathbf{1})/(\mathbf{1}^T G^{-1}\mathbf{1})$, where $G = (O\mathbf{1}^T - X)^T(O\mathbf{1}^T - X)$, $\mathbf{1}$ is a column vector of ones, and the columns of X are the selected $\{\{d_i^a\},\ i \in \mathscr{K}\}$. Following the principles of FRI which performs reasoning by analogy, the weights w_i derived for the antecedent part are applied to the consequent part to attain similarity. Thus, the newly interpolated rule for the current individual has the following format:

$$R: If\ x_1\ is\ A_1\ and \ldots and\ x_m\ is\ A_m,\ \ then\ y = \sum_{j=0}^{m} p_j x_j \tag{4.5}$$

where

$$\begin{cases} A_j = \sum_{i\in\mathscr{K}} w_i A_{ij}^{\mathscr{A}_t},\ j = 1,2,\cdots,m,\ t \in \{1,2\} \\ p_j = \sum_{i\in\mathscr{K}} w_i p_{ij}^{\mathscr{A}_t},\ j = 0,1,2,\cdots,m,\ t \in \{1,2\} \end{cases} \tag{4.6}$$

4.2.2.2 Generating initial population

By collecting all the interpolated candidate rules, the initial population will be set up. Depending on which of the two chromosome representation strategies is used, there are slightly differences when forming the initial population. Using rule based chromosome representation, all the candidate rules within one cluster form an initial sub-population for this cluster. Therefore, there are C initial sub-populations in total, each of which will initiate an independent evolutionary learning process, as shown in Figure 4.3. While using ANFIS based chromosome representation, all the candidate rules are collected to form a global initial population, as shown in the Figure 4.4. Therefore, there is just one initial population in this case.

4.2.3 Crossover and mutation

For an evolutionary process, from the initial sub-population or the entire population (depending upon which coding style is used for chromosome representation), a subset of chromosomes (of an even cardinality) are randomly selected to perform crossover with a pre-specified crossover probability (denotes by p_c). The chosen chromosomes are then, randomly paired up. For each (so-called parent) pair $(ch1, ch2)$, the standard 'two-point' crossover is applied, in which a start position and an end position are randomly determined and subsequently, the genes between them are exchanged. In so doing, two child chromosomes are generated per pair. This procedure is illustrated in Figure 4.1(c).

After crossover, mutation operation follows. Similar to the crossover operation, a subset of the (sub-)population is selected for mutation with a pre-determined probability (denotes by p_m). For each chosen chromosome, a mutation position is randomly generated. Then, the mutated chromosome is created by randomly adding or subtracting a small value ε to the gene at that position. The mutation procedure is illustrated in Figure 4.1(d).

Note that as with common approaches in the literature, a larger crossover probability and a smaller mutation probability are empirically assumed.

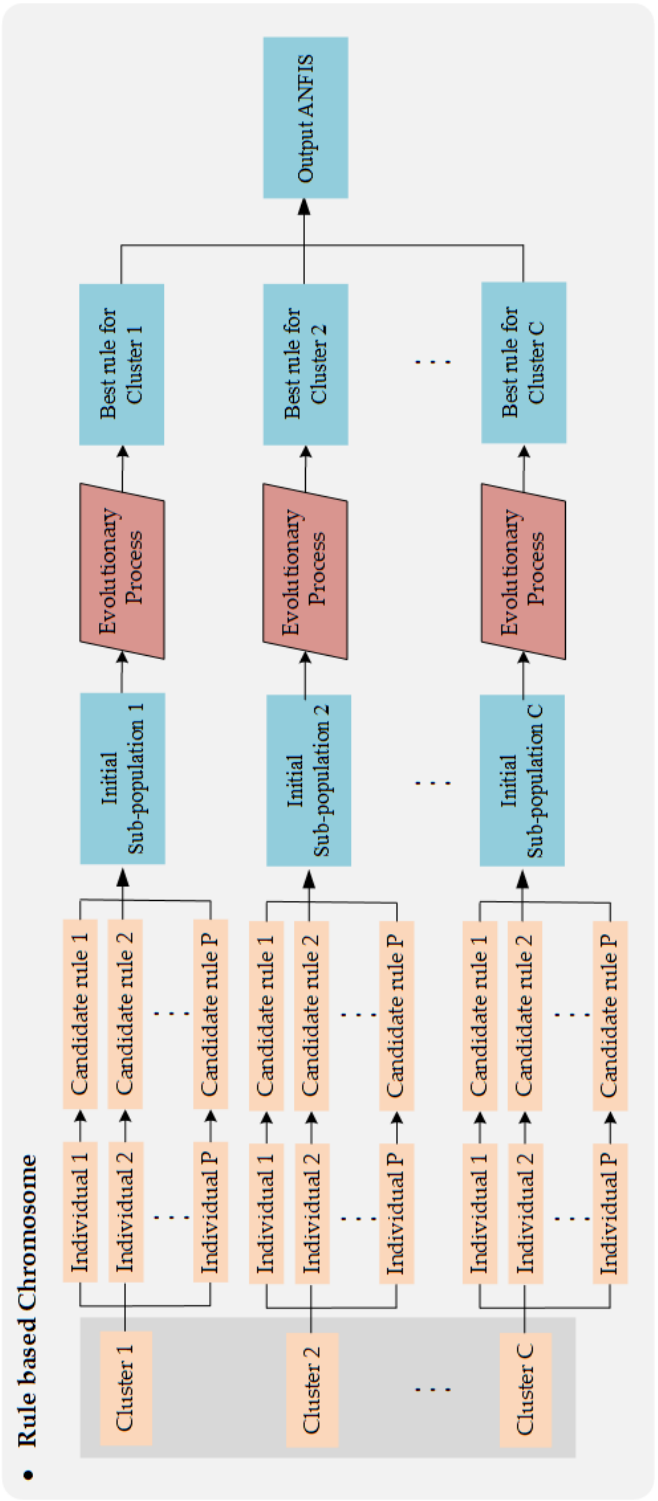


Figure 4.3: Rule based chromosome representation used in implementation.

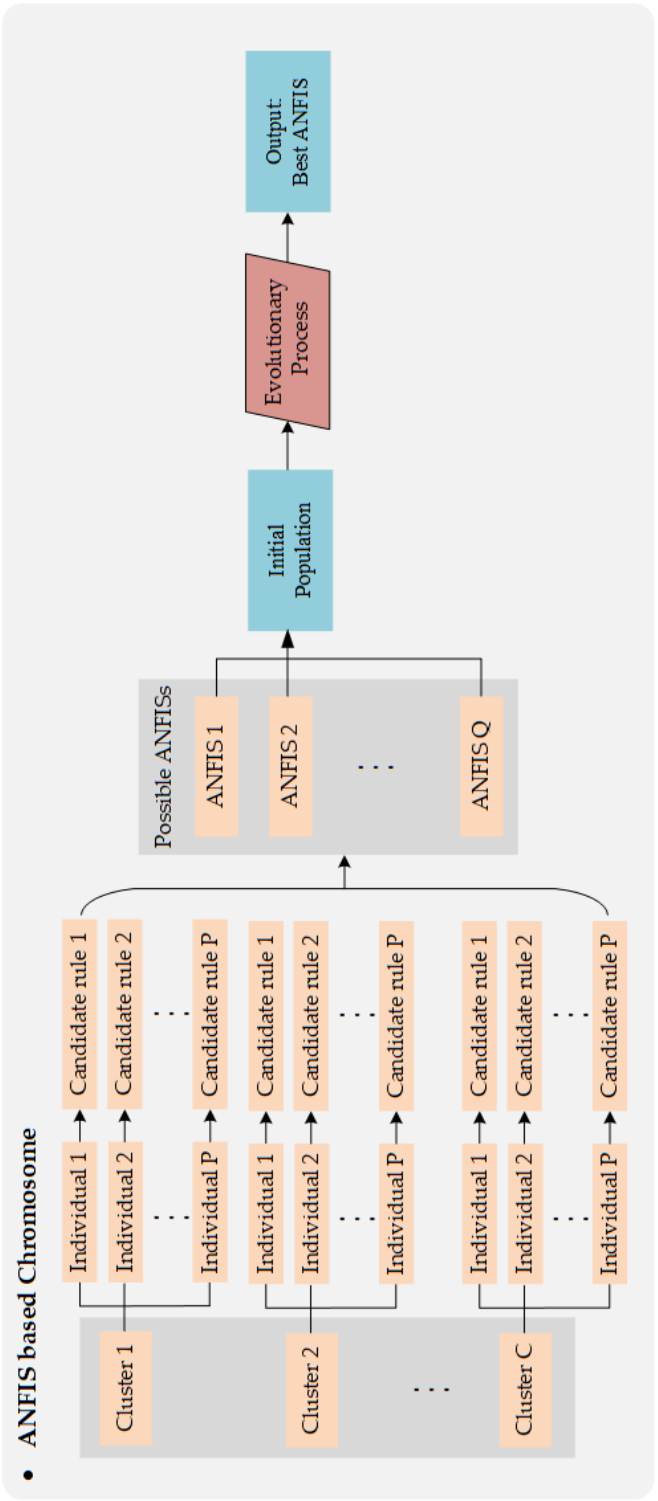


Figure 4.4: ANFIS based chromosome representation used in implementation.

4.2.4 Fitness function

Fitness function is used to evaluate the performance of the chromosomes so that better performers can be maintained to enter the next iteration. Designing an appropriate fitness function is important for any evolutionary algorithm. How to define a fitness function however, depends on the application problem at hand. For prediction and estimation tasks, functions that compute the mean square error or the absolute difference error are the most commonly adopted. In this work, the fitness function is implemented on the basis of the RMSE (Root-Mean-Squared Error), which is defined per chromosome as follows:

$$E = \sqrt{\frac{\sum_{i=1}^{N}(g_k - \mathscr{A}(x_k))^2}{N}} \tag{4.7}$$

where N is the number of training data, g_k and $\mathscr{A}(x_k)$ are the kth expected and estimated output over the evolutionary process, respectively. Suppose that the maximum error permitted is E_{max}, then such a fitness function may be specified by

$$F = E_{max} - E \tag{4.8}$$

For the present application, a meaningful fitness measure should be given for an entire ANFIS instead of a single rule. Thus, further consideration may be due, depending upon which chromosome representation is utilised. For ANFIS based chromosome representation, the fitness of each chromosome can be directly obtained using the fitness function above, as per Equation (4.8). However, if individual rule based chromosome representation is employed, the fitness of such a chromosome is obtained using a score table that evaluates and records the performance of every candidate rule, as per Table 4.1, where $score_{ij}$ denotes the evaluated (fitness) value of the jth candidate rule in the ith cluster and S is the size of candidates in each cluster.

Note that to capture and reflect the essence of the fitness of a candidate rule, computationally, $score_{ij}$ is calculated as the averaging performance of the jth candidate rule in the ith cluster across a certain number of randomly generated possible ANFISs. This is necessary because in dealing with practical problems, the number of possible ANFISs can be rather large. For example, suppose that there are 10 rules in the rule-base of an ANFIS, and that there are 5 candidates per underlying

Table 4.1: Score table

Candidate \ Cluster	1	2	⋯	C
1	$score_{11}$	$score_{21}$	⋯	$score_{C1}$
2	$score_{12}$	$score_{22}$	⋯	$score_{C2}$
⋮	⋮	⋮	⋱	⋮
S	$score_{1S}$	$score_{2S}$	⋯	$score_{CS}$

rule. Then, the number of possible ANFISs will be 5^{10}, which is a huge figure that requires substantial storage and computation power. Thus, in implementation, just a portion of the possible ANFISs are randomly chosen for evaluation. To balance effectiveness and efficiency, the number of chosen possible ANFISs, denoted by A_{max}, may be set empirically (and in this research, $A_{max} = 500$). This practice (of only a randomly selected portion of possible ANFIS chromosomes being used for evaluation) is common in the literature of applied evolutionary algorithms.

The evaluated ANFISs are subsequently sorted according to their RMSEs in descending order. For individual rule-based representation, the score table is initialised with each cell set to zero and, for the kth sorted ANFIS ($k \in \{1, 2, \ldots, A_{max}\}$), record the candidate rules used, and the scores of those relevant candidate rules are added by k. After doing this for all those created possible ANFISs, the score table results. The procedure for computing the score table is summarised as given in Algorithm 4.1.

4.2.5 Summary of GA-based ANFIS interpolation

The proposed evolutionary ANFIS interpolation algorithms are summarised as Algorithm 4.2.

4.2.6 ANFIS Interpolation: contrasting two algorithms

The two algorithms, implemented depending upon which chromosome representation is employed, have already been illustrated in Figures 4.3 and 4.4 respectively. There exist three major differences between these two implementations: 1) The objects

Algorithm 4.1: Calculation of score table.

Input:
Candidate rules;
C – Number of clusters;
S – Number of candidate rules in each cluster;
A_{max} – Number of ANFISs for evaluation.

```
Make a zero Score Table of S rows and C columns;
Make a zero Counting Table of S rows and C columns;
Make A_max possible ANFISs randomly and record candidate rules used;
Evaluate ANFISs and sort them according to RMSE in descending order;
for k = 1 to k = A_max do
    for i = 1 to i = C do
        record the candidate rule number j used;
        Score Table: score_ij ← score_ij + k;
        Counting Table: count_ij ← count_ij + 1;
    end
end
for i = 1 to i = C do
    for j = 1 to j = S do
        Score Table: score_ij ← score_ij / count_ij;
    end
end
```

Output:
Score table of all candidate rules.

that are evolved during the evolutionary process are different. One using rule based chromosome (termed 'Method 1' hereafter) conducts one evolutionary process per cluster, whilst the other using ANFIS based chromosome (termed 'Method 2') initialises the population directly with ANFISs, conducting the evolutionary process only once. This is of course, the fundamental reason for all the differences between these two algorithms. 2) The population initialisation methods are different, as discussed previously in Section 4.2.2.2. 3) The fitness measurement means are different, as described in Section 4.2.4.

Both algorithms have their own advantages and disadvantages, as summarised below. In particular, the individual rule-based approach requires less storage – Each chromosome only represents one rule, meaning that it is generally much shorter than a chromosome in ANFIS based method. This also leads to less running time, as to be indicated in an illustrative example later. However, the entire ANFIS-based representation offers a more convenient means for evaluation – The process of

Algorithm 4.2: ANFIS Interpolation - An evolutionary approach.

Input:
Source ANFISs in source domains: $\mathscr{A}_1, \mathscr{A}_2$;
Sparse training data in target domain: $\{(\mathbf{x}, y)\}$;
Population size (ANFIS chromosome): Q.

```
// - Population Initialization
```

Extract fuzzy rules $\{R_i\}$ from $\mathscr{A}_1$ and $\mathscr{A}_2$;
Construct rule dictionary D by Equations (3.3) and (3.4);
Divide sparse training data into C clusters, with respect to Equation (3.5);
for *each cluster* **do**
    Use centre of each cluster as seed, generate a set of individuals;
    for *each individual* **do**
        Interpolate one candidate rule;
    end
end
if *chromosome is rule type* **then**
    for *each cluster* **do**
        Collect candidate rules within the current cluster, to form an initial sub-population;
    end
end
else if *chromosome is ANFIS type* **then**
    for $i = 1$ *to* $i = Q$ **do**
        for *each cluster* **do**
            Choose one candidate rules in the current cluster;
        end
        Form an ANFIS chromosome by all chosen rules;
    end
    Collect all ANFIS chromosomes as initial population;
end
repeat

```
// - Crossover and Mutation
```

    Choose chromosomes for crossover with probability p_c;
    Do two-point crossover for chosen chromosomes;
    Choose chromosomes for mutation with probability p_m;
    Do mutation for chosen chromosomes;

```
// - Evaluation
```

    if *chromosome is rule type* **then**
        Evaluate chromosomes using Score Table;
    end
    else if *chromosome is ANFIS type* **then**
        Evaluate chromosomes using Equation (4.8);
    end
    Update population by roulette selection;
until *stop criteria is met*;

Output:
Interpolated ANFIS $\mathscr{A}_{int}$ in target domain.

assessing the quality of chromosomes to decide whether any of them will enter the next iteration can be directly evaluated in this approach, whilst the evaluation step of rule based chromosomes involves a much more complex procedure (using a score table and using an additional parameter of A_{max}).

4.2.7 Fine-tuning of GA-learned ANFIS model

A GA-learned network (through the process as described above) is used as an intermediate network for further fine-tuning with the sparse training data in target domain through the standard ANFIS learning algorithm. Compared with the use of the original standard ANFIS training procedure, which initialises the network parameters as 'zeros' or 'random values', the proposed ANFIS learning mechanism employs both fuzzy rule interpolation and GA-based evolutionary procedure, in an effort to produce higher quality initial network parameters. In so doing, the sparse training data is used more efficiently in the fine-tuning procedure for generating ANFIS of improved performance.

4.3 Complexity Analysis

The time complexity of the proposed methods (for both the rule based and ANFIS based chromosome representation) is analysed here. According to Algorithm 4.2, there are generally three stages in performing evolutionary ANFIS interpolation: population initialisation, crossover and mutation, and chromosome evaluation. As indicated previously, triangular fuzzy sets are utilised in implementing both methods and hence, the complexity analysis only involves the use of such fuzzy sets. The notations used are listed as follows:

m : number of antecedent attributes

n : number of fuzzy rules in rule dictionary

N : number of sparse training data points

C : number of clusters in training data

P : number of individuals in each cluster

Q : population size for ANFIS based chromosomes

A_{max} : number of ANFISs used in calculating score table

4.3.1 Rule-based chromosome representation

In the 'Population Initialisation' step, lines 1-2 extract all the rule parameters from sources ANFISs to construct the rule dictionary. There are totally n rules, with $4m+1$ parameters in each rule. Thus, running lines 1-2 costs $O(4mn+n)$. Following this, line 3 for clustering takes $O(NCm)$. Lines 4-9 repeat $C \times P$ times implementing candidate rule interpolation, each of which includes three sub-steps: (1) selecting K closest rules (at the cost of $O(n^2)$), (2) calculating weights (at $O(K)$) and (3) generating new rules (at $O(4m+1)$). Therefore, the complexity for computing lines 4-9 is $C \times P \times [O(n^2)+O(K)+O(4m+1)] = O(CPn^2)$. Next, lines 10-14 jointly lead to C initial populations, costing $O(C)$. Hence, the sub-total complexity for this step is $O(4mn+n)+O(NCm)+O(CPn^2)+O(C) = O(NCm) + O(CPn^2)$.

In the 'Crossover and Mutation' step, the crossover and mutation operation repeats C times. Line 25 selects $p_c P$ rules for crossover, costing $O(p_c P)$. In line 26, there are $\lfloor (p_c P)/2 \rfloor$ pairs of chromosomes. For each pair, in the worst case, all the parameters within the two chromosomes are exchanged, taking $O(4m+1)$. Then, in performing the mutation operation, lines 27-28 take $O(p_m P)$. Thus, the sub-total complexity is $C \times [O(p_c P) + \lfloor (p_c P)/2 \rfloor \times O(4m+1) + O(p_m P)] = O(mCP)$.

The rule based chromosome uses a score table to implement the 'Chromosome Evaluation' step. In running Algorithm 4.1, lines 1-2 take $O(SC)$. Line 3 makes A_{max} ANFISs with C rules in each, costing $O(A_{max}C)$. Line 4 costs $O(A_{max}^2)$ for sorting the A_{max} RMSEs. Following this, lines 5-11 repeat $A_{max}C$ times, costing $O(A_{max}C)$. Lines 12-16 repeat SC times with a complexity of $O(1)$ each time. Thus, the sub-total complexity for the 'Chromosome Evaluation' step is $O(SC)+O(A_{max}C)+O(A_{max}^2) = O(A_{max}^2)$.

In summary, the overall complexity for the proposed approach with rule based chromosome is: $O(NCm)+O(CPn^2)+O(mCP)+O(A_{max}^2)$.

4.3.2 ANFIS-based chromosome representation

For the method that exploits ANFIS based chromosome representation, the 'Population Initialisation' step is almost the same as that for the method using rule based chromosomes, expect for its final population construction process as given in lines 15-23, which costs $O(QC)$. Thus, running this 'Population Initialisation' step is at the cost of $O(4mn+n)+O(NCm)+O(CPn^2)+O(QC) = O(NCm)+O(CPn^2)+O(QC)$. Different from the method with rule based chromosome representation, here, the crossover and mutation operation are only implemented once within the population of the size Q. Line 25 takes $O(p_cQ)$. In line 26 the crossover operation repeats for $\lfloor(p_cQ)/2\rfloor$ times, and in the worst case, each crossover incurs exchanges across all the $C(4m+1)$ parameters in each ANFIS based chromosome, thereby costing $O(C(4m+1))$. Following this, lines 27-28 take $O(p_mQ)$. The sub-total complexity for the 'Crossover and Mutation' step is therefore, $O(p_cQ)+\lfloor(p_cQ)/2\rfloor\times O(C(4m+1))+O(p_mQ) = O(mCQ)$. Finally in the 'Chromosome Evaluation' step, line 33 evaluates all the Q chromosomes by sorting the related RMSEs, costing $O(Q^2)$. Hence, the overall complexity for the method with ANFIS based chromosome representation is: $O(NCm)+O(CPn^2)+O(mCQ)+O(Q^2)$.

Table 4.2: Summary of complexity analysis

Main steps	Rule-based chromosome	ANFIS-based chromosome
Population Initialization	$O(NCm)+O(CPn^2)$	$O(NCm)+O(CPn^2)+O(QC)$
Crossover and Mutation	$O(mCP)$	$O(mCQ)$
Chromosome Evaluation	$O(A_{max}^2)$	$O(Q^2)$
Overall	$O(NCm)+O(CPn^2)+$ $O(mCP)+O(A_{max}^2)$	$O(NCm)+O(CPn^2)+$ $O(mCQ)+O(Q^2)$

For clarity, the outcomes of the above computational complexity analysis are summarised in Table 4.2. Comparing the overall complexity of running the method using rule based chromosome representation and that using ANFIS based chromosome representation, it can be seen that the first two items ($O(NCm)+O(CPn^2)$) in each are the same. Moreover, in implementations, A_{max} and Q are usually set as the same number because both denote the number of ANFISs to be evaluated. Thus, the only difference between the complexities of these two methods lies in their respective

third items: ($O(mCP)$ vs. $O(mCQ)$). From this analysis, it can be concluded that the complexities of the two proposed methods do not differ very much. The time complexity of using the ANFIS based chromosome representation is slightly higher than that of using the rule based one, because that the value of Q is typically larger than that of P. This is verified experimentally later.

4.4 Experimentation and Validation

This section presents a systematic experimental evaluation of the proposed approach. Section 4.4.1 provides the general experimental set-up, including the parameters used, comparative methods employed, and performance index measured. Section 4.4.2 validates the two proposed algorithms by looking into a few synthetic function modelling cases, while Section 4.4.3 shows the effectiveness of the proposed approach in performing TSK regression over ten benchmark datasets.

4.4.1 Experimental setup

In the experimental studies, triangular membership functions are used in implementing the first layer of an ANFIS due to their popularity and simplicity. Both normalised and unnormalised data are used, reflecting the capability of the proposed approach in dealing with different data representations. Particularly, original data is used without normalisation in the synthetic data experiments for the convenience of result illustration. While in the experiments involving benchmark datasets, all input domains of the original data are normalised to $[0, 1]$. The classical roulette algorithm is employed for chromosome selection when updating the population in an evolutionary process. Following the common practice in the literature, the crossover and mutation probabilities are chosen as $p_c = 0.8$ and $p_m = 0.2$, respectively, unless otherwise stated. In setting up the initial population, the number of candidate rules of each cluster is set to $P = 5$. Note that P is the number of candidate rules per cluster in the very original population before any crossover and mutation, which differs in principle from the figure S in Table 4.1, that stands for the number of candidate rules in each cluster in the evaluation step.

Regression results using different ANFISs are compared, including: 1) An original ANFIS trained with the classical ANFIS learning method [66], using the sparse data

in the target domain only, named as 'Original ANFIS' hereafter; 2) An interpolated ANFIS obtained by the first ANFIS interpolation method described in Chapter 3, named as 'Method 0' (which interpolates one rule with respect to the centre of each cluster without involving evolutionary computation); 3) An interpolated ANFIS obtained by the proposed 'Method 1' (via rule based chromosome representation); and 4) An interpolated ANFIS obtained by the proposed 'Method 2' (via ANFIS based chromosome representation).

RMSE is taken to evaluate the performance of different ANFISs, as per the definition of Equation (4.7), where N now represents the number of the testing data points $\{x_k\}$ in the target domain; g_k denotes the corresponding ground truth of the kth data point; and $\mathscr{A}(x_k)$ stands for the output of different ANFISs on the data point x_k. Obviously, a smaller RMSE indicates a better performance, given otherwise the same conditions while performing statistical analyses.

4.4.2 Experiments on synthetic data

In this experimental study, synthetic data is created by sampling three non-linear functions, including two 1-D functions and one 2-D function, these function relationships used in the experiments are listed in Table 4.3.

Table 4.3: Functions used

No.	Function	x range
1	$cos(x) \cdot x$	$x \in [-10, 10]$
2	$sin(2x)/e^{x/5}$	$x \in [-10, 10]$
3	$sin(x_1/\pi)sin(x_2/\pi)$	$x_1 \in [-30, 30]$, $x_2 \in [-10, 10]$

4.4.2.1 Illustrative example

This is presented in order to show the main working procedures of the proposed ANFIS interpolation approach and the differences between the two types of chromosome representation. The first one dimensional non-linear function [$y = cos(x) \cdot x$] is used for giving the illustrative example, the underlying function is plotted for illustration in Figure 4.5.

As shown in Figure 4.5, the input domain $[-10, 10]$ is divided into three parts to simulate the source data and the target data. In particular, there are totally 201 data points sampled from this continuous function (with a sampling step of 0.1), in which the data in the left part (67 data points, shown in dashed line) forms the first source domain, which is used for training the first source ANFIS $\mathscr{A}_1$, and the data in the right part (67 data points, shown in dotted line) forms the second source domain for training the corresponding second ANFIS $\mathscr{A}_2$. These two source ANFISs ($\mathscr{A}_1$ and $\mathscr{A}_2$) are pre-trained using the standard ANFIS learning algorithm as described in Section 2.1, with 4 and 5 fuzzy rules resulted, respectively. The remaining data of the middle part (also 67 data points, shown in real line) forms the target domain, unlike the two source domains, it will be subsequently divided into two sub-parts with a small portion (20%, 13 data points) for training and the rest 80% (54 data points) for testing.

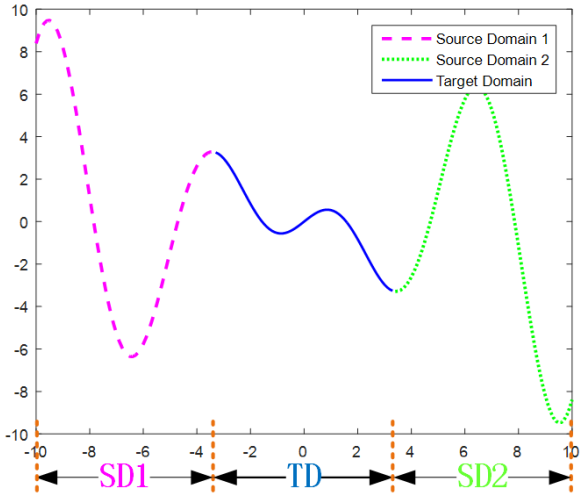


Figure 4.5: Illustration of source data and target data.

Following the proposed approach, the fuzzy rules embedded within the two well trained ANFISs $\mathscr{A}_1$ and $\mathscr{A}_2$ are extracted, forming the columns of the rule dictionary, with $4+5=9$ rules in total, as shown in Table 4.4. Note that the rule dictionary is the same for either the rule chromosome based method or the ANFIS chromosome based one.

Next, the sparse training data (13 data points) is clustered into C clusters, $C = \lfloor (4+5)/2 \rfloor = 4$, as determined with respect to Equation (3.5). Details of each cluster are listed in Table 4.5, from which it can be seen that the number of raw data in each

Table 4.4: Rule dictionary

Rule	Source ANFIS	Antecedent (a_0, a_1, a_2)	Consequent
1	$\mathscr{A}_1$	$(-12.2, -10.02, -7.83)$	$14.09x + 149.33$
2	$\mathscr{A}_1$	$(-9.95, -7.82, -5.55)$	$7.16x + 55.41$
3	$\mathscr{A}_1$	$(-7.79, -5.59, -3.39)$	$13.83x + 72.83$
4	$\mathscr{A}_1$	$(-5.62, -3.39, -1.2)$	$10.99x + 40.84$
5	$\mathscr{A}_2$	$(1.75, 3.39, 5.09)$	$7.16x - 27.67$
6	$\mathscr{A}_2$	$(3.39, 5.05, 6.7)$	$10.02x - 48.99$
7	$\mathscr{A}_2$	$(5.01, 6.7, 8.33)$	$6.36x - 36.38$
8	$\mathscr{A}_2$	$(6.71, 8.35, 9.96)$	$3.30x - 31.56$
9	$\mathscr{A}_2$	$(8.36, 10.01, 11.65)$	$9.81x - 106.53$

cluster is different, particularly in Cluster 1 there is only one data point. Thus, it can be very difficult to control the size of the initial population while scaling up, if the raw data is directly used. This issue is remedied with artificially generated individual instances, using the cluster centre as the seed by adding Gaussian white noise from it, with the seed itself being the mean and a small number δ (here $\delta = 0.2$) acting as the standard deviation. For example, the individuals of Cluster 3 are generated, consisting of the cluster centre itself (1.26) and the four randomly generated data points (1.13, 1.37, 0.89 and 1.05). Thus, the number of individuals P in each cluster become the same.

For each generated individual, a candidate rule is interpolated. As such, there are totally $4 \times 5 = 20$ interpolated rules (namely, number of clusters times that of individuals in each cluster), which are subsequently used to construct the initial population. For the method using rule based chromosome representation, the initial population contains 4 initial sub-populations, each with 5 chromosomes. For the method with ANFIS based chromosome representation, one candidate rule in each cluster is chosen and used for constructing possible ANFISs. There are 5 different choices in each cluster, so the number of possible ANFISs is $5^4 = 625$, which is also the size of the initial population. This is a doable number for the current simple illustrative case, however, for more complex problems with tens of clusters, this number will become excessively large and therefore, will require a huge storage space. In order to keep this under control, only a portion of all possible ANFISs are randomly selected as the chromosomes in implementing the initial population.

Table 4.5: Generating individuals for each cluster ($P = 5$)

Cluster	Center	Raw data	Number of data points	Generated individuals
1	-2.3	-2.3	1	$-2.3, -2.43, -2.18, -2.67, -2.5$
2	-0.62	$0.2, -1.1, -0.4, -0.6, -1.2$	5	$-0.62, -0.75, -0.5, -0.99, -0.82$
3	1.26	$1.4, 0.9, 1.5$	3	$1.26, 1.13, 1.37, 0.89, 1.05$
4	2.52	$2.2, 3.2, 2, 2.7$	4	$2.52, 2.38, 2.63, 2.15, 2.31$

Denote the number of possible ANFISs to take as Q, then, as stated previously, $Q = 500$ is empirically set in the present experimental studies.

In this illustrative example, the length of the individual rule-based chromosome is $(4m+1) = 5$ ($m = 1$), while that of the entire ANFIS-based chromosome is $5 \times 4 = 20$ (with 4 rules in each ANFIS). From this setup of the initial population, the crossover and mutation operations are followed. The changes incurred to the number of the chromosome during and after one evolutionary iteration is shown in Table 4.6. Of course, for rule based chromosome representation, this number is counted within one sub-population. Taking the method running on the entire ANFIS-based chromosome representation as an example, the population is initialised as $Q = 500$ chromosomes at the beginning. Then 400 chromosomes are chosen as parents for crossover with a crossover probability $p_c = 0.8$, resulting in 400 children. After crossover, the number of chromosomes becomes 500+400=900. Similarly, 100 chromosomes are chosen for mutation with a mutation probability $p_m = 0.2$, resulting in 100 new chromosomes, and the number of chromosomes after mutation becomes 900+100=1000. Finally, the evaluation process will select 500 chromosomes out of these 1000 to enter into the next iteration, ensuring that the population size remains the same.

Table 4.6: Number of chromosomes in one (sub-)population during different processes within one evolutionary iteration

	Initial Population	After Crossover	After Mutation	After Evaluation
Rule chromosome	5	9	10	5
ANFIS chromosome	500	900	1000	500

As mentioned previously, the methods for fitness evaluation of the two types of chromosome encoding are different. For the entire ANFIS-based representation, the evaluation is simply done using Equation (4.8) and hence, its illustration is omitted. Only is the evaluation of the individual rule-based chromosomes explained below.

Having produced four sub-populations (each with 10 chromosomes), a score table of size 10×4 is computed according to Algorithm 4.1. Figure 4.6 presents four instances of computed tables during one iteration. Firstly, the score table is initialised as a zero table of size 10×4, as illustrated in Table (a) of this figure. Then,

500 (A_{max}) possible ANFISs are randomly generated, with the rules used in these ANFISs recoded. The resulting ANFISs are evaluated using Equation (4.7) and are sorted in descending order according to their RMSEs. In this illustrative example, the first ANFIS (with the largest RMSE) is composed of 4 candidate rules indexed by candidates $[3, 10, 6, 4]$. That is, the first rule of this ANFIS is the third candidate rule (or chromosome) in cluster 1, the second rule is the tenth candidate in cluster 2, the third rule is the sixth candidate in cluster 3, and the last rule is the forth candidate rule in cluster 4. The value in each of the four corresponding locations (or cells) in the score table is therefore, added by 1, resulting in Table (b) of Figure 4.6. Next, given the ordered indices $[1, 7, 6, 10]$ of the second ANFIS, the values in their corresponding locations are added by 2, leading to Table (c) of Figure 4.6. This process is repeated for all of the 500 ANFISs. Averaging over such 500 tables results in the final score table that consists of the scores or fitness of all candidate rules, as shown in Table (d) of Figure 4.6, where the candidate rule with the largest average score in each cluster is shown in bold. Of course, here, the scores are the averages of how many times a certain candidate rule is utilised by the randomly generated (500) ANFISs.

4.4.2.2 Accuracy and runtime performance

The classical 5×5-fold cross validation is applied in the experiments to statistically evaluate the performance of different ANFISs. The mean and standard deviation values of RMSE using different ANFISs are listed in the first half of Table 4.7, while the visual result of one randomly selected fold regarding the first function modelling problem is shown in Figure 4.7.

As can be seen from the results, the 'Original ANFIS' gives the worst outcomes (which is not surprising due to data shortage in the target domain). While 'Method 0' (proposed in Chapter 3) shows an already significantly improved performance over the 'Original ANFIS', the GA-based algorithms produce further enhanced results. Quantitatively, running either 'Method 1' or 'Method 2' leads to smaller RMSE values with a narrower standard deviation, as shown in Table 4.7. Qualitatively, the shape of the estimated function by either of the proposed methods is much closer to that of the ground truth, as depicted in Figure 4.7.

Comparing the two proposed algorithms themselves, the results demonstrate that 'Method 1' (which represents individual rules as chromosomes) performs slightly

Candidate \ Cluster	1	2	3	4
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0

(a)

Candidate \ Cluster	1	2	3	4
1	0	0	0	0
2	0	0	0	0
3	1	0	0	0
4	0	0	0	1
5	0	0	0	0
6	0	0	1	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	1	0	0

(b)

Candidate \ Cluster	1	2	3	4
1	2	0	0	0
2	0	0	0	0
3	1	0	0	0
4	0	0	0	1
5	0	0	0	0
6	0	0	3	0
7	0	2	0	0
8	0	0	0	0
9	0	0	0	0
10	0	1	0	2

(c)

Candidate \ Cluster	1	2	3	4
1	221.3	311.1	244.2	373.5
2	281.6	258.1	217.5	258.1
3	220.2	333.2	152.7	**409.0**
4	265.9	169.3	339.1	148.3
5	178.2	229.6	287.3	245.0
6	260.0	245.1	164.3	312.1
7	236.1	159.7	**358.2**	132.9
8	**296.1**	313.2	222.2	197.2
9	268.9	**355.1**	283.2	373.7
10	283.5	131.2	250.1	116.1

(d)

Figure 4.6: Example for score table computation

better than 'Method 2' (which expresses an entire ANFIS as one chromosome), for this function modelling problem. The likely reason for this is that, as the rule based chromosomes are shorter than the ANFIS based ones, there are more opportunities for crossover and mutation operations to take effect under the same probabilistic set up.

Both implementations for the proposed ANFIS interpolation approach are also compared with five conventional machine learning methods: linear regression (LinR), support vector regression (SVR), classification and regression tree (CART), random forest (RF), and an evolutionary ANFIS method (E-ANFIS [111]) in which the genetic algorithm is used to learn the network parameters (without interpolation). The

Table 4.7: Accuracy (Mean ± Standard deviation) and running time (seconds) of different methods on synthetic data

Methods	Function 1	Function 2	Function 3	Run time
Original ANFIS	0.913 ± 1.184	1.329 ± 1.153	0.468 ± 0.244	0.32 ± 0.21
Method 0	0.364 ± 0.406	0.905 ± 0.743	0.351 ± 0.148	0.65 ± 0.26
Method 1	**0.294** ± 0.401	**0.691** ± 0.461	0.334 ± 0.147	40.21 ± 27.33
Method 2	0.317 ± 0.441	0.794 ± 0.552	**0.325** ± 0.096	55.42 ± 29.70
LinR	1.243 ± **0.055**	0.772 ± **0.022**	0.471 ± **0.007**	3.04 ± 2.52
SVR	1.234 ± 0.064	0.789 ± 0.030	0.470 ± **0.007**	3.44 ± 2.13
CART	1.302 ± 0.329	1.394 ± 0.235	0.393 ± 0.037	3.32 ± 1.75
RF	0.547 ± 0.133	0.805 ± 0.167	0.408 ± 0.024	3.61 ± 0.78
E-ANFIS	1.552 ± 0.478	0.836 ± 0.048	0.476 ± 0.012	359.21 ± 126.42

evaluation results are listed in the second half of Table 4.7. From the accuracy over the three function modelling case studies, it can be observed that the proposed ANFIS interpolation methods ('Method 1' or 'Method 2') perform better in terms of mean error values. Although the classical machine learning methods (especially the LinR approach) may give better standard deviation values, their overall accuracy is lower than what is achieved by the proposed methods.

The average running time (also showing in the 'Mean ± Standard deviation' form) of different methods over the three function modelling cases is shown in the last column of Table 4.7. Comparing the two proposed methods with other approaches, as can be expected, they run faster than 'E-ANFIS' but incur more computation than the rest. However, the considerably improved accuracy they gain over the other approaches justifies this increase in computational cost. Comparing the two proposed methods themselves, it can be seen that the method employing individual rule-based

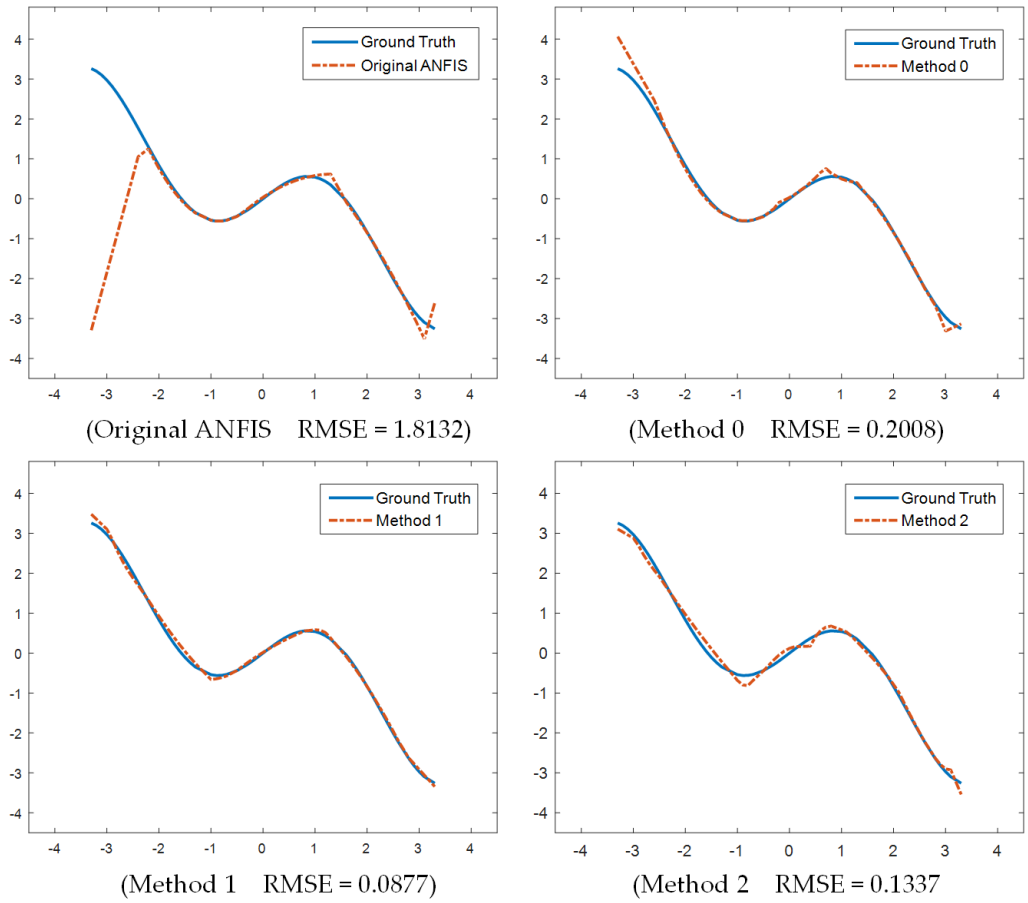


Figure 4.7: Visual illustration of one-fold result by different ANFISs

chromosome representation consumes less time, this conforms to the theoretical analysis presented in Section 4.3.

4.4.3 Experiments on real data

A number of investigations are carried out here, looking into the issues regarding accuracy, amount of training data and key GA parameters (namely, crossover and mutation rates).

4.4.3.1 Datasets and experimental environment

Table 4.8 lists the ten popular benchmark real-world regression datasets, taken from the KEEL data respository [5], which are used here to further evaluate the

performance of both GA-based ANFIS interpolation algorithms.

Table 4.8: Datasets used in experimental study

Dataset	Attribute No.	Instance No.
Diabetes	2	43
Plastic	2	1650
Quake	3	2178
Laser	4	993
AutoMPG6	5	392
Delta-ail	5	7129
Friedman	5	1200
Dee	6	365
Delta-elv	6	9517
ANACALT	7	4052

To conduct this set of experimental investigations, the source domains and target domain are created in a similar manner to the synthetic data experiments, by splitting each entire dataset into three parts according to one of the input variables.

4.4.3.2 Accuracy analysis

The 5×5-fold cross validation results over the ten datasets are shown in Table 4.9, with the average RMSE values listed in the last row and the best results indicated in bold. Similar observation can be made from Table 4.9 to those results obtained from the synthetic data experiments. The 'Original ANFIS' gives the poorest performance as expected, and the other three interpolated ANFISs improve the inference results obviously. Amongst the three interpolation-based methods, both GA-based implementations outperform 'Method 0' for most cases, though there is no clear winner between the two themselves.

The p-values of the pairwise t-test ($p = 0.05$) over different methods are listed in Table 4.10. In this table, sign (v) ($p < 0.05$) represents that the improvement is of statistic significance, while sign (*) denotes that the improvement is not significant statistically. From the results, it can be seen that both 'Method 1' and 'Method 2'

Table 4.9: Experimental results using different ANFISs on real world datasets

Datasets	Mean ± Standard deviation of ANFISs			
	Original ANFIS	Method 0	Method 1	Method 2
Diabetes	1.527 ± 0.829	1.058 ± 0.497	$\mathbf{0.989 \pm 0.359}$	1.011 ± 0.451
Plastic	2.003 ± 0.246	1.843 ± 0.105	1.836 ± 0.093	$\mathbf{1.828 \pm 0.078}$
Quake	1.264 ± 0.641	0.542 ± 0.248	$\mathbf{0.458 \pm 0.175}$	0.498 ± 0.225
Laser	12.087 ± 4.739	3.757 ± 2.724	$3.476 \pm \mathbf{1.069}$	$\mathbf{3.279} \pm 1.262$
AutoMPG6	12.770 ± 1.431	5.120 ± 1.038	$\mathbf{4.776} \pm 1.015$	$4.987 \pm \mathbf{0.843}$
Delta-ail	$3.53 \times 10^{-4} \pm 1.09 \times 10^{-4}$	$1.78 \times 10^{-4} \pm 1.06 \times 10^{-5}$	$\mathbf{1.73 \times 10^{-4} \pm 8.87 \times 10^{-6}}$	$1.75 \times 10^{-4} \pm 9.49 \times 10^{-6}$
Friedman	5.177 ± 0.633	3.004 ± 0.409	$\mathbf{2.900 \pm 0.335}$	2.930 ± 0.405
Dee	1.017 ± 0.339	0.870 ± 0.225	$\mathbf{0.824 \pm 0.188}$	0.839 ± 0.191
Delta-elv	$0.0034 \pm 6.195 \times 10^{-4}$	$0.0020 \pm 1.69 \times 10^{-4}$	$\mathbf{0.0018} \pm 7.11 \times 10^{-5}$	$\mathbf{0.0018 \pm 5.90 \times 10^{-5}}$
ANACALT	1.209 ± 0.966	0.994 ± 0.768	0.999 ± 0.675	$\mathbf{0.986 \pm 0.681}$
Average	3.706 ± 0.982	1.719 ± 0.601	$\mathbf{1.626 \pm 0.391}$	1.635 ± 0.412

provide improvements of statistical significance compared with the original ANFIS model. However, as shown in the last column of Table 4.10, all p values are larger than 0.05, indicating that there is no statistical improvements between the two algorithms with different chromosomes. This can be expected as the experimental results on RMSEs also shown that this is no clear winner between 'Method 1' and 'Method 2'.

Table 4.10: p values of pairwise t-test results

Dataset	Method 1 v.s. Original ANFIS	Method 2 v.s. Original ANFIS	Method 1 v.s. Method 2
Diabetes	0.0043 (v)	0.0108 (v)	0.6948 (*)
Plastic	0.0035 (v)	8.3728e-04 (v)	0.5797 (*)
Quake	1.36e-06 (v)	2.52e-06 (v)	0.3557 (*)
Laser	7.96e-10 (v)	9.67e-10 (v)	0.4834 (*)
AutoMPG6	1.17e-18 (v)	1.46e-18 (v)	0.0558 (*)
Delta-ail	1.27e-08 (v)	1.73e-08 (v)	0.1894 (*)
Friedman	9.61e-15 (v)	6.22e-16 (v)	0.4926 (*)
Dee	0.0046 (v)	0.0033 (v)	0.5649 (*)
Delta-elv	2.10e-12 (v)	1.52e-12 (v)	0.6828 (*)
ANACALT	0.0165 (v)	0.0111 (v)	0.1954 (*)

4.4.3.3 Comparison with other machine learning methods

As with the experiments on synthetic data, the proposed approach is also compared with five conventional machine learning methods: LinR, SVR, CART, RF, and E-ANFIS. Different from those earlier experiments (where triangular membership functions and a very simple clustering method were used), to enrich the experimental investigation, Gaussian membership functions and fuzzy c-means clustering are utilised here, in implementing the proposed ANFIS interpolation methods. The results are listed in Table 4.11 with the best highlighted in bold.

The proposed approach outperforms the others in a substantial majority of cases. For the three particular cases (of LinR on the Quake dataset, SVR on Dee, and RF on AutoMPG6) where an existing method shows the best outcome, the accuracy of the proposed approach is close. In particular, the proposed approach with ANFIS based

Table 4.11: Accuracy (Mean ± Standard deviation) of different methods on real data

Dataset	Results of different methods						
	LinR	SVR	CART	RF	E-ANFIS	Method 1	Method 2
Diabetes	0.533 ± 0.100	0.441 ± 0.052	0.779 ± 0.130	0.391 ± 0.052	1.172 ± 1.100	$\mathbf{0.362 \pm 0.045}$	0.377 ± 0.050
Plastic	1.623 ± 0.044	$2.285 \pm \mathbf{0.037}$	2.245 ± 0.128	1.886 ± 0.069	1.865 ± 0.348	1.620 ± 0.046	$\mathbf{1.603} \pm 0.056$
Quake	$\mathbf{0.194 \pm 0.006}$	0.206 ± 0.007	0.243 ± 0.015	0.195 ± 0.008	0.259 ± 0.153	$0.198 \pm \mathbf{0.006}$	0.199 ± 0.007
Laser	6.934 ± 1.337	$15.522 \pm \mathbf{0.435}$	12.87 ± 2.685	7.680 ± 0.801	5.644 ± 2.636	4.547 ± 1.632	$\mathbf{4.030} \pm 1.539$
AutoMPG6	3.569 ± 0.685	3.336 ± 0.353	4.044 ± 0.559	$\mathbf{3.282 \pm 0.331}$	5.625 ± 1.275	3.628 ± 0.696	3.575 ± 0.572
Delta-ail	$1.55 \times 10^{-4} \pm 1.48 \times 10^{-6}$	$1.59 \times 10^{-4} \pm 1.66 \times 10^{-6}$	$1.91 \times 10^{-4} \pm 7.33 \times 10^{-6}$	$1.66 \times 10^{-4} \pm 5.13 \times 10^{-6}$	$1.64 \times 10^{-4} \pm 3.86 \times 10^{-5}$	$\mathbf{1.50 \times 10^{-4} \pm 1.45 \times 10^{-6}}$	$1.51 \times 10^{-4} \pm 1.46 \times 10^{-6}$
Friedman	2.795 ± 0.049	$2.771 \pm \mathbf{0.047}$	7.224 ± 0.483	2.771 ± 0.156	2.408 ± 1.325	2.041 ± 0.075	$\mathbf{2.030} \pm 0.052$
Dee	0.424 ± 0.023	$\mathbf{0.416 \pm 0.021}$	1.016 ± 0.063	0.482 ± 0.035	0.810 ± 0.366	0.477 ± 0.048	0.486 ± 0.064
Delta-elv	$1.51 \times 10^{-3} \pm 1.17 \times 10^{-5}$	$1.53 \times 10^{-3} \pm 1.23 \times 10^{-5}$	$1.95 \times 10^{-3} \pm 5.19 \times 10^{-5}$	$1.51 \times 10^{-3} \pm 1.26 \times 10^{-5}$	$1.55 \times 10^{-3} \pm 1.78 \times 10^{-4}$	$\mathbf{1.48 \times 10^{-3} \pm 1.03 \times 10^{-5}}$	$\mathbf{1.48 \times 10^{-3}} \pm 1.09 \times 10^{-5}$
ANACALT	$0.405 \pm \mathbf{0.020}$	0.333 ± 0.021	0.299 ± 0.038	0.292 ± 0.043	0.357 ± 0.100	0.280 ± 0.029	$\mathbf{0.279} \pm 0.028$
Average	1.648 ± 0.227	$2.531 \pm \mathbf{0.097}$	2.873 ± 0.409	1.698 ± 0.149	1.814 ± 0.811	1.316 ± 0.258	$\mathbf{1.258} \pm 0.236$

chromosome representation gives by far, the best mean value of the overall RMSE averaged across the ten datasets, and the proposed with rule based chromosome representation achieves the second best (as shown in the last row of Table 4.11). This superiority in performance remains to be true taking into consideration the standard deviations. Note however, that in terms of the overall averaged standard deviation measure, the classical SVR has the least value, demonstrating its excellent stability.

Table 4.12 shows the running time performance of different methods (given in the 'Mean ± Standard deviation' form), in which each line is obtained by averaging the running time over different datasets. It can be observed that the standard deviations of the two proposed methods ('Method 1' and 'Method 2') and the 'E-ANFIS' method are relatively larger than that of the rest. The running time of an algorithm on different dataset with different numbers of input variables may vary, and for the ANFIS-based approaches, the running time is more sensitive to the change of the number of input variables. For datasets with more input variables, these algorithms will produce a large rule-base in the resulting ANFIS, and operating on this large rule-base will consume much more time than coping with datasets containing less input variables. This leads to the relatively larger standard deviation of the ANFIS-based methods.

From Table 4.12, similar observations as per those on the synthetic data case studies can be drawn: The proposed methods are faster than E-ANFIS, but slower than the rest, since the evolutionary process incurs more computation. Between the two proposed methods themselves, it is clear that the implementation with ANFIS-based chromosome representation consumes more time. Again, this is expected, reflecting the result of theoretical analysis on computational complexity.

4.4.3.4 Effects of sparsity in training data

This particular experimentation is conducted to investigate the performance of different ANFISs in response to the use of different percentages of training data, instead of just the 20% and 80% split. To focus on the discussion about the performance of varying the amount of training data, only the results on the 'Quake' dataset are shown here, as given in Figure 4.8.

Table 4.12: Average running time (seconds) of different methods on real data

Methods	Running time
LinR	1.46 ± 0.15
SVR	13.34 ± 15.57
CART	2.38 ± 1.85
RF	3.47 ± 2.5
E-ANFIS	225.3 ± 171.65
Method 1	35.15 ± 20.2
Method 2	64.91 ± 36.55

It can be seen that in general, the RMSE values decrease as the percentage of training data increases, independent of which learning method is used. Importantly, also independent of what percentage of training data is utilised, the three interpolated ANFISs all remarkably outperform the 'Original ANFIS' (until training data reaches 90%). Furthermore, the RMSE values of the three interpolated ANFISs generally decline much less rapidly than the RMSE of the 'Original ANFIS', indicating that the interpolated ANFISs are less sensitive to the decrease of training data. Indeed, the performances of these three methods using 30% data are very close to those using 90% data. When the available data percentage is 20%, the results are of course the same as discussed in the preceding sub-section, with the three interpolated ANFISs significantly beating the 'original ANFIS' and the two GA-based methods outperforming 'Method 0'. When only 10% of data is available for training, the performances of the interpolated methods degrade but still beat the 'Oriainl ANFIS' trained with the conventional method substantially. Most interestingly, even in this case, either of the two proposed methods retains its superior performance over 'Method 0', especially when rule based chromosome representation is used. Overall, the proposed ANFIS interpolation algorithms via evolutionary computation achieve a much better result while using less training data than the existing techniques.

4.4.3.5 Effects of crossover and mutation probability

The crossover probability p_c and mutation probability p_m are two important parameters in the evolutionary process. Generally, larger crossover probability and smaller mutation probability are a common choice for most evolutionary problems.

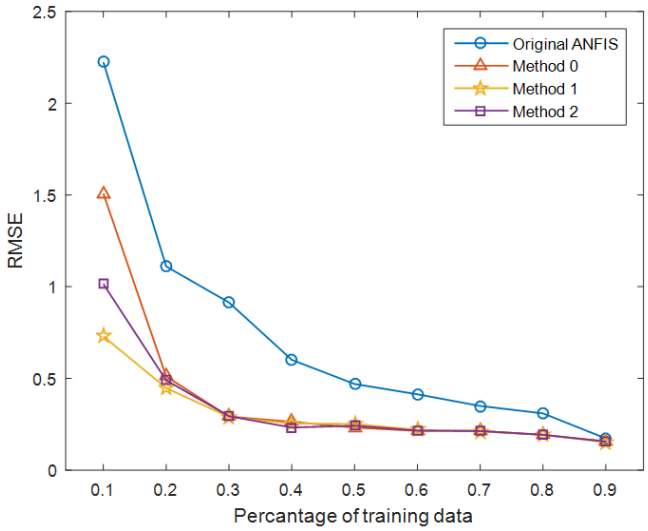


Figure 4.8: Performance vs. percentage of training data used

In the above-reported experimental investigations, these two parameters are set to $p_c = 0.8, p_a = 0.2$ respectively. In order to examine the effects on performance of using different p_c and p_m, these probabilities are varied from 0.1 to 0.9 (with step=0.1) in this experimentation. This part of the experimental studies is focussed on the use of the entire ANFIS based chromosome representation since the number of the individual rule-based chromosome in each sub-population is very small (only 5 in the experiments, so the results of using different p_c and p_m do not differ much in the first place). The resulting box-plot over the 'Quake' dataset is shown in Figure 4.9 and Figure 4.10. In conformation with common knowledge in evolutionary computation the results reveal that larger p_c as well as smaller p_m give a better value for both the median and the interquartile range.

4.5 Summary

This chapter has presented a new ANFIS interpolation approach via evolutionary computation (implemented with GAs), in an effort to improve the learning of ANFIS when there is significant shortage of training data for the problem concerned. The concept of 'ANFIS interpolation' means the interpolation of an entire inference system, or a whole group of rules in the target region. This is enabled with the assistance of well-trained ANFISs in the neighbouring regions. Two forms of chromosome

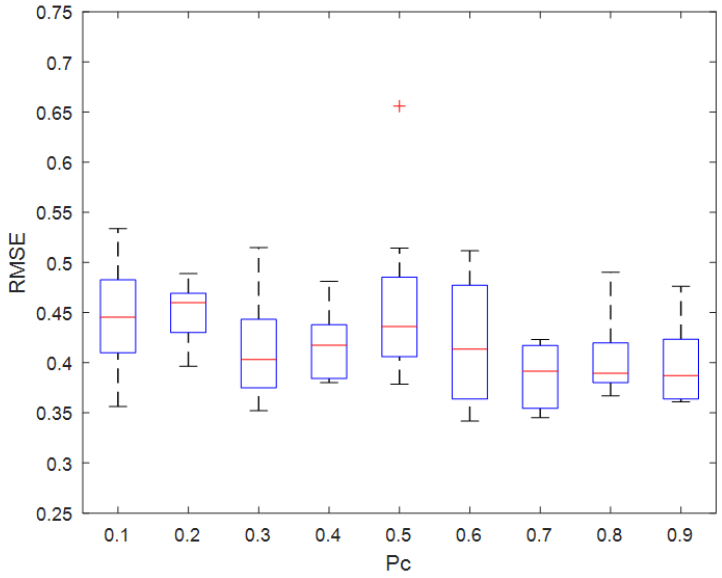


Figure 4.9: Performance vs. crossover rate.

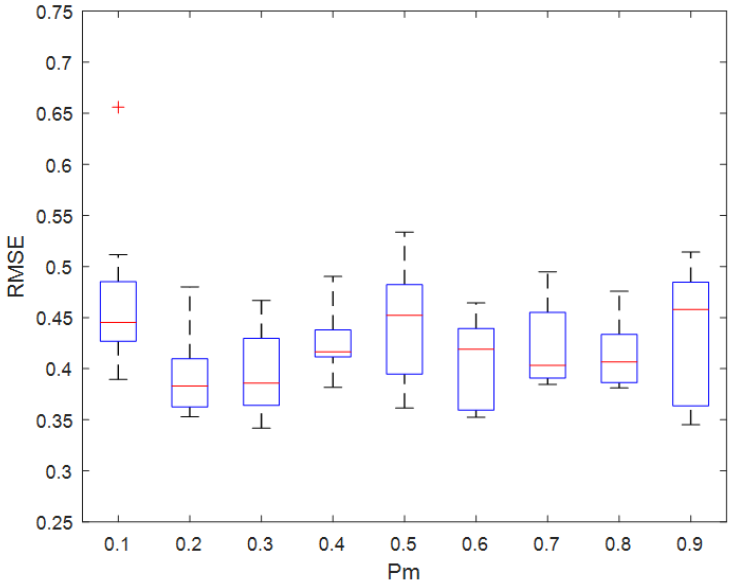


Figure 4.10: Performance vs. mutation rate.

representation are considered: one encoding individual rules and the other expressing an entire possible ANFIS. The proposed approach has been tested on both three function modelling problems with synthetic data and real world regression problems involving 10 benchmark datasets, demonstrating its ability in significantly improving the inference performance when compared with existing techniques.

Compared with the ANFIS interpolation approach proposed in Chapter 3 (based on group-based rule interpolation), this evolutionary approach has the following new contributions:

1. Unlike the initial approach which terminates when a group of rules are interpolated, this improved ANFIS interpolation is implemented within the framework of a genetic algorithm which iteratively updates the interpolated rules. This is the fundamental contribution from which the following two innovative technical developments have been made.
2. Two types of chromosome representations are designed to perform the GA process, resulting in two slightly different evolutionary algorithms that have been systemically compared in this chapter.
3. As an ANFIS can be equivalently interpreted as a set of fuzzy rules, the fitness of each single rule based chromosome is very hard to measure. In order to solve this problem, the concept of 'score table' has been developed in this chapter, from which the performance of all rule type chromosomes are clearly exhibited.

Further to the above systematic evaluation with benchmark datasets, the proposed approaches are also applied to dealing with the problem of image super resolution. Particularly, Chapter 5 will use full training data to test the performance of the learned ANFIS mappings under the condition where sufficient training data is available.

Chapter 5

Image Super Resolution with Full Training Data via Multiple Learned ANFIS Mappings

In most existing learning based super resolution algorithms, the mappings from low resolution space to its corresponding high resolution space is learned using non-fuzzy techniques [43, 149, 171]. However, image super resolution is a highly ill-posed problem where an observed low resolution (LR) image patch can be generated from any of the many possible high resolution (HR) patches. Therefore, it would be intuitive to introduce imprecision (or fuzzy) techniques to the learning based image super resolution (ISR) tasks. Hither to little work in this direction has been developed [6]. However, an interesting approach has been proposed recently for fuzzy rule based ISR [122] whose basic idea is to develop a set of fuzzy rules describing the required non-linear mappings. These fuzzy rules are learned from the low resolution and high resolution (LR-HR) image pairs of a given training data set, and in this learning process, the most important part is how to decide the rule parameters.

Based on the fact that ANFIS has been proven to be a powerful tool for generating effective fuzzy rules, it would be interesting to investigate its potential to help resolve the image super resolution problem. This chapter introduces the work for an ANFIS to learn its parameters to perform the ISR task with full training data. A new algorithm for single frame image super resolution using multiple learned ANFIS mappings is

developed. In particular, a large number of LR-HR patch pairs from a collection of natural images are generated and used as training data, from which multiple ANFIS mappings from LR space to HR space can be formulated. Those learned ANFIS mappings are tested against a range of natural images and over different scale factors.

The rest of this chapter is organised as follows. Section 5.1 outlines the framework of the proposed ANFIS based image SR approach. Section 5.2 details the proposed approach, performing ISR by an integrated use of the learned ANFIS mappings and a number of pre-processing and post-processing techniques. Experimental results against a number of natural images are shown in Section 5.3, demonstrating the effectiveness of the proposed method. Section 5.4 provides a summary of the reported work in this chapter.

5.1 Framework of Proposed Approach

This section describes the framework of the proposed image super resolution algorithm, which involves two phases: (1) Offline learning, where multiple ANFIS mappings are learned via a large amount of LR-HR image patch pairs. (2) Online application, where the learned multiple ANFIS mappings are used to generate the corresponding HR image from an input LR image.

The flowchart of the proposed algorithm is shown in Figure 5.1. At the learning stage, an external database of LR and HR image pairs is used as the training set. Then each image pair in the training set is divided into small LR-HR patch pairs with a fixed size. Next, the resulting patch pairs are grouped into different clusters using a K-means clustering method, and for each cluster, an ANFIS mapping is learned using the standard ANFIS learning process. When in use (after training), a given input LR image is firstly divided into image patches with the same size as the patches in training stage. For each patch, the most relevant ANFIS mapping is then selected to perform SR, resulting in its corresponding HR patch. By combining all the HR patches, a reconstructed image is obtained. Following this, a non-local means (NLM) filter (which works by exploiting the image's non-local self similarity) and a iterative back projection (IBP) operator are employed to refine the reconstructed image. This process will be repeated for several times and will finally output the estimated HR image.

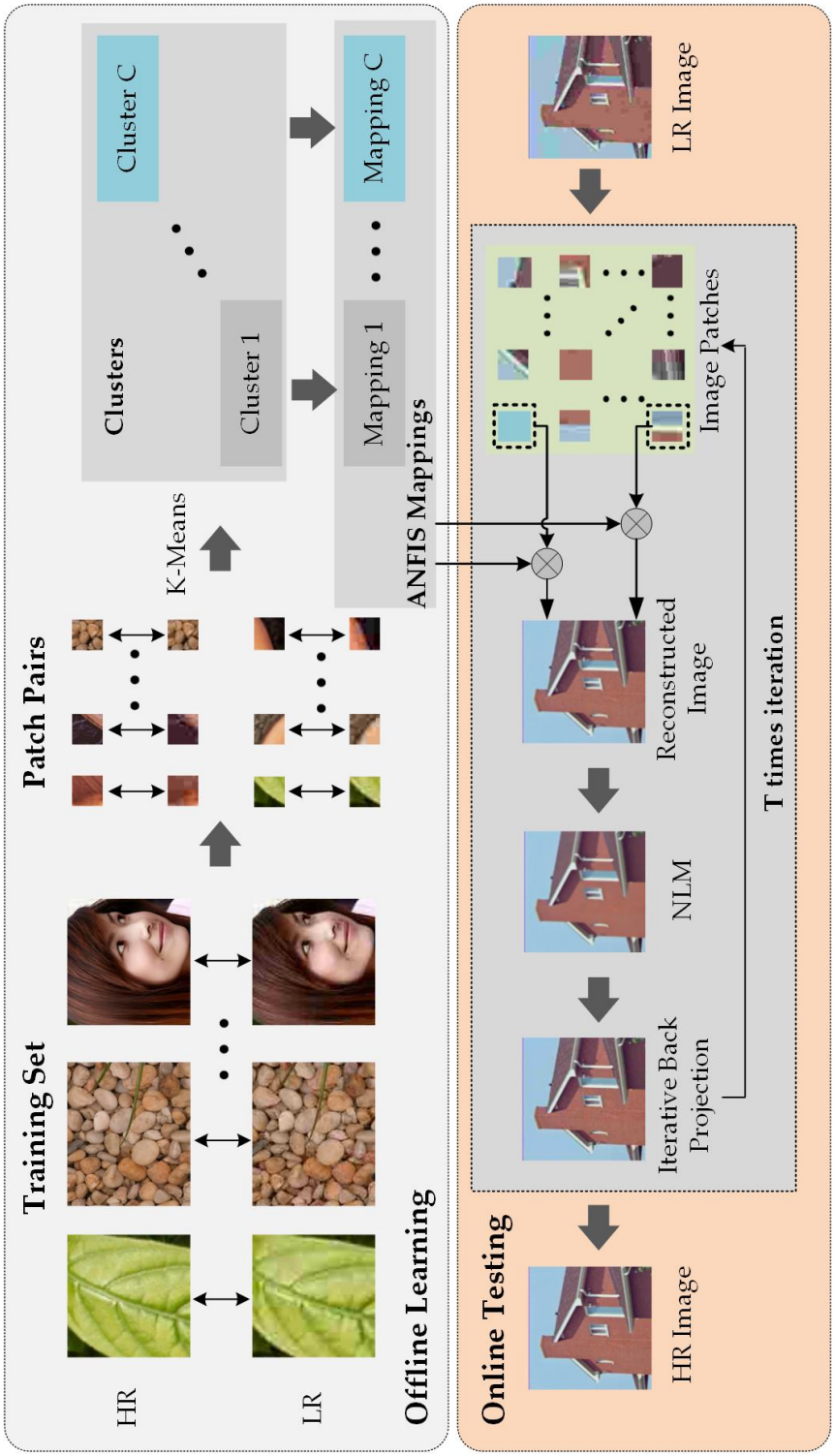


Figure 5.1: Flowchart of the proposed approach

5.2 Implementation of the Proposed Approach

This section presents the implementation details of the framework outlined in Section 5.1. The learning process is introduced in 5.2.1, while the testing process as well as the pre-processing and post-processing techniques utilized in this work are presented in 5.2.2.

5.2.1 Learning of multiple ANFIS mappings

5.2.1.1 Training set generation

At the beginning of the training stage, the training data set, which is a set of image patch pairs of both HR and LR resolutions, is generated. A (prescribed) large number of sharp natural images $\{\mathbf{Y}^h\}$ are firstly collected and used to form the training database, including various types of scenes (e.g. people, animals, buildings and plants) with fine textures. Some representative images are shown in Figure 5.2.

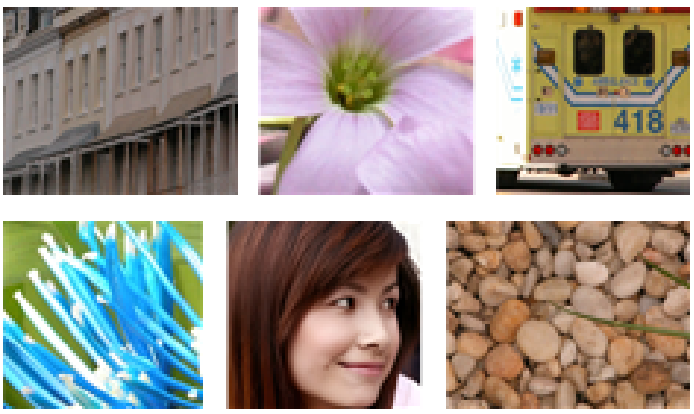


Figure 5.2: Example images used for training.

Those sharp images are considered as HR images in the training set, and the corresponding LR version $\{\mathbf{X}^l\}$ of the HR images $\{\mathbf{Y}^h\}$ are simulated by implementing the following steps for each HR image: 1) Down sample the HR image with a scale factor of s; 2) Up scale the down sampled images to a certain desired size by bicubic interpolation, so that the LR and the desired HR images are of the same size but the former are of lower resolution (which may mean they contain less information).

Then, the LR-HR image pairs $\{(\mathbf{X}^l, \mathbf{Y}^h)\}$ are divided into overlapping patch pairs with a size of $q \times q$, denoted as $\mathbf{Q} = \{(\mathbf{x}^l, \mathbf{y}^h)\}$. After vectorization, each patch is a column vector $\mathbf{x}^l \in R^{q^2}, \mathbf{y}^h \in R^{q^2}$ containing q^2 pixels (each pixel is denoted by x^i and y^i respectively). After that, the LR-HR image patch pairs $\mathbf{Q}$ can be used to train the multiple ANFIS mappings.

5.2.1.2 Multiple ANFISs learning

The above resulting patch pairs $\mathbf{Q} = \{(\mathbf{x}^l, \mathbf{y}^h)\}$ are subsequently partitioned into P clusters with centroids $\mathbf{V}$ by the use of conventional K-Means clustering algorithm:

$$\mathbf{V} = \{\mathbf{v}_p^* = (\mathbf{v}_p^l, \mathbf{v}_p^h), p = 1, 2, \cdots, P\} \tag{5.1}$$

For each cluster, a mapping from the LR image patch to the HR patch is trained over an ANFIS network. The learning procedure is summarised as Algorithm 5.1.

Algorithm 5.1: Learning multiple ANFIS mappings.

Input:
A set of sharp HR natural images $\{\mathbf{Y}^h\}$;
Scale factor s;
`// Training set Generation`
1 **for** *each* $\mathbf{Y}^h$ **do**
2 Down sample $\mathbf{Y}^h$ with scale factor $\frac{1}{s}$, resulting in $\mathbf{Y}^l$;
3 Up sample $\mathbf{Y}^l$ with scale factor s using bicubic interpolation algorithm, resulting in $\mathbf{X}^l$;
4 **end**
5 Generate LR-HR image pairs $\{(\mathbf{X}^l, \mathbf{Y}^h)\}$;
6 Divide $\{(\mathbf{X}^l, \mathbf{Y}^h)\}$ into $q \times q$ patch pairs $\mathbf{Q} = \{(\mathbf{x}^l, \mathbf{y}^h)\}$ with overlaps;
7 Vectorize patch pairs to pixel pairs such that $\mathbf{x}^l \in R^{q^2}, \mathbf{y}^h \in R^{q^2}$;
`// Multiple ANFISs Learning`
8 Partition $\mathbf{Q}$ into P clusters by K-Means with centroids $\mathbf{V} = \{(\mathbf{v}_p^l, \mathbf{v}_p^h)\}$;
9 **for** *each cluster* **do**
10 learn an ANFIS mapping $\mathscr{A}_p$;
11 **end**
Output:
Learned multiple ANFIS mappings $\{\mathscr{A}_p\}$;
Centroids $\mathbf{V} = \{(\mathbf{v}_p^l, \mathbf{v}_p^h)\}$.

5.2.2 Image SR with learned ANFIS mappings

In the testing stage, a given new LR image can be reconstructed using the trained ANFIS mappings. This process is implemented with the following three sub-stages: pre-processing, inference with learned mappings and post-processing.

5.2.2.1 Pre-processing

Denote the given testing LR image as $\mathbf{X} \in R^{m \times n}$, where m is the image length and n is the image width, and denote the desired HR image as $\mathbf{Y} \in R^{M \times N}$, so there exists the relationship that $M = sm, N = sn$. For training convenience, in the learning stage the LR images of the LR-HR training image pairs are simulated by down sampling then up sampling the HR image, so the resulting LR images are also of big size $M \times N$. However, the real input LR image is with smaller size $m \times n$. Therefore, similar with the training stage, the input testing LR image will be pre-processed using pixel based interpolation method before inferencing using the learned ANFIS mappings.

Existing representative pixel based interpolation algorithms include the nearest neighbour interpolation, the linear interpolation, the bilinear interpolation and the bicubic interpolation [89]. Amongst them, the bicubic interpolation [56, 82] is the most commonly used one that increases the image pixels by interpolating existing ones using bicubic basic functions, so is employed in the proposed work.

5.2.2.2 Inference with learned ANFIS mappings

After pre-processing, the resulting large size image (also denoted by $\mathbf{X}$ here) will be subsequently enhanced using the learned ANFIS mappings obtained from the learning stage. More specifically, the large size image $\mathbf{X}$ is firstly divided into image patches of size $q \times q$ with overlaps. Here, the patch size q is the same as that used during the training process, and the image patches $\mathbf{x} \in R^{q \times q}$ are also vectorized for computation convenience. Then, for every vectorized image patch $\mathbf{x} \in R^{q^2}$, an appropriate ANFIS mapping is selected subject to the following constraint:

$$p = \min_{p} \| \mathbf{x} - \mathbf{v}_p^l \|_2^2 \tag{5.2}$$

where $\mathbf{v}_p^l$ is the first part of the centroid $\mathbf{v}_p^*$ in Equation (5.1). And the pth ANFIS mapping $\mathscr{A}_p$ is selected to estimate the corresponding HR image patch. For each image patch $\mathbf{x}$, the basic idea for inferencing can be expressed by:

$$If\ \mathbf{x}\ is\ close\ to\ \mathbf{v}_p^l\ \ then\ \mathbf{y} = \mathscr{A}_p(\mathbf{x}). \tag{5.3}$$

where $\mathbf{y} \in R^{q^2}$ is the corresponding HR patch. The behaviour of the function $\mathscr{A}_p(.)$ is inference using the pth ANFIS mapping from input LR patch to HR patch. Now, as the LR patch is q^2-dimensional ($\mathbf{x}$ is a vector with q^2 pixels), the antecedent part of Equation (5.3) can be viewed as a conjunction of q^2 atomic clauses:

$$If\ \mathbf{x}\ is\ close\ to\ \mathbf{v}_p^l \approx \begin{cases} If\ x^1\ is\ close\ to\ v_p^{l1} \\ If\ x^2\ is\ close\ to\ v_p^{l2} \\ \cdots\cdots \\ If\ x^{q^2}\ is\ close\ to\ v_p^{lq^2} \end{cases} \tag{5.4}$$

Similarly, the consequent part of Equation (5.3) can be rewrite as:

$$\begin{cases} y^1 = \mathscr{A}_p(x^1) \\ y^2 = \mathscr{A}_p(x^2) \\ \cdots\cdots \\ y^{q^2} = \mathscr{A}_p(x^{q^2}) \end{cases} \tag{5.5}$$

where y^i denotes arbitrary pixel in patch $\mathbf{y}$. Each $\mathscr{A}_p(x^i)$ is expressed by:

$$\mathscr{A}_p(x^i) = \sum_{j=1}^{C} w_j \cdot y_j^i,\ p = 1, 2, \cdots, P,\ i = 1, 2, \cdots, q^2. \tag{5.6}$$

where C stands for the number of rules in an ANFIS. Each ANFIS mapping $\mathscr{A}_p(x^i)$ is composed of a set of linear rules, with the following format: If x^i (the input pixel) is A_j, then $y_j^i = p_j x + r_j$. As with the work of [122], the Gaussian function defined as Equation (2.3) is herein chosen as the membership function to describe A_j and to obtain the firing strength of the fuzzy rules in each ANFIS network under training. By combining all the estimated HR image patches, a potential HR image results.

5.2.2.3 Post-processing

Two post-processing techniques are employed in the testing stage, including a non-local means filter that aims to suppress the artefacts and noise caused in the inference process, and a iterative back projection algorithm that ensuring the fidelity of the obtained HR image with the LR image.

A. Non-Local Means (NLM)

To smooth the combination process, the NLM filter is used to suppress undesirable artefacts, as well as to reduce the noise in the ANFIS-constructed HR image. The non-local means algorithm is first developed in [17] for the purpose of image denoising. Generally, the denoised pixel y_i is updated as the weighted average of the similar pixels $\{y_j\}$ whose neighbourhoods are similar to its neighbourhood. The denoising process can be expressed as follows:

$$y_i = \sum_{j \in S} w_{ij} y_j \tag{5.7}$$

where S denotes a certain searching window centred at y_i, the weight w_{ij} is defined as follows:

$$w_{ij} = \frac{1}{Z} exp(- \parallel \mathbf{p}_i - \mathbf{p}_j \parallel^2_{2,a} / h^2) \tag{5.8}$$

where $\mathbf{p}_i$ and $\mathbf{p}_j$ are two image patches centred at y_i and y_j respectively. Such design of the weight relies on the similar degree between $\mathbf{p}_i$ and $\mathbf{p}_j$, larger similar degree will lead to larger weight. $a > 0$ denotes the standard deviation of the Gaussian kernel function, and the parameter h controls the decay of the exponential function. Z is a normalisation constant defined as:

$$Z = \sum_{j} exp(- \parallel \mathbf{p}_i - \mathbf{p}_j \parallel^2_{2,a} / h^2) \tag{5.9}$$

The underlying idea of NLM can be illustrated as shown in Figure 5.3. In the simple example shown in this figure, the central pixel of patch $\mathbf{p}_i$ is updated as the weighted average of itself and its similar pixels. Here, 'similar pixels' are identified as the central pixels of patches $\mathbf{p}_{j1}, \mathbf{p}_{j2}$ and $\mathbf{p}_{j3}$. As can be observed in the figure, patches $\mathbf{p}_{j1}$ and $\mathbf{p}_{j2}$ are very similar with patch $\mathbf{p}_i$, so are assigned with larger weights (w_1 and w_2), while patch $\mathbf{p}_{j3}$ which is rather different to $\mathbf{p}_i$ is assigned with a small weight (w_3).

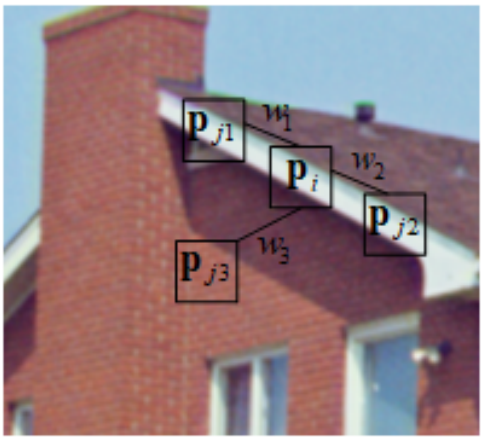


Figure 5.3: Basic idea of NLM – Similar pixel neighbourhoods lead to a large weight (w_1 or w_2), and different neighbourhoods result in a small weight (w_3).

B. Iterative Back Projection (IBP)

Finally, the iterative back projection method [62] is employed to ensure the fidelity with the LR image, contributing to the reduction of the reconstruction error. The IBP is implemented as the following iterative equation:

$$\mathbf{Y}^{t+1} = \mathbf{Y}^t + \lambda * I(\mathbf{X} - D(\mathbf{Y}^t)) \tag{5.10}$$

where $t \in 0, 1, \cdots, T$, T is the maximum number of iterations. $\mathbf{Y}^0$ is initialized as $\mathbf{Y}^0 = \mathbf{Y}$. λ is the step size, $I(\cdot)$ and $D(\cdot)$ are interpolation operator and down sample operator, respectively.

By iteratively implementing the inference and post-processing sub-stages described above, the final output HR image is obtained. The steps of the proposed testing stage can be summarised as given in Algorithm 5.2.

5.3 Experimentation and Validation

To verify the effectiveness of the proposed approach, experimental comparative studies are carried out and the results are analysed both qualitatively and quantitatively.

Algorithm 5.2: Image super resolution via multiple ANFIS mappings.

Input:
Low resolution image $\mathbf{X}$;
Scale factor s;
Learned ANFIS mappings $\{\mathscr{A}_p\}$;
Centroids $\mathbf{V} = \{(\mathbf{v}_p^l, \mathbf{v}_p^h)\}$.

`// Pre-processing`

Do Bicubic interpolation for $\mathbf{X}$;
repeat
  `// Inference with learned ANFIS mappings`
  Divide $\mathbf{X}$ into $q \times q$ patches $\{\mathbf{x}\}$ with overlaps;
  for *each patch* $\mathbf{x}$ *in* $\mathbf{X}$ **do**
    Select an ANFIS mapping $\mathscr{A}_p$ with minimum cost using Equation (5.2);
    Estimate corresponding HR patch $\mathbf{y} = \mathscr{A}_p(\mathbf{x})$;
  end
  Integrate $\{\mathbf{y}\}$ to get $\mathbf{Y}$;
  `// Post-processing`
  Apply NLM method, for every pixel y_i in $\mathbf{Y}$: $y_i = \sum_{j \in S} w_{ij} y_j$;
  Apply iterative back projection: $\mathbf{Y}^{t+1} = \mathbf{Y}^t + \lambda * I(\mathbf{X} - D(\mathbf{Y}^t))$;
  $\mathbf{X} \leftarrow \mathbf{Y}$;
until *maximum iteration reached*;

Output:
Estimated high resolution image $\hat{\mathbf{Y}}$.

5.3.1 Experimental setup

In this experiment-based investigation, 6 typical natural images each with a size of 256×256 are used as the original HR images (ground truth), and the corresponding testing LR images are simulated by down-sampled HR ones using a scale factor of 2 and 3. Parameters in the implemented system are empirically set as follows: the image patch size $q = 9$; the number of clusters $P = 6$, which is also the number of the ANFIS mappings; the number of fuzzy rules in each ANFIS is set to $C = 9$; and in the training of ANFIS, 10,000 data samples (image patch pairs) are used. All the experiments are performed on a laptop with an Intel Core i7-6700 CPU @ 2.6 GHz and 16GB RAM, using MATLAB R2015a.

The performance of the proposed approach is compared with the bicubic interpolation [56], the standard fuzzy rule based method [122], and the popular sparse

representation method [150]. The results of only using the ANFIS mappings without post-processing are also provided. For the colour images used, they are firstly converted into the YCbCr space from the original GRB space, and since the human eyes are more sensitive to the intensity channel (Y), the proposed approach is only applied on the intensity channel.

5.3.2 Evaluation index

To quantitatively access the performance of the proposed work, the Peak Signal-to-Noise Ratio (PSNR) and the Structure SIMilarity (SSIM) indexes are computed. PSNR assesses the reconstruction performance through calculating the Euclidean distance between the original HR image $\mathbf{Y}$ and the estimated HR image $\hat{\mathbf{Y}}$, in which the Mean Square Error (MSE) between them is firstly calculated:

$$MSE = \frac{\| \mathbf{Y} - \hat{\mathbf{Y}} \|_F^2}{MN} \tag{5.11}$$

$$PSNR = 10 \log_{10}(\frac{255^2}{MSE}) \tag{5.12}$$

where M, N are the image length and width respectively, and $\| \cdot \|_F$ stands for the Frobenius norm of matrix. SSIM reflects the degree of structural similarity between the original and the estimated HR image, which is defined by

$$SSIM = \frac{4\mu_{\mathbf{Y}}\mu_{\hat{\mathbf{Y}}}\sigma_{\mathbf{Y},\hat{\mathbf{Y}}}}{(\mu_{\mathbf{Y}}^2 + \mu_{\hat{\mathbf{Y}}}^2)(\sigma_{\mathbf{Y}}^2 + \sigma_{\hat{\mathbf{Y}}}^2)} \tag{5.13}$$

where $\mu_{\mathbf{Y}}$ and $\mu_{\hat{\mathbf{Y}}}$ are the mean value of the original and that of the estimated HR image, respectively; $\sigma_{\mathbf{Y}}$ and $\sigma_{\hat{\mathbf{Y}}}$ are their respective standard deviations; and $\sigma_{\mathbf{Y},\hat{\mathbf{Y}}}$ is the covariation of $\mathbf{Y}$ and $\hat{\mathbf{Y}}$. A lager PSNR means a better result with the best being $+\infty$, while the range of the SSIM index is $[0, 1]$ with its best being 1.

5.3.3 Results and discussion

Visual comparisons on the use of different algorithms are shown in Figure 5.4 ($s = 2$) and Figure 5.5 ($s = 3$), in which the detailed image of a small random patch is provided in the second row of each. Qualitatively, these results show that the system implementing the proposed approach produces images with less artefacts and sharper

Figure 5.4: Visual comparison of results from different methods on image 'Butterfly' with scale factor being 2 – From left to right: (a) HR image. (b) Bicubic. (c) Fuzzy Rule. (d) Sparse Representation. (e) ANFIS. (f) ANFIS & Post-processing.

Figure 5.5: Visual comparison of results from different methods on image 'Child' with scale factor being 3 – From left to right: (a) HR image. (b) Bicubic. (c) Fuzzy Rule. (d) Sparse Representation. (e) ANFIS. (f) ANFIS & Post-processing.

Table 5.1: SR results of different algorithms with scale factor = 2

Image	Index	Algorithm				
		Bicubic	Fuzzy Rule [122]	Sparse [150]	ANFIS	ANFIS & NLM
Lena	PSNR	32.79	34.13	35.04	33.92	**35.14**
	SSIM	0.901	0.911	0.930	0.920	**0.932**
Child	PSNR	33.96	35.44	**36.36**	35.56	36.24
	SSIM	0.927	0.937	0.951	0.948	**0.953**
Girl	PSNR	33.87	**35.37**	35.01	34.68	35.23
	SSIM	0.931	0.933	**0.948**	0.942	**0.948**
Butterfly	PSNR	27.46	30.36	31.23	29.06	**31.41**
	SSIM	0.909	0.944	0.954	0.925	**0.956**
Hat	PSNR	31.73	33.13	34.02	32.87	**34.11**
	SSIM	0.886	0.898	0.921	0.907	**0.922**
House	PSNR	32.10	**34.65**	33.34	32.78	33.49
	SSIM	0.892	0.893	0.909	0.901	**0.910**
Average	PSNR	31.99±2.39	33.85±1.91	34.17±1.77	33.15±2.27	**34.27 ±1.69**
	SSIM	0.908±**0.0183**	0.919±0.0216	0.936±0.0184	0.924±0.0186	**0.937**±0.0185

Table 5.2: SR results of different algorithms with scale factor = 3

Image	Index	Algorithm				
		Bicubic	Fuzzy Rule [122]	Sparse [150]	ANFIS	ANFIS & NLM
Lena	PSNR(dB)	30.10	**31.50**	31.05	30.76	31.33
	SSIM	0.827	0.856	0.851	0.843	**0.859**
Child	PSNR(dB)	30.11	31.41	31.30	31.08	**31.66**
	SSIM	0.835	0.865	0.869	0.863	**0.874**
Girl	PSNR(dB)	32.57	33.27	33.31	33.23	**33.52**
	SSIM	0.877	0.888	0.895	0.893	**0.900**
Butterfly	PSNR(dB)	24.05	26.41	25.44	25.04	**27.07**
	SSIM	0.808	0.885	0.844	0.822	**0.889**
Hat	PSNR(dB)	29.20	30.62	30.10	29.79	**31.11**
	SSIM	0.811	0.844	0.834	0.825	**0.854**
House	PSNR(dB)	29.54	**32.36**	30.30	30.05	30.96
	SSIM	0.835	0.859	0.848	0.839	**0.860**
Average	PSNR(dB)	29.26±2.81	30.93±2.39	30.25±2.62	29.99±2.71	**30.94±2.11**
	SSIM	0.832±0.0248	0.866±**0.0172**	0.857±0.0219	0.848±0.0267	**0.873**±0.0185

details, especially around content boundaries (such as those on the hat brims and eyebrow). The reconstructed images by the proposed approach (in Figure 5.4(f) and Figure 5.5(f)) appear to be closer to the original HR ones (in Figure 5.4(a) and Figure 5.5(a)). The results also illustrate that those resulting images using only multiple ANFIS mappings without post-processing contain irregular artefacts and have block effects, decreasing the visual quality. The joint use of the multiple ANFIS mappings and the post-processing algorithms can effectively suppress or even eliminate such displeasing artefacts, generating HR images with better visual effects.

The results of quantitative comparisons with scale factors 2 and 3 are given in Table 5.1 and Table 5.2 respectively. Best results, corresponding to the maximum values in each line, are listed in bold. It can be observed that the system implementing the proposed approach outperforms the rest in all but two or three cases. Furthermore, the mean value and standard deviation of the PSNR and SSIM indices show the robustness of the proposed system. Additionally, the quantitative comparisons once again illustrate that using multiple ANFIS mappings alone may lead to the reconstructed images containing irregular artefacts and block effects. The integrated use of both multiple ANFIS mappings and post-processing techniques can effectively suppress or in certain cases, eliminate the displeasing artefacts.

It is known that for images, especially structured images with rich details, adopting one single and universal learned mapping for SR is impossible [171]. The above experimental comparisons demonstrate that through K-Means clustering and learning relevant mapping on each cluster, the proposed approach is able to exploit the advantage of selecting the most appropriate mapping with respect to different image patches and hence, to reduce the possibility of introducing addition noise. This strategy improves the system's representation capability over different types of region, thereby contributing to the increase in reconstruction accuracy.

The number of clusters (denoted by P) is an important parameter in the K-means clustering algorithm, which is set to be $P = 6$ empirically in the above experimentation. In order to investigate how the algorithm may behave when the value of P varies, experiments with different P values (from 3 to 9) are conducted here. The performance (PSNR) and training time are shown in Figure 5.6 and Figure 5.7 respectively. It can be seen from the curve of Figure 5.6 that the PSNR values do not vary much when different numbers of clusters (P) are used (although the PSNR value when $P = 6$ is slightly better), showing the robustness of the proposed

algorithm. From Figure 5.7 it can be observed that the algorithm costs the longest training time when $P = 3$. This is because it takes longer to train an ANFIS model with more training data than to do so with less training data. Thus, despite the number of clusters is small, it is more time consuming. At the beginning, the training time reduces as the number of clusters increases. However, when this number increases to a certain point ($P = 6$), the algorithm will involve the training of many models, but consumes as much time as training a few models with a large amount of data. As such, after that point the training time changes slowly. Therefore, by trading off the reconstruction accuracy and the computational cost, the number of clusters is set to be $P = 6$ in this work.

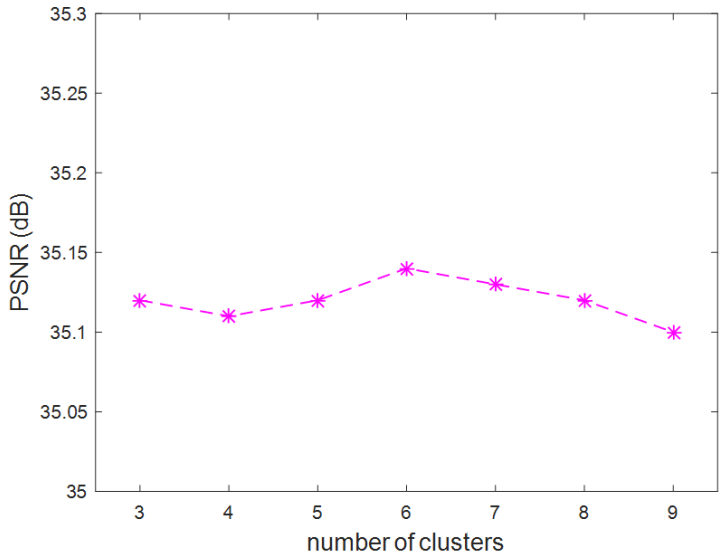


Figure 5.6: PSNR index of using different P (number of clusters in K-means algorithm) on Lena image with scale factor being 2.

5.4 Summary

In this chapter, a single frame image super resolution approach using ANFIS has been proposed, with sufficient training data provided. It works by employing multiple ANFIS networks that learn the mappings between the LR patches and the HR patches. Different ANFIS mappings are learned from an external database of a large number of LR and HR image patch pairs. For each patch of a new input LR image, the

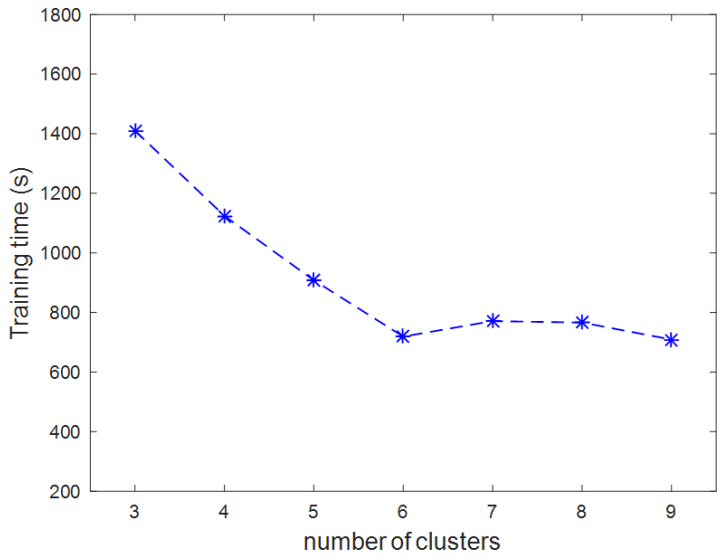


Figure 5.7: Training time of using different P (number of clusters in K-means algorithm).

most relevant trained ANFIS is selected to perform super resolution. The non-local means filter as well as the iterative back projection algorithm are used to refine the resulting image. Comparative experimental results presented have demonstrated the effectiveness of the proposed approach.

Having shown the effectiveness of the learned ANFIS mappings in dealing with the problem of image super resolution with full training data, the next chapter will evaluate the performance of the interpolated ANFIS mappings, when solving the hyperspectral image super resolution task with sparse training data.

Chapter 6

Hyperspectral Image Super Resolution with Sparse Training Data via ANFIS Interpolation

THE proposed ANFIS interpolation approaches have been proven to be successful in dealing with regression tasks over benchmark datasets with insufficient training data, as reported in Chapters 3 and 4. This current chapter presents a systematic application of the proposed ANFIS interpolation techniques for the problem of hyperspectral image super resolution.

Hyperspectral image is wildly used in real world applications, such as terrain detection [32, 45], military surveillance [46] and medical diagnosis [102], owing to its multi-spectral property. The super resolution of hyperspectral images is also very important for providing high quality images for those special applications. As mentioned previously (in Section 2.3.2), the learning based SR methods require an extra training dataset with large number of training images to learn an effective model. However, the special imaging mechanism makes the acquire of sufficient hyperspectral images for training a very difficult task in certain applications. This practical training data shortage problem makes the hypersectral image super resolution a more difficult task compared with natural image super resolution. In order to deal with this challenging problem, in this chapter, a new hyperspectral image super resolution approach with sparse training data is developed by introducing the proposed ANFIS interpolation techniques.

The rest of this chapter is organised as follows. For academic completeness, Section 6.1 presents a brief introduction to hyperspectral image and an overview of hyperspectral image super resolution methods. Section 6.2 details the proposed ANFIS Interpolation based hyperspectral ISR approach, along with its training and testing phases. Experimental investigation is reported and discussed in Section 6.3, including the experimental setup, performance criteria and the experimental results. Section 6.4 concludes the chapter.

6.1 Hyperspectral Image Super Resolution

In this section, the background knowledge related to the work in this chapter is presented. The concept of hyperspectral image is introduced firstly, then existing typical hyperspectral image super resolution methods are reviewed.

6.1.1 Hyperspectral image (HSI)

Hyper-Spectral Image (HSI) is defined as an image cube with a number of image channels (or image bands). Different with traditional single-channel black-and-white image or three-channel RGB color image, an HSI usually contains dozens or even hundreds of spectral bands (as shown in Figure 6.1). Another equivalent interpretation of HSI is "the hyperspectral imaging sensor acquires a spectral vector with dozens or hundreds of elements from every pixel in a given scene, the result is the so called hyperspectral image" [15].

For a given HSI $\mathbf{X} \in R^{M \times N \times L}$, different forms of representation can be used:

- *Spectral representation:* each image pixel in an HSI $\mathbf{X}$ can be represented as a vector with L elements: $x \in R^L$, and there are totally $M \times N$ pixels.
- *Spatial representation:* each image band (or channel) is a two dimensional image matrix $X_j \in R^{M \times N}$, with L image bands in total.

Both of these two types of representation are commonly used in processing an HSI, and when processing a single image band (or pixel), the neighbouring bands

(a) Black-and-white (b) RGB (c) Hyper-spectral

Figure 6.1: Hyperspectral Image (HSI).

(or pixels) are usually taken into consideration due to the high correlation among neighbouring bands (or pixels).

Different from the traditional three-channel colour images that can be obtained easily using optical cameras, the acquire of hyperspectal images is more complex. HSIs are usually obtained through the use of special imaging equipments such as hyperspectral cameras, which acquires each image band within a certain spectral response interval, such that different image bands in an HSI are acquired with different spectral responses. This special imaging mechanism makes different image bands within the same HSI may show different information, which is very useful in special real world applications. For example, Figure 6.2 shows three band images (band 10, band 100 and band 200) of a terrain HSI named 'cuprite', from which it can be seen that the band images reflect different image details. In band 10, the bottom-left part of the image is almost black, but some textures can be observed in band 100 while in band 200 the image textures are much clearer. These different information provided by different bands will be helpful for the subsequent terrain detection tasks.

6.1.2 HSI super resolution (HSI SR)

Since the wide use of high quality HSIs, the super resolution of HSI is becoming one of the most popular topics in the recent years. Due to the special property of HSI as mentioned above, the HSI super resolution is more complex than the traditional black-and-white images or the RGB ones. Generally, hyperspectral image super

(a) Band 10 (b) Band 100 (c) Band 200

Figure 6.2: Band images of an HSI

resolution (HSI SR) [3] refers to the techniques that enlarge each band of an given low resolution HSI (LR HSI) to get a high resolution HSI (HR HSI) with HR images in every bands.

The HSI SR approaches in the existing literatures can be approximately classified into three categories: multi-frame HSI SR, single-frame HSI SR, and the fusion-based HSI SR algorithms. In multi-frame HSI SR approaches, multiple low resolution hyperspectral observations from the same scene are detected, geometrically registered and combined to generate one image of high spatial resolution. These multiple observations of the same scene can be obtained from overlapping flight lines, multi-angle data [21], or multiple instances of different time [170]. Similar with the traditional multi-frame image SR technique which is introduced previously in Section 2.3, multi-frame HSI SR approaches need sub-pixel level registration for every band, which is much more time consuming than that of the traditional images. Fusion-based approaches fuses the LR HSI with other available image sources of high spatial resolution, acquired by other sensors (e.g. multi-spectral image sensors or panchromatic image sensors mounted on satellites). A large majority of fusion-based methods injects the high spatial information of the panchromatic image into the hyperspectral image bands [100], which is also called HSI pan-sharpening. Another branch of fusion-based approaches is to adopt spatial-spectral fusion between the LR hyperspectral image and an HR multispectral image that can deliver the required spectral information [151, 160, 174].

According to the two forms of HSI representation mentioned in Section 6.1.1, there are generally three classes of approaches in single-frame HSI SR category. 1) Spatial based approaches [119, 123, 162]: These approaches use the spatial representation of an HSI in which the given LR HSI is considered as L LR band images. Those LR band images are then enlarged either independently or collaboratively. 2) Spectral based approaches [109, 110]: The spectral based approaches also termed as 'hyperspectral unmixing'. In these approaches, the given LR HSI is decomposed into a number of pixels according to the spectral representation of an HSI. Each pixel is assumed as a linear mixture of several pure image pixels [61]. Therefore, through the use of the spectral unmixing techniques [49, 81], the number of pixels of an HSI will be increased such that the image is enlarged. Both of these two kinds of approaches have their advantages and disadvantages. For instance, the spatial based approaches don not take the spectral information into consideration,

while on the contract, the spectral based approaches also do not consider the spatial correlation among pixels. 3) Spatial-Spectral based approaches [57, 176]: In order to overcome these disadvantages in the spatial or spectral based methods, the third class of HSI SR approaches is proposed, by combining both the spatial and the spectral approaches. These approaches always iteratively conduct the spatial regulation step and the spectral regulation step, in an effort to maintain both the spatial and spectral information.

Current HSI HR approaches usually assume that there exit sufficient training HSIs. However, different from the natural images that usually can be obtained very easily in the daily life by using optical cameras, the hyperspectal images are obtained using special imaging equipments that leads to the process of getting an image very time consuming. Also, usually hyperspectral cameras are carried on special machines such as satellite, aircraft or medical device, resulting in that the acquirement of images are often effected by the weather, noise or other random factors. All these reasons make the acquire of sufficient hyperspectral images a very difficult task. Therefore the proposed ANFIS interpolation method is used in this chapter to perform HSI SR task under sparse training data situations. In this chapter, the spatial based single frame super resolution strategy is employed in the proposed work.

6.2 HSI SR with Sparse Data through ANFIS Interpolation

In this section the proposed approach for HSI SR is described in detail. The whole process is divided into two stages: the training stage and the testing stage. In the training stage, as the same with the image SR with full training data described in Chapter 5, the whole training set are grouped into a small number of sub-sets. Among these subsets, some are with full training data while some are with sparse data. An individual ANFIS mapping is then learned for each subset with full data, while for those with sparse data, the corresponding ANFIS mapping is constructed using the ANFIS interpolation method. In the testing stage, the estimated HR image is obtained by inferencing using these learned or interpolated ANFIS mappings.

6.2.1 Training stage

The proposed training process is shown in Figure 6.3, and the outline of the training process is shown in Algorithm 6.1.

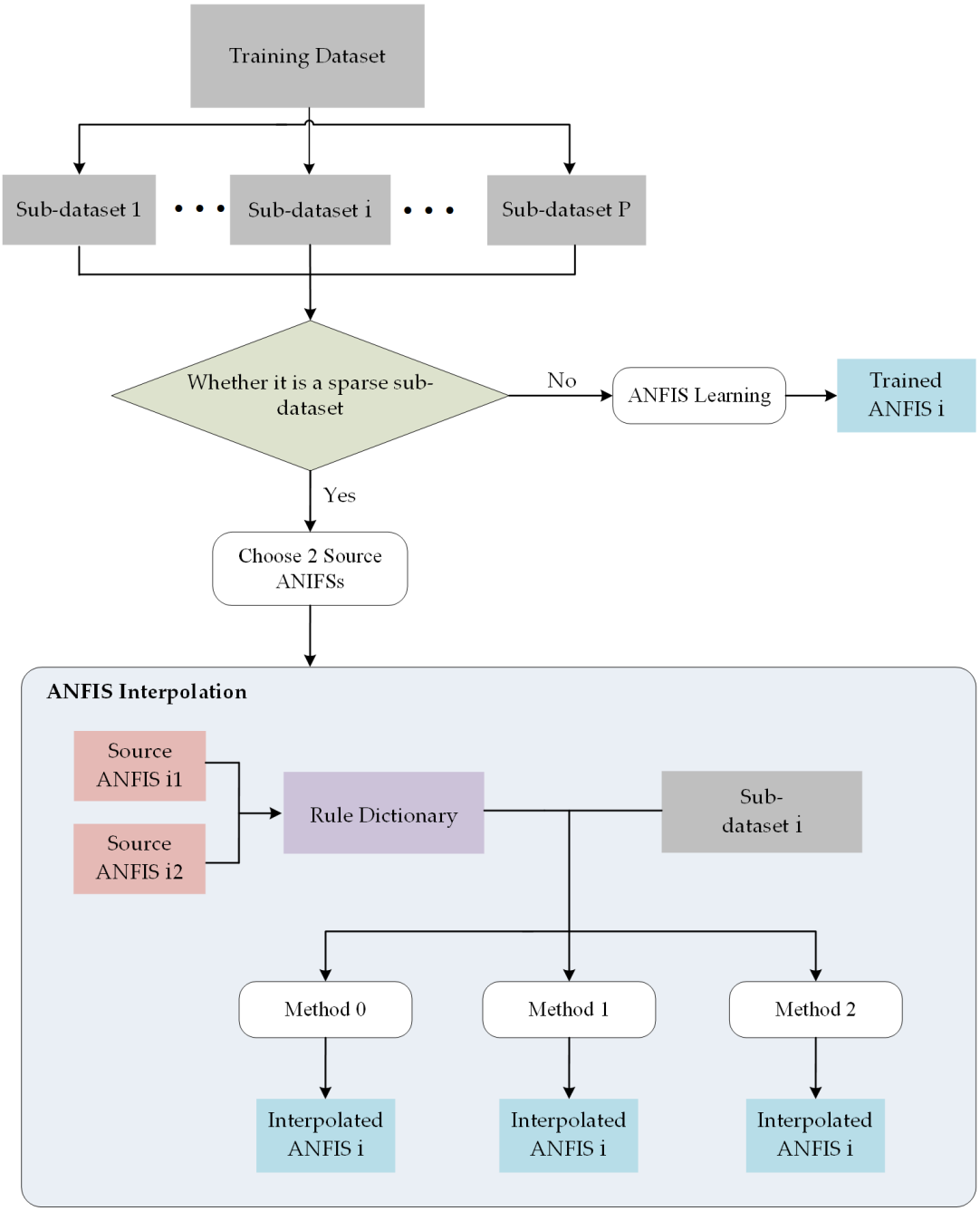


Figure 6.3: Flowchart of the training process

Algorithm 6.1: Training stage for HSI SR with sparse data.

Input:
Training HSI dataset: $\{\mathbf{X}^l, \mathbf{Y}^h\}$;
Scale factor: s;
Interpolation method mode: $Method$;

$S = \varnothing$;
Divide training dataset into P sub-sets $\{\mathbf{P}_i; i = 1, 2, \cdots, P\}$ with centroids $\{V_i; i = 1, 2, \cdots, P\}$;
for $i = 1$ *to* $i = P$ **do**
    if $\mathbf{P}_i$ *contains sufficient data* **then**
        Train ANFIS $\mathscr{A}_i$ with standard training method;
    else
        $S \leftarrow \{S, i\}$;
    end
end
for *each* $i \in S$ **do**
    Choose 2 closest ANFISs $\mathscr{A}_{i1}$ and $\mathscr{A}_{i2}$ as source ANFISs;
    Generate rule dictionary D by extracting rules from $\mathscr{A}_{i1}$ and $\mathscr{A}_{i2}$;
    switch *Method* **do**
        case *0* **do**
            Interpolate ANFIS $\mathscr{A}_i$ using 'Method 0';
        case *1* **do**
            Interpolate ANFIS $\mathscr{A}_i$ using 'Method 1';
        case *2* **do**
            Interpolate ANFIS $\mathscr{A}_i$ using 'Method 2';
    end
end

Output:
Multiple learned ANFIS mappings $\{\mathscr{A}_i\}$;
Centroids $\{V_i\}$.

6.2.1.1 Training dataset generation

The training dataset consists of a number of low resolution and high resolution hyperspectral image pairs $\{\mathbf{X}^l, \mathbf{Y}^h\}$, with a scale factor s. Here each $\mathbf{X}^l \in R^{m \times n \times L}$, each $\mathbf{Y}^h \in R^{M \times N \times L}$, the length of LR-HSI $\mathbf{X}^l$ and that of HR-HSI $\mathbf{Y}^h$ have the relationship $M = sm$, and similarly the relation between the width of $\mathbf{X}^l$ and that of $\mathbf{Y}^h$ is $N = sn$. The training dataset can be generated as the following steps: 1) A number of high quality HR HSIs are collected, forming the $\{\mathbf{Y}^h\}$; 2) For each $\mathbf{Y}^h$, its corresponding low resolution image pair $\mathbf{X}^l$ is constructed as follows: every band of $\mathbf{Y}^h$ (denoted as $Y^h_j, j = 1, \ldots, L$) is firstly down sampled with the scale factor s, and then up sampled

using the bicubic interpolation method; 3) Combine all the image pairs to form the training dataset.

6.2.1.2 Train ANFIS mappings for source domains

In the training stage, the previously generated training dataset is firstly divided into several subsets according to the different characteristics of band images. However, the number of training samples between different subsets can be rather imbalanced. That is, for certain subsets there may be sufficient training samples available whilst for others there may not. For those with sufficient training samples, it is easy to learn effective ANFIS mappings. Yet, for those without sufficient samples, it can be rather difficult to derive quality models. Denote the subsets with full training data as source domains, the subsets with sparse training data as target domains. The proposed approach adopts different strategies for source domains and target domains. In source domains, the ANFIS mappings are directly trained. While in target domains, the relevant mappings are interpolated by the support of two source ANFIS mappings in source domains.

Suppose that the given training dataset is divided into P subsets $\{\mathbf{P}_i\}$, and the index set of the subsets with sparse training data is denoted by S. For the subsets with sufficient training data (i.e. $\{\mathbf{P}_i; i \notin S\}$), the learning of ANFIS mappings adopts the standard ANFIS learning method which is described in Chapter 5, so is omitted here.

6.2.1.3 Interpolate ANFIS mappings for target domains

For those subsets with sparse data, the interpolation procedure of ANFIS mappings is described in detail as follows. For an arbitrary subset with sparse data $\mathbf{P}_i, i \in S$, two closest ANFISs is selected firstly as source ANFISs according to the distances d_j between the center of the sparse training data $V_i, i \in S$ and the center of the subsets with full training data $V_j, j \notin S$. The distance d_j is calculated as:

$$d_j = \sqrt{(V_i - V_j)^2} \tag{6.1}$$

where the ANFISs related to the two subsets with the smallest distances d_j are chosen as the two source ANFISs, denoted as $\mathscr{A}_{i1}$ and $\mathscr{A}_{i2}$. With the two source ANFISs,

and the sparse training data in $\mathbf{P}_i$, the ANFIS mapping $\mathscr{A}_i$ can be interpolated using the proposed ANFIS interpolation method.

Rule dictionary generation

According to the proposed ANFIS interpolation method, the first step is to generate a rule dictionary, which is used to store the extracted fuzzy rules from the source ANFISs $\mathscr{A}_{i1}$ and $\mathscr{A}_{i2}$. For the present application of HSI SR, assuming that the ith rule can be expressed in the following format:

$$R_i : if\ \ x\ is\ A_i,\ \ then\ y_i = p_i x + r_i \tag{6.2}$$

where x denotes an input pixel from an arbitrary band of the LR HSI, A_i is the corresponding fuzzy set, y_i represents the output of the ith rule (which contributes to the final outcome of the HR HSI being constructed). The rule dictionary $D = \{D_a, D_c\}$ is generated by reorganising the above extracted fuzzy rules, with the antecedent part D_a and the consequent part D_c each collecting all the antecedents and consequents of those rules. Thus, D_a consists of the antecedent parts of all the rules:

$$D_a = \{A_1\ A_2\ \cdots\ A_n\} \tag{6.3}$$

where n is the number of rules in the rule dictionary. And the consequent part D_c consists of the consequents of the rules, where each column denotes the linear coefficients in the consequent part of a certain rule:

$$D_c = \begin{bmatrix} p_1 & p_2 & \cdots & p_n \\ r_1 & r_2 & \cdots & r_n \end{bmatrix} \tag{6.4}$$

Interpolation using three ANFIS interpolation methods

There are totally three ANFIS interpolation methods proposed in this thesis, including the initial one described in Chapter 3 (termed as 'Method 0'), and the GA based two methods introduced in Chapter 4 with either the rule type chromosome (termed as 'Method 1') or the ANFIS type chromosome (termed as 'Method 2').

For 'Method 0', the small number of training data in $\mathbf{P}_i$ is divided into C clusters using the K-means algorithm. For the centre of each cluster, a new fuzzy rule is interpolated by exploiting the LLE algorithm. Then, by aggregating all interpolated

rules, an intermediate ANFIS results. Finally the intermediate ANFIS is fine-turned to get the final interpolated output.

Similar with 'Method 0', in 'Method 1' and 'Method 2', the small number of training data in $\mathbf{P}_i$ are also divided into C clusters using the K-means algorithm. The centre of each cluster is used as a seed for generating a number of individuals, which are utilized as observations for interpolating a number of new rules subsequently. Those interpolated rules form the initial population, either in individual rule form (Method 1) or in gathered form (Method 2). Then the initial population will be evolved through a GA process and finally output the interpolated ANFIS mapping.

6.2.2 Testing stage

Given a test LR HSI $\mathbf{X} \in R^{m \times n \times L}$, the testing stage is to test the performance of the learned or interpolated ANFIS mappings $\{\mathscr{A}_i\}$. At the beginning, a preprocessing step is conducted to up scale the LR HSI to the same size with the desired HR HSI using bicubic interpolation, resulting in $\mathbf{X} \in R^{M \times N \times L}$. Next, the image details will be enhanced by the ANFIS mappings. Particularly, each image band $X_j \in R^{M \times N}$ is firstly vectorized as a vector format, which is also denoted by X_j and here $X_j \in R^{MN}$. A corresponding ANFIS mapping is then selected to perform inference for this band, by solving the following optimization formula:

$$i = \min_i \| X_j - V_i \|_2^2 \tag{6.5}$$

where $\{V_i\}$ denotes the centroids of the subsets from the training stage, and the ith ANFIS $\mathscr{A}_i$ is chosen as the relevant mapping for inference. Therefore the corresponding HR image band Y_j can be obtained by:

$$Y_j = \mathscr{A}_i(X_j) \tag{6.6}$$

Finally, the post-processing technique (Iterative Back Projection, IBP) is used here to finish the testing stage and then produce the estimated HR HSI. Note that different with the post-processing process in Chapter 5 where a NLM filter is also employed, here only the IBP operator is utilized to perform the post-processing. This is in order to avoid that the details of the resulting images are over-smoothed by the NLM filter, so that the differences between the results of the interpolated and non-interpolated ANFIS models can be clearly shown. The testing stage is summarized as Algorithm 6.2.

Algorithm 6.2: Testing stage for HSI SR with sparse data.

Input:
Learned and interpolated ANFIS mappings $\{\mathscr{A}_i\}$;
Testing LR image $\mathbf{X}$;
Scale factor s;
Centroids $\{V_i\}$.

Pre-processing: Upscale the testing LR HSI $\mathbf{X}$ by bicubic interpolation;
for *each band of upscaled image* X_j **do**
    Choose relevant ANFIS model $\mathscr{A}_i$;
    Inference using corresponding ANFIS: $Y_j = \mathscr{A}_i(X_j)$;
end
Integrate HR image bands $\{Y_j\}$ to form HR image $\mathbf{Y}$;
Post-processing: Apply IBP $\mathbf{Y}^{t+1} = \mathbf{Y}^t + \lambda * I(\mathbf{X} - D(\mathbf{Y}^t))$

Output:
HR image $\hat{\mathbf{Y}}$.

6.3 Experimentation and Discussion

This section presents and discusses the results of experimental comparison. Section 6.3.1 gives the basic experimental settings, Section 6.3.2 introduces the performance criteria for hyperapectral image SR used in the thesis, and Section 6.3.3 presents the experimental results over the CAVE HSI dataset.

6.3.1 Experimental setup

The CAVE dataset [20] is used here for the HSI SR experiments. The CAVE dataset contains 31 HSIs of a wide variety of real-world materials and objects, each of which consists of 31 image bands, coving the range of wavelength from 400nm to 700nm. In the experiments, 26 HSIs are used as the training dataset, and the remaining 5 images are used for testing. The training dataset is manually divided into $P = 5$ sub-datasets based on the fact that different image bands of HSIs usually have different characteristics, and the characteristics of the neighbouring bands are usually similar. The division of the 5 sub-datasets is as follows, the first 1 to 6 bands of all HSIs in the training dataset form the first sub-dataset ($\mathbf{P}_1$), then every 6 neighbouring bands of all HSIs in the training dataset form a sub-dataset ($\mathbf{P}_i, i = 2, 3, 4$), the last sub-dataset ($\mathbf{P}_5$) contains the remaining 7 image bands.

Assume that the second and the fourth sub-datasets are with sparse data, the other 3 sub-datasets are with full training data. The simulation of the sparse sub-datasets is done by manually delete a large portion of randomly selected image pixels. So in this experiments the ANFIS mappings $\mathscr{A}_1, \mathscr{A}_3, \mathscr{A}_5$ are learned by the standard ANFIS learning method with full training data, while the ANFIS mappings $\mathscr{A}_2, \mathscr{A}_4$ are interpolated with sparse training data. The partition of bands that forms the 5 subsets are illustrated in Figure 6.4.

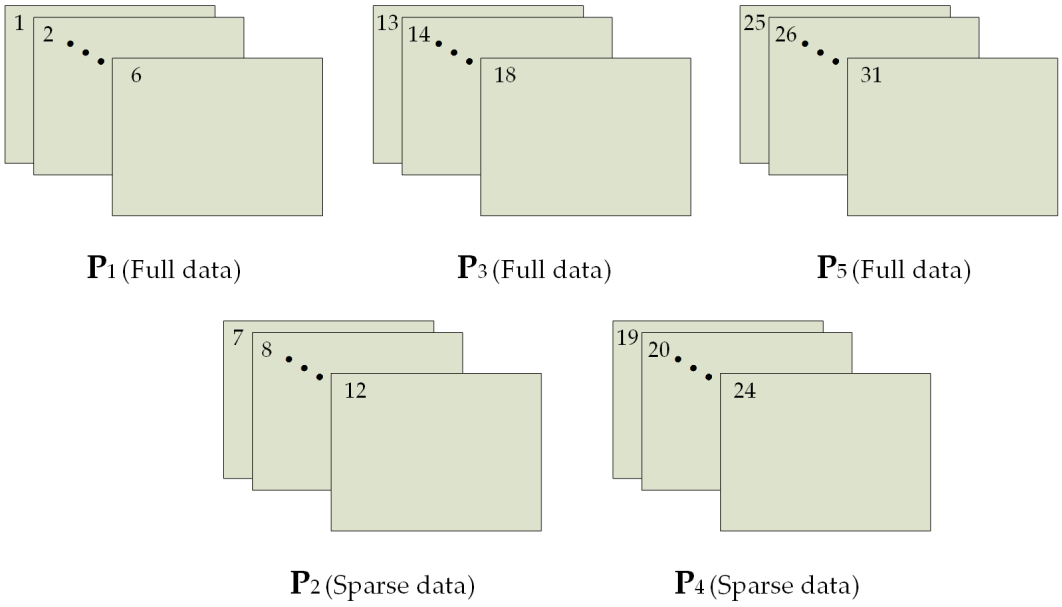


Figure 6.4: Illustration of the 5 subsets

In the experiments, the LR HSIs are enlarged with the scale factor $s = 2$ and $s = 3$ respectively. Similarly with the proposed ANFIS interpolation method described in Chapter 3 and Chapter 4, the triangular membership function is employed in this super resolution problem. Some parameters used in the experiments are set as follows. The number of rules in each ANFIS is set to $C = 6$. In the interpolation process, the number of closest rules is set to $K = 3$.

Five methods are compared in the experiments: 1) Experiments with full training data for all the subsets, which is used as a reference model denoted as 'M-full' hereafter; The other four methods are with sparse training data in some of the subsets. 2) Experiments with sparse training data using the original ANFIS learning method without interpolation, denoted as 'M-sparse' hereafter; 3) Experiments with sparse training data using the initial ANFIS interpolation method, denoted

as 'Method 0' hereafter; 4) Experiments with sparse training data using the rule chromosome based evolutionary ANFIS interpolation method, denoted as 'Method 1' hereafter; 5) Experiments with sparse training data using the ANFIS chromosome based evolutionary ANFIS interpolation method, denoted as 'Method 2' hereafter.

6.3.2 Performance criteria

Different from the two-dimensional natural images used in Chapter 5, the HSIs are image cubes. Therefore the commonly used measures (PSNR and SSIM) can not be directly used. Instead, the average PANR (A-PSNR), the average SSIM (A-SSIM), as well as the RMSE indices, which are designed for evaluating performance of the HSI SR methods, are employed to quantitatively evaluate the performance of different SR methods. The RMSE (Root-Mean-Squared Error) for this HSI SR problem is defined as:

$$RMSE = \sqrt{\frac{\| \bar{\mathbf{Y}} - \hat{\bar{\mathbf{Y}}} \|_F^2}{M \times N \times L}} \tag{6.7}$$

where M, N, L are the image length, image width and number of image bands respectively, and $\| \cdot \|_F$ stands for the Frobenius norm of matrix. $\bar{\mathbf{Y}} \in R^{L \times MN}$ and $\hat{\bar{\mathbf{Y}}} \in R^{L \times MN}$ denotes the transformed two dimensional matrices of the ground truth HR image $\mathbf{Y}$ and the estimated HR image $\hat{\mathbf{Y}}$ respectively. The A-PSNR index assesses the model results by computing the average PSNR among all image bands, which is calculated as follows:

$$A-PSNR = \frac{1}{L} \sum_{j=1}^{L} PSNR_j \tag{6.8}$$

where L denotes the number of image bands, $PSNR_j$ is the PSNR value of the jth band that is computed using Equation (5.11) and (5.12). Similarly, the A-SSIM (Average Structure SIMilarity) index is the average SSIM among all image bands, which is defined by the following:

$$A-SSIM = \frac{1}{L} \sum_{j=1}^{L} SSIM_j \tag{6.9}$$

where $SSIM_j$ denotes the SSIM of the jth image band which is calculated using Equation (5.13). The range of the A-SSIM index is $[0, 1]$. Generally, a smaller RMSE and a lager A-PSNR or A-SSIM means a better result.

6.3.3 Results and discussion

The experimental results of scale factor being 2 are listed in Table 6.1, and that of scale factor being 3 are listed in Table 6.2. The average performance of the 5 testing images are shown in the last row, represented by the 'Mean ± Standard Deviation (SD)' form.

Table 6.1: HSI SR results of different algorithms with scale factor = 2

Image	Index	Algorithm				
		M-full	M-sparse	Method 0	Method 1	Method 2
Balloons	RMSE	1.482	4.102	1.700	1.547	1.743
	A-PSNR	44.711	35.871	43.520	44.343	43.304
	A-SSIM	0.9898	0.9839	0.9876	0.9877	0.9879
Toy	RMSE	4.431	7.448	4.751	4.519	4.773
	A-PSNR	35.201	30.689	34.595	35.029	34.555
	A-SSIM	0.9540	0.9400	0.9508	0.9507	0.9517
Flowers	RMSE	2.126	2.154	2.146	2.154	2.129
	A-PSNR	41.578	41.467	41.499	41.465	41.568
	A-SSIM	0.9755	0.9737	0.9744	0.9740	0.9751
Painting	RMSE	3.349	3.624	3.316	3.251	3.275
	A-PSNR	37.632	36.947	37.719	37.891	37.827
	A-SSIM	0.9338	0.9265	0.9271	0.9285	0.9296
Peppers	RMSE	1.906	1.959	1.879	1.908	1.876
	A-PSNR	42.531	42.287	42.655	42.520	42.664
	A-SSIM	0.9856	0.9831	0.9841	0.9838	0.9851
Average	RMSE	2.659 ±1.209	3.857 ±2.208	2.758 ±1.279	**2.675 ±1.211**	2.759 ±1.277
	A-PSNR	40.333 ±3.846	37.452 ±4.689	39.997 ±3.745	**40.249** ±3.748	39.983 **±3.702**
	A-SSIM	0.9677 ±0.0234	0.9614 ±0.0264	0.9648 ±0.0255	0.9649 ±0.0249	**0.9658 ±0.0247**

The average RMSE values are herein taken as the basis upon which to analyse the results. The 'M-full' model in Table 6.1 is trained with sufficient training data,

Table 6.2: HSI SR results of different algorithms with scale factor = 3

Image	Index	Algorithm				
		M-full	M-sparse	Method 0	Method 1	Method 2
Balloons	RMSE	4.457	5.396	4.671	4.453	4.461
	A-PSNR	35.151	33.489	34.742	35.157	35.143
	A-SSIM	0.9641	0.9500	0.9485	0.9584	0.9578
Toy	RMSE	10.511	12.916	10.699	10.714	10.619
	A-PSNR	27.698	25.908	27.543	27.532	27.609
	A-SSIM	0.8712	0.8516	0.8557	0.8630	0.8633
Flowers	RMSE	3.649	3.931	3.804	3.704	3.689
	A-PSNR	36.887	36.240	36.527	36.757	36.794
	A-SSIM	0.9313	0.9247	0.9233	0.9235	0.9232
Painting	RMSE	4.597	5.076	4.915	4.798	4.799
	A-PSNR	34.881	34.021	34.301	34.509	34.508
	A-SSIM	0.8790	0.8136	0.8578	0.8609	0.8615
Peppers	RMSE	3.016	3.356	3.179	3.137	3.104
	A-PSNR	38.543	37.616	38.084	38.202	38.291
	A-SSIM	0.9619	0.9517	0.9467	0.9505	0.9507
Average	RMSE	5.246 ±3.012	6.135 ±3.880	5.453 ±**3.013**	5.361 ±3.061	**5.334** ±3.027
	A-PSNR	34.632 ±4.147	33.455 ±4.537	34.239 ±**4.034**	34.431 ±4.116	**34.469** ±4.109
	A-SSIM	0.9215 ±0.04438	0.8983 ±0.06240	0.9064 ±**0.04641**	0.9112 ±0.04684	**0.9113** ±0.04648

and is used as a reference model. From the the last row of Table 6.1, it can be observed that the 'M-full' model gives the best result, with the smallest 'RMSE' value. The best average results of the other four ANFIS models that are constructed with insufficient training data ('M-sparse', 'Method 0', 'Method 1' and 'Method 2') are shown in bold. Compared with the outcome of running the 'M-full' model, the result of the 'M-sparse' model degrades heavily due to the training data shortage. The average RMSE value increased to 3.857 from the 2.659 of the reference model. Using the same sparse training data, the initial ANFIS interpolation method proposed in Chapter 3 ('Method 0') gives much more improved result (with the RMSE value

reduced to 2.758). The evolutionary ANFIS interpolation method with rule-typed chromosome further improves the result. The RMSE value reduced to 2.675, which is very close to that of the reference model trained with full data (2.659).

Similar conclusion goes to the standard deviation values. Due to the lack of sufficient training data, the performance of the 'M-sparse' model is not stable. It gives the largest and also the worst standard deviation (2.208), which is much larger than the SD value of the reference model (1.209). The three interpolation based models provide more stable results, with much closer standard deviation values to the SDs of 'M-full' model.

The super resolution problem with scale factor being 3 is harder than that of scale factor being 2, as more information will be needed to construct the high resolution image. So compared with Table 6.1, the average performance in the last row of Table 6.2 is worse. By analysing the experimental results in Table 6.2, it can be concluded that the 'M-sparse' method performs the worst, giving the largest RMSE values and the smallest A-PSNR and A-SSIM values among the five methods. The three ANFIS interpolation methods significantly improve the super resolution performance. This is verified by the minor difference between the result of the interpolated ANFISs and that of the reference model. For example, the average RMSE value of 'Method 2' (5.334 ± 3.027) is very close to that value (5.246 ± 3.012) produced by the reference method. Comparing the performance among the three interpolated ANFISs themselves ('Method 0', 'Method 1' and 'Method 2'), the two evolutionary ANFIS interpolation methods (especially the one with ANFIS-based chromosome) give better results than the 'Method 0' model.

The visible results are shown in Figure 6.5 - Figure 6.8. In order to show the details of the results using different methods, two randomly selected patches from the resulting images are shown in the second row and third row respectively. It can be observed from these patches that in the resulting image using the original ANFIS with sparse data, there are obvious noise and bad edges, and that running the interpolated ANFISs leads to significant improvement. For example, focusing on the detailed patches of HSI 'Balloons' (as shown in the second and third rows of the Figure 6.8), there are unpleasing circles in the results of 'M-sparse' (ANFIS trained with sparse data). The three interpolation based methods ('Method 0', 'Method 1' and 'Method 2'), especially the two evolutionary interpolation methods ('Method 1' and 'Method 2') obviously eliminate those circles, making the results much closer to that of the 'M-full' model and even the ground truth.

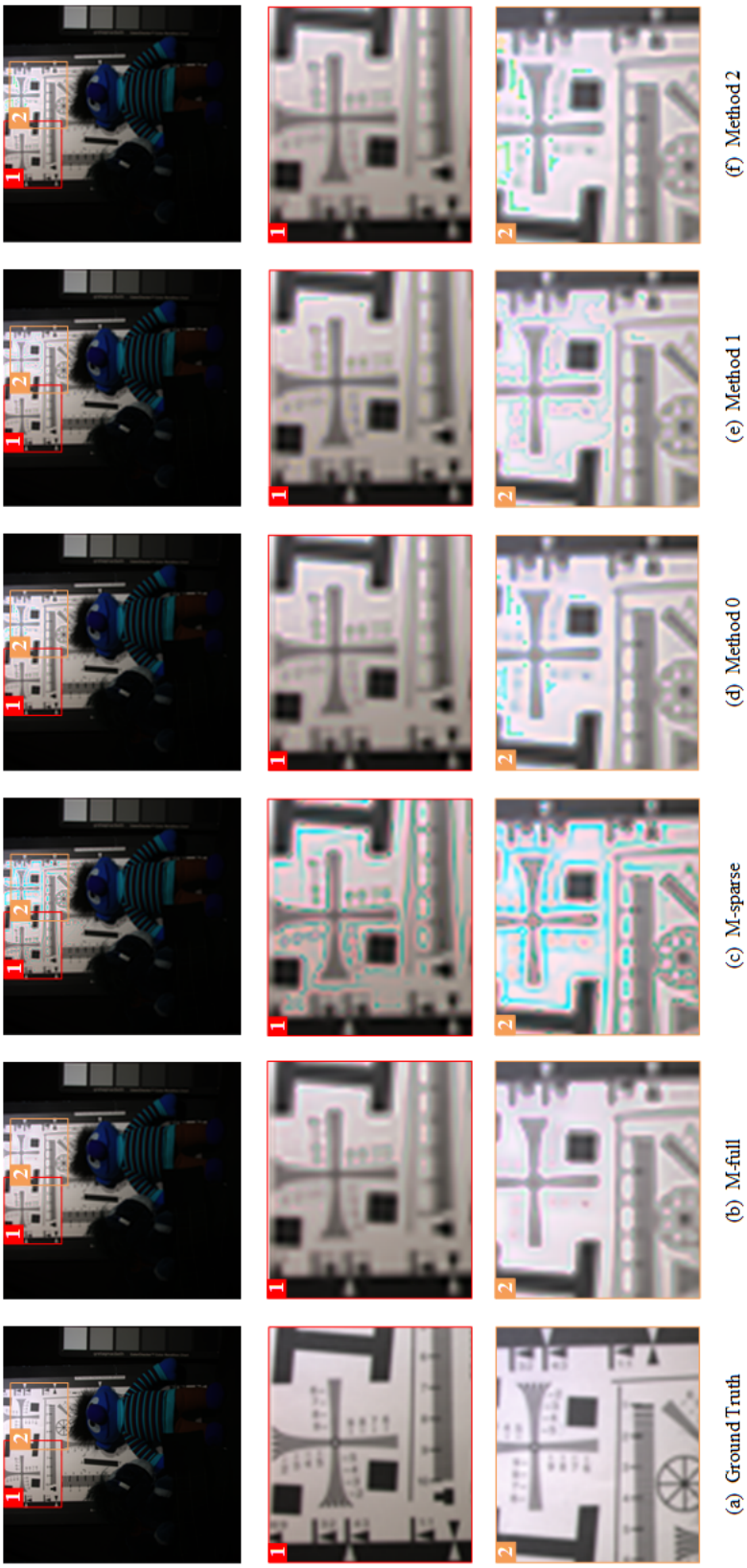


Figure 6.5: Super resolution results using different ANFISs on HSI 'toy' with scale factor being 2.

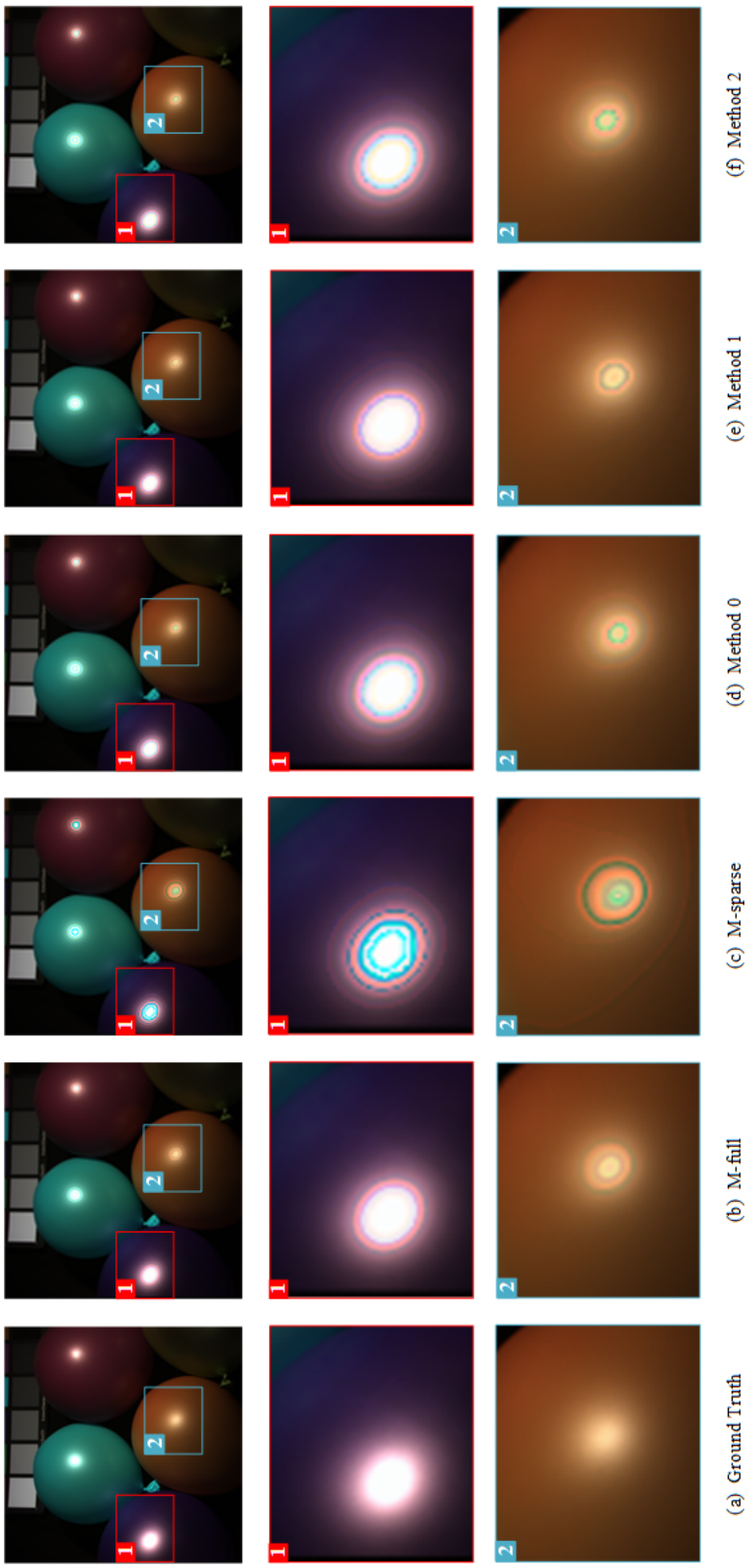


Figure 6.6: Super resolution results using different ANFISs on HSI 'balloons' with scale factor being 2.

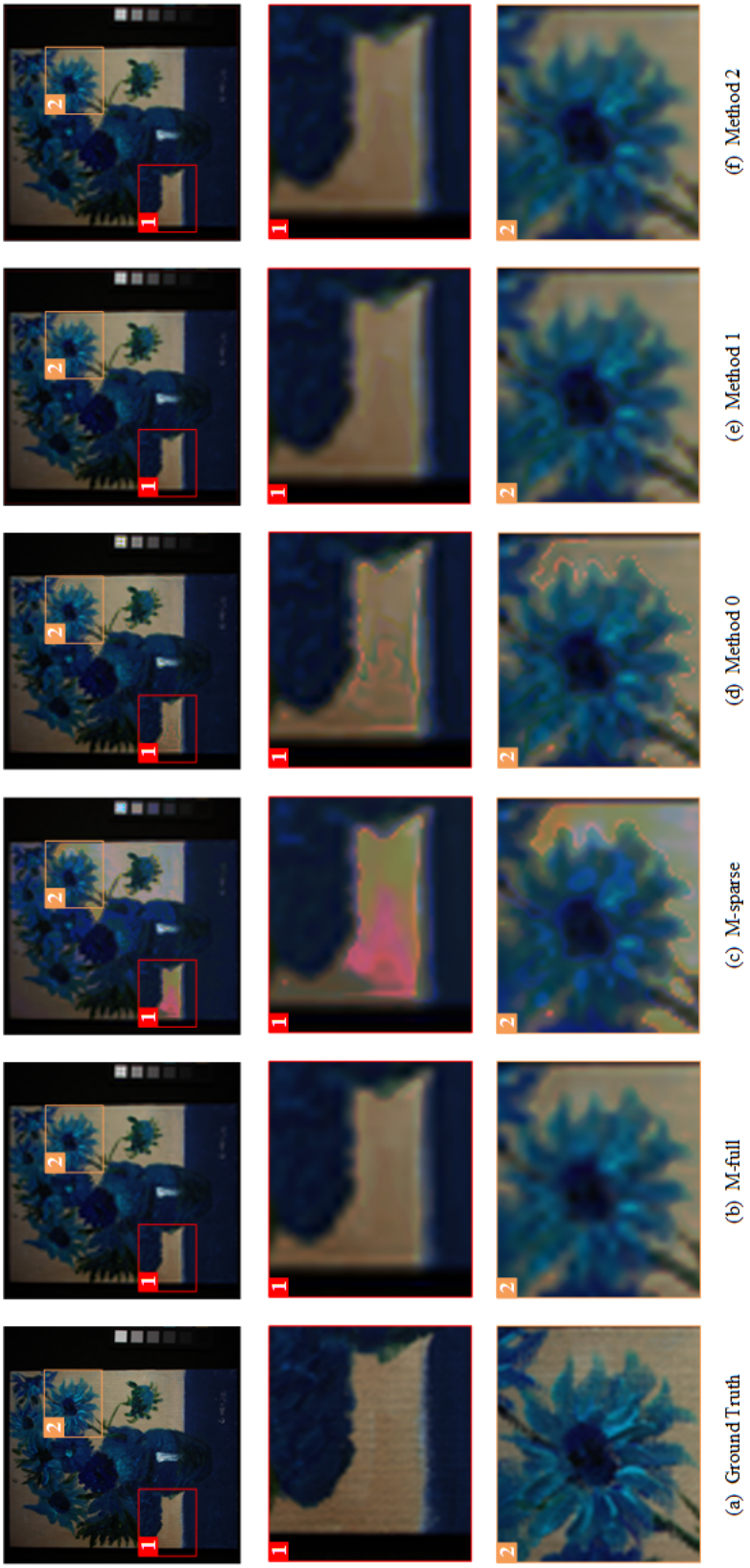


Figure 6.7: Super-resolution results using different ANFISs on HSI 'flowers' with scale factor being 3.

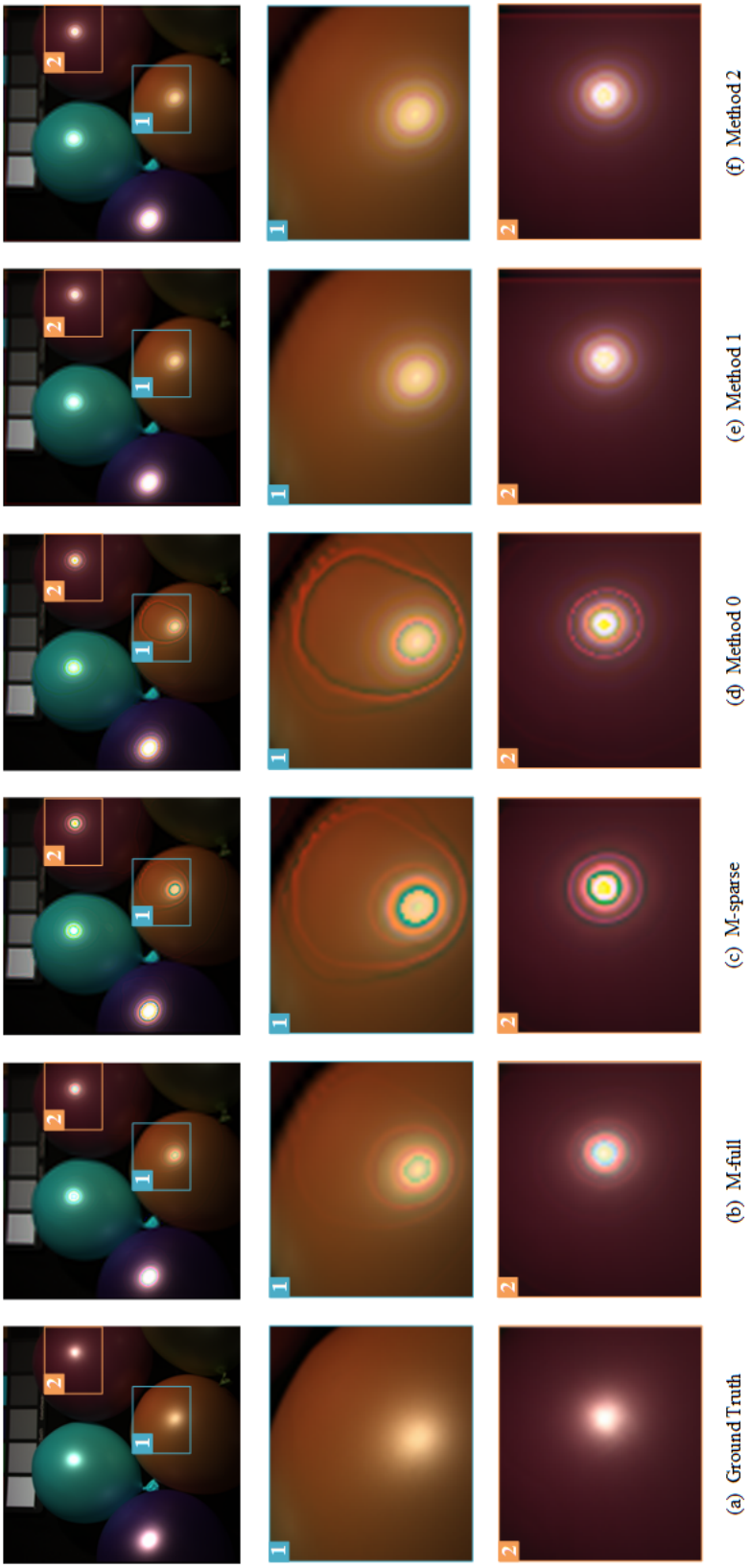


Figure 6.8: Super-resolution results using different ANFISs on HSI 'balloons' with scale factor being 3.

6.4 Summary

This chapter has presented a novel approach for hyperspectral image super resolution with spare training data, via the use of the proposed ANFIS interpolation technique. This study extends the prior work of Chapter 5 where multiple ANFISs were trained over full training data and mapped with the patches of an LR image to obtain the final HR image. According to the characteristics of different image bands, the training sets are divided into subsets with sufficient or insufficient data. Subsets with sufficient training data are referred to as source domains, where ANFIS models can be produced directly and used as source ANFISs for the interpolation in the target domains, where a large number of training data is missing. The performance of all the three interpolation methods in dealing with the HSI SR problem has been evaluated against five testing HSIs, showing that they provide similar results as those attainable by the reference model with full data in all subsets, while significantly improving upon the outcomes achieved by the ANFISs directly trained with sparse data without interpolation. The next chapter will conclude this thesis and discuss possible future work that may be developed to improve the work in the thesis.

Chapter 7

Conclusion and Future Work

THIS chapter presents a summary of this thesis as detailed in the preceding chapters. In particular, Section 7.1 summarizes the main contributions made from this research. Having reviewed the theoretical basis of a popular fuzzy inference system (ANFIS), this thesis has proposed a number of improved techniques over it, with both theoretical and practical promising results. The effectiveness of ANFIS with full training data is verified by applying multiple learned ANFIS mappings in the popular super resolution problem. The proposed approaches for interpolation with a rule group and GA based ANFIS interpolation significantly improve the performance of an ANFIS given sparse training data. The practical application of ANFIS interpolation to hyperspectral image super resolution validates its efficacy in addressing real world problems. Whilst the work is promising, Section 7.2 of this chapter points out some initial thoughts for further improving this research, on both the theoretical 'ANFIS interpolation' aspect and the practical 'image super resolution' aspect.

7.1 Summary of Thesis

The core work presented in this thesis is the concept and implementation of group based fuzzy rule interpolation, which is a novel extension of traditional single rule based FRI approaches. It is able to make a fuzzy inference system (specified in ANFIS within this thesis) to be learned over sparse training data. ANFIS interpolation as having been presented in this thesis follows the general idea of FRI, while possessing a number of specific characteristics:

1. Unlike traditional FRI approaches, which interpolate one single rule to perform fuzzy inference for an unfired observation, a group of fuzzy rules are collectively interpolated covering the entire target domain.

2. Instead of using Mamdani type of rules [108] for knowledge representation, which almost all existing FRI techniques (other than those reported in [31, 95, 153]) take, TSK type of rules [131] is utilised.

3. The proposed approach handles situations that are rather different from those dealt with by traditional FRI techniques, where training data (or more generally the rules) regarding the actual problem domain (namely, the target domain) are extremely sparse, but there are sufficient data (or rules) in the neighbouring domains (or the source domains). That is, the knowledge about the target domain is so poor that traditional FRI methods simply fail to work (with few rules available to carry out interpolation). Combining these characteristics makes it a very difficult task to interpolate a set of accurate rules in an effort to construct a required ANFIS. Of course, if there were a fair amount of training data available in the target domain, then a traditional FRI that works for TSK models would be sufficient to perform interpolation; that is, there would not be a need for the separation of source domains from the target domain in the first place.

Particularly, the main contributions of this thesis include a theoretical part on 'ANFIS interpolation' and an application part on 'image super resolution'.

7.1.1 ANFIS interpolation methods

Chapter 3 has presented a primary ANFIS interpolation method, i.e., to use an interpolation scheme based on a group of fuzzy rules to construct an ANFIS with insufficient training data. In this approach, the concept of rule dictionary has been proposed to restore the extracted rules, which acts in the same role as a sparse rule-base in traditional fuzzy rule interpolation. Then a group of new rules are interpolated by exploiting the LLE algorithm for the clustered sparse training data, forming an intermediate ANFIS. It is subsequently used as an initial network, with which the fine turning process can work more efficiently. Systematic experiments have been conducted, demonstrating that the performance of ANFIS with sparse data can

be significantly improved by this primary ANFIS interpolation method. Experiments using different percentages of training data have also been conducted, showing the effectiveness of the proposed interpolation method over extremely sparse data. Furthermore, experiments over an approximate linear function modelling problem have showed that the proposed approach may still work even no training data is available in the target domain.

Based on this primary method, Chapter 4 has presented an alternative ANFIS interpolation approach that implements the interpolation with a GA process. Two strategies of chromosome representation (rule based one and ANFIS based one) have been proposed, forming two different interpolation algorithms. The implementation of algorithm with the ANFIS based chromosomes is easier as there are only one global population involved. While for the algorithm with rule type chromosomes, the whole population is initialized as several sub populations, and the GA process are conducted within these sub populations. Also, a score table has been designed for the convenience of evaluating the rule type chromosomes. Systematic experimental examinations of the proposed approach have been carried out. Experimental results have shown that the proposed method helps boost the regression performance compared to both the original ANFIS and the primary interpolation method. This improved approach has also been compared with several classical machine learning methods, showing its competitiveness in dealing with regression problems.

7.1.2 Real world applications

ANFIS has been applied to perform the image super resolution task in Chapter 5, with sufficient training data provided. The proposed approach can be divided into a learning stage and a testing stage. Multiple nonlinear mappings from the low resolution space to the high resolution space are implemented using ANFIS through a training stage. While in the testing stage, a pre-processing technique is firstly carried out to enlarge input images of a small size, through interpolating LR image pixels. Each patch that is extracted from the pre-processed image is then enhanced by choosing and employing one of the learned ANFIS mappings. Post-processing techniques are finally performed to suppress the displeasing artefacts, resulting in the required HR image as output. The proposed approach has been compared with a number of popular image SR algorithms, and experimental results have revealed

the outstanding performance of the proposed approach over the rest, indicating the superiority of ANFIS in ISR problems when sufficient data is available.

Having shown the effectiveness of ANFIS in dealing with ISR using full training data in Chapter 5, Chapter 6 has investigated the performance of three proposed ANFIS interpolation methods in coping with the hyperspecral image super resolution problem using sparse data. In the training stage, the image cubes are clustered into several subsets according to the different characteristics of image bands. Some bands are with sufficient data while some are with sparse data. The subsets with sufficient training data are then used for learning ANFIS mappings using the same way as in Chapter 5, and for each of the remaining subsets with sparse data, the proposed ANFIS interpolation methods are applied, where two of the learned ANFISs with sufficient data are selected as the source ANFISs. Experimental results have demonstrated that all of the three interpolated models improve the super resolution results for the bands with sparse data.

7.2 Future Work

Although promising, much can be done to further improve the work presented so far in this thesis. The following addresses a number of interesting issues that may help strength the current research.

7.2.1 On ANFIS interpolation

In the proposed ANFIS interpolation methods, some preliminary techniques are employed due to their popularity and simplicity, which can be modified with more sophisticated and more powerful alternatives. As stated throughout this thesis, the simplest K-means algorithm is used for clustering; using an automated data clustering method such as those proposed in [16] would help derive more accurate interpolative results. Similarly, instead of using LLE, it is worth examining how other optimisation mechanisms may be adapted for integrated use within the interpolation process. Also, the simplest evolutionary process (genetic algorithm) is utilised, implementing the underlying ideas with more advanced evolutionary algorithms may lead to further strengthened interpolative results. Purely for convenience in implementation, in the

evolutionary ANFIS interpolation method, the sizes of different clusters are herein assumed to be the same, but there is no reason why an adaptive mechanism cannot be introduced to form different cluster sizes, reflecting the amounts of training data scales given per cluster. This may further improve the learning performance. Furthermore, most implementations are based on the use of triangular fuzzy sets, following the common practice in the FRI literature. Nonetheless, this is not necessary, other forms of member functions can be exploited. Whilst it is not expected to adversely affect the performance of the proposed approaches, verifications of this postulation requires future experimental investigation.

Currently, the interpolation methods either terminate once an interpolated ANFIS covering an originally sparse data area is obtained (for the initial ANFIS interpolation approach presented in Chapter 3), or iterates with a fixed number of rules (for the evolutionary ANFIS interpolation in Chapter 4). If the fuzzy rules can be updated dynamically and adaptively with respect to both the rule parameters and the rule numbers, it can be expected that more interpolated ANFIS models may be attained. Thus, it would be very useful to consider extending the existing ideas on dynamic generalization and promotion of interpolated rules as of [113] to create novel ANFIS models on the fly, gaining overall inference efficiency as well as effectiveness. Also, ANFIS is herein chosen as a representative fuzzy inference system to perform the proposed interpolation of FISs. Yet, apart from ANFIS, other inference systems may also suffer from the data shortage problem in the process of model development. Thus, extending the present results of devising and running interpolation based on a group of rules with other fuzzy inference systems would be another interesting piece of further research.

Finally, the proposed approach has been evaluated with several function modelling problems and a range of real world benchmark datasets, plus a practical application for hyperspectral image super resolution. It would be interesting to extend the experimental investigations with more real world tasks and more complicated problems. A particular focus on the practical application of this research may be for image modelling and analysis, especially in the areas of remote sensing [151, 169] and medical diagnosis [91, 104].

7.2.2 On image super resolution

The proposed image super resolution methods (either with full training data or sparse training data) currently work on raw image pixels. However, much research has shown that image features may be significantly more representative than raw image pixels. The gradient and wavelet features which are commonly used in image processing tasks are likely candidates for this. Therefore, extracting and using appropriate image features to train more efficient ANFIS mappings is of great interest to potentially improve the work. Furthermore, as widely recognised in the literature, the use of less but more informative features may do better than utilising all extracted features [68, 140], integrating feature selection techniques (e.g., [42, 69]) into such further developments may boost their efficiency.

Also, all parameters used in the ISR methods are empirically preset in this thesis. For example, the number of ANFIS mappings and the maximum number of fuzzy rules in an ANFIS are set as pre-fixed values by trading off the algorithm's accuracy and efficiency. However, the best parameters for different images may be various, affected by different image characteristics such as image size, image contents and image types. Thus, how to determine the appropriate parameters adaptively is an issue that needs further investigation, if this process is to be automated with minimal human intervention. Besides, the proposed HSI SR approach has been evaluated against the CAVE dataset only. Yet, there are various hyperspectral image datasets that may have different characteristics from those currently employed in this thesis. Therefore, modifying and testing the ANFIS interpolation aided HSI SR against other representative datasets also forms an important future work.

Appendix A

Publications Arising from the Thesis

A number of publications have been generated from this research carried out within the PhD project. The resultant publications that are in close relevance to the thesis are listed below, including both journal and conference papers.

A.1 Journal Articles

1. **Jing Yang**, Changjing Shang, Ying Li, Fangyi Li and Qiang Shen, Constructing ANFIS with Sparse Data through Group-Based Rule Interpolation: An Evolutionary Approach. IEEE Transactions on Fuzzy Systems, 2021. DOI: 10.1109/TFUZZ.2021.3049949

2. **Jing Yang**, Changjing Shang, Ying Li, Fangyi Li and Qiang Shen, ANFIS Construction with Sparse Data via Group Rule Interpolation. IEEE Transactions on Cybernetics, vol. 51, no. 5, pp. 2773-2786, 2019.

3. **Jing Yang**, Ying Li, Jonathan Cheung-Wai Chan and Qiang Shen, Image Fusion for Spatial Enhancement of Hyperspectral Image via Pixel Group Based Non-Local Sparse Representation. Remote Sensing, vol. 9, no. 1, 2017.

4. Tianhua Chen, Changjing Shang, **Jing Yang**, Fangyi Li and Qiang Shen, A New Approach for Transformation-based Fuzzy Rule Interpolation. IEEE Transactions on Fuzzy Systems, vol. 28, no. 12, pp. 3330-3344, 2020.

5. Fangyi Li, Changjing Shang, Ying Li, **Jing Yang** and Qiang Shen, Fuzzy Rule Based Interpolative Reasoning Supported by Attribute Ranking. IEEE Transactions on Fuzzy Systems, vol. 26, no. 5, pp. 2758-2773, 2018.

6. Fangyi Li, Changjing Shang, Ying Li, **Jing Yang** and Qiang Shen, Interpolation with Just Two Nearest Neighbouring Weighted Fuzzy Rules. IEEE Transactions on Fuzzy Systems, vol. 28, no. 9, pp. 2255-2262, 2020.

7. Fangyi Li, Changjing Shang, Ying Li, **Jing Yang** and Qiang Shen, Approximate Reasoning with Fuzzy Rule Interpolation: Background and Recent Advances. Artificial Intelligence Review, 2021. DOI: https://doi.org/10.1007/s10462-021-10005-3

A.2 Conference Papers

8. **Jing Yang**, Changjing Shang, Ying Li, Fangyi Li and Qiang Shen, Generating ANFISs Through Rule Interpolation: An Initial Investigation. Proceedings of the 2018 UK Workshop on Computational Intelligence, 2018.

9. **Jing Yang**, Changjing Shang, Ying Li and Qiang Shen, Single Frame Image Super Resolution via Learning Multiple ANFIS Mappings. Proceedings of the 26th International Conference on Fuzzy Systems, 2017.

10. Muhammad Ismail, **Jing Yang**, Changjing Shang and Qiang Shen, Single Frame Image Super Resolution using ANFIS Interpolation: An Initial Experiment-Based Approach. Proceedings of the 2019 UK Workshop on Computational Intelligence, 2019.

11. Muhammad Ismail, **Jing Yang**, Changjing Shang and Qiang Shen, Image Super Resolution with Sparse Data Using ANFIS Interpolation. Proceedings of the 29th International Conference on Fuzzy Systems, 2020.

Appendix B

List of Acronyms

5-FCV 5-Fold Cross-Validation

ANFIS Adaptive Network-based Fuzzy Inference System

CART Classification And Regression Tree

CNN Convolutional Neural Network

CRI Compositional Rule of Inference

COG Centre Of Gravity

EC Evolutionary Computation

EFS Evolutionary Fuzzy System

FIS Fuzzy Inference System

FRI Fuzzy Rule Interpolation

GA Genetic Algorithm

GP Genetic Programming

HR High Resolution

HSI Hyper-Spectral Image

IBP Iterative Back Projection

ISR Image Super Resolution

KEEL Knowledge Extraction based on Evolutionary Learning

KH Kóczy and Hirota

LinR Linear Regression

LLE Local Linear Embedding

LR Low Resolution

LSE Least Square Estimation

MF Membership Function

NLM Non-Local Means

PSNR Peak Signal-to-Noise Ratio

PSO Particle Swarm Optimisation

RD Rule Dictionary

RF Random Forest

RMSE Root-Mean-Squared Error

SD Standard Deviation

SSIM Structure SIMilarity

SVR Support Vector Regression

TSK Takagi-Sugeno-Kang

T-FRI Transformation-based Fuzzy Inference System

Bibliography

[1] A. Agapitos, R. Loughran *et al.*, "A survey of statistical machine learning elements in genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 1029–1048, 2019.

[2] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the 1993 ACM SIGMOD international conference on Management of Data*, 1993, pp. 207–216.

[3] T. Akgun, Y. Altunbasak, and R. M. Mersereau, "Super-resolution reconstruction of hyperspectral images," *IEEE Transactions on Image Processing*, vol. 14, no. 11, pp. 1860–1875, 2005.

[4] J. Alcalá-Fdez and J. M. Alonso, "A survey of fuzzy systems software: Taxonomy, current research trends, and prospects," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 1, pp. 40–56, 2016.

[5] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework." *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, 2011.

[6] K. K. Amanatiadis A, Andreadis I, "Design and implementation of a fuzzy area-based image-scaling technique," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 8, pp. 1504–1513, 2008.

[7] P. Angelov, D. P. Filev, and N. Kasabov, *Evolving intelligent systems: methodology and applications*. John Wiley & Sons, 2010, vol. 12.

[8] K. J. Åström and B. Wittenmark, *Computer-controlled systems: theory and design*. Courier Corporation, 2013.

[9] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Evolutionary computation 1: Basic algorithms and operators*. CRC press, 2018.

[10] T. Bäck, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, 1997.

[11] A. Bagheri, H. M. Peyhani, and M. Akbari, "Financial forecasting using anfis networks with quantum-behaved particle swarm optimization," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6235–6250, 2014.

[12] P. Baranyi, L. T. Kóczy, and T. D. Gedeon, "A generalized concept for fuzzy rule interpolation," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 6, pp. 820–837, 2004.

[13] M. Bevilacqua, A. Roumy, and Guillemot, "Single-image super-resolution via linear mapping of interpolated self examples," *IEEE transactions on Image Processing*, vol. 23, no. 12, pp. 5334–5347, 2014.

[14] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Single-image super-resolution via linear mapping of interpolated self-examples," *IEEE transactions on Image Processing*, vol. 23, no. 12, pp. 5334–5347, 2014.

[15] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 2, pp. 6–36, 2013.

[16] T. Boongoen, C. Shang, N. Iam-On, and Q. Shen, "Extending data reliability measure to a filter approach for soft subspace clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 6, pp. 1705–1714, 2011.

[17] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2005, pp. 60–65.

[18] J. Buckley, W. Siler, and D. Tucker, "A fuzzy expert system," *Fuzzy Sets and Systems*, vol. 20, no. 1, pp. 1–16, 1986.

[19] Y. Chai, L. Jia, and Z. Zhang, "Mamdani model based adaptive neural fuzzy inference system and its application," *International Journal of Computational Intelligence*, vol. 5, no. 1, pp. 22–29, 2009.

[20] A. Chakrabarti and T. Zickler, "Statistics of real-world hyperspectral images," in *CVPR 2011*. IEEE, 2011, pp. 193–200.

[21] J. C.-W. Chan, J. Ma, P. Kempeneers, and F. Canters, "Superresolution enhancement of hyperspectral chris/proba images with a thin-plate spline nonrigid transform model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 6, pp. 2569–2579, 2010.

[22] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2004, pp. I–I.

[23] P.-C. Chang and C.-H. Liu, "A tsk type fuzzy rule based system for stock price prediction," *Expert Systems with applications*, vol. 34, no. 1, pp. 135–144, 2008.

[24] C. Chen, N. Mac Parthaláin, Y. Li, C. Price, C. Quek, and Q. Shen, "Rough-fuzzy rule interpolation," *Information Sciences*, vol. 351, pp. 1–17, 2016.

[25] H. Chen, X. He, Q. Teng, and C. Ren, "Single image super resolution using local smoothness and nonlocal self-similarity priors," *Signal Processing: Image Communication*, vol. 43, pp. 68–81, 2016.

[26] S. Chen and Y. Ko, "Fuzzy interpolative reasoning for sparse fuzzy rule-based systems based on α-cuts and transformations techniques," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1626–1648, 2008.

[27] S. Chen, Y. Ko, Y. Chang, and J. Pan, "Weighted fuzzy interpolative reasoning based on weighted increment transformation and weighted ratio transformation techniques," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 6, pp. 1412–1427, 2009.

[28] S.-M. Chen and Y.-C. Chang, "Weighted fuzzy rule interpolation based on ga-based weight-learning techniques," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 4, pp. 729–744, 2011.

[29] S.-M. Chen, Y.-C. Chang, and J.-S. Pan, "Fuzzy rules interpolation for sparse fuzzy rule-based systems based on interval type-2 gaussian fuzzy sets and genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 412–425, 2012.

[30] T. Chen, C. Shang, P. Su, and Q. Shen, "Induction of accurate and interpretable fuzzy rules from preliminary crisp representation," *Knowledge-Based Systems*, vol. 146, pp. 152–166, 2018.

[31] T. Chen, C. Shang, J. Yang, F. L. Li, and Q. Shen, "A new approach for transformation-based fuzzy rule interpolation," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 12, pp. 3330–3344, 2020.

[32] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Sparse representation for target detection in hyperspectral imagery," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 629–640, 2011.

[33] J.-H. Chiang and P.-Y. Hao, "Support vector learning mechanism for fuzzy rule-based modeling: a new approach," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 1, pp. 1–12, 2004.

[34] K. B. Cho and B. H. Wang, "Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction," *IEEE Transactions on Fuzzy Systems*, vol. 83, no. 3, pp. 325–339, 1996.

[35] T. S. Cho, C. L. Zitnick, N. Joshi, S. B. Kang, R. Szeliski, and W. T. Freeman, "Image restoration by matching gradient distributions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 683–694, 2011.

[36] C.-C. Chuang, S.-F. Su, and S.-S. Chen, "Robust tsk fuzzy modeling for function approximation with outliers," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 6, pp. 810–821, 2001.

[37] O. Cord *et al.*, *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*. World Scientific, 2001, vol. 19.

[38] O. Cordón, "A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems," *International Journal Approximate Reasoning*, vol. 52, no. 6, pp. 894–913, 2011.

[39] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: current framework and new trends," *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 5–31, 2004.

[40] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[41] 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*, vol. 21, no. 4, pp. 597–609, 2013.

[42] R. Diao and Q. Shen, "Two new approaches to feature selection with harmony search," in *International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2010, pp. 1–7.

[43] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[44] L. Duckstein *et al.*, *Fuzzy rule-based modeling with applications to geophysical, biological, and engineering systems*. CRC press, 1995, vol. 8.

[45] M. T. Eismann, J. Meola, and R. C. Hardie, "Hyperspectral change detection in the presenceof diurnal and seasonal variations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 1, pp. 237–249, 2007.

[46] M. T. Eismann, A. D. Stocker, and N. M. Nasrabadi, "Automated hyperspectral cueing for civilian search and rescue," *Proceedings of the IEEE*, vol. 97, no. 6, pp. 1031–1055, 2009.

[47] B. B. Ekici and U. T. Aksoy, "Prediction of building energy needs in early stage of design by using anfis," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5352–5358, 2011.

[48] A. Esfahanipour and W. Aghamiri, "Adapted neuro-fuzzy inference system on indirect approach tsk fuzzy rule base for stock market analysis," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4742–4748, 2010.

[49] R. Feng, Y. Zhong, and L. Zhang, "Adaptive non-local euclidean medians sparse unmixing for hyperspectral imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 97, pp. 9–24, 2014.

[50] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *12th IEEE International Conference on Computer Vision*. IEEE, 2009, pp. 349–356.

[51] X. Gu, P. Angelov, and Q. Shen, "Particle swarm optimized autonomous learning fuzzy system," *IEEE Transactions on Cybernetics*, 2020.

[52] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 1664–1673.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[54] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks for Perception*. Elsevier, 1992, pp. 65–93.

[55] F. Herrera, M. Lozano, and J. L. Verdegay, "Tuning fuzzy logic controllers by genetic algorithms," *International Journal Approximate Reasoning*, vol. 12, no. 3-4, pp. 299–315, 1995.

[56] H. Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 6, pp. 508–517, 1978.

[57] W. Huang, L. Xiao, H. Liu, and Z. Wei, "Hyperspectral imagery super-resolution by compressive sensing inspired dictionary learning and spatial-spectral regularization," *Sensors*, vol. 15, no. 1, pp. 2041–2058, 2015.

[58] Z. Huang and Q. Shen, "Fuzzy interpolative reasoning via scale and move transformations," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 2, pp. 340–359, 2006.

[59] Z. Huang, Q. Shen *et al.*, "Fuzzy interpolation and extrapolation: A practical approach," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 1, pp. 13–28, 2008.

[60] P. Hungerlander and F. Rendl, "A feasible active set method for strictly convex quadratic problems with simple bounds," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1633–1659, 2015.

[61] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 6, pp. 2014–2039, 2011.

[62] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 3, pp. 231–239, 1991.

[63] H. Ishibuchi and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 506–515, 2001.

[64] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 260–270, 1995.

[65] H. Ishibuchi and T. Yamamoto, "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining," *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 59–88, 2004.

[66] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[67] J.-S. R. Jang *et al.*, "Fuzzy modeling using generalized neural networks and kalman filter algorithm." in *AAAI*, vol. 91, 1991, pp. 762–767.

[68] R. Jensen and Q. Shen, "Are more features better? a response to attributes reduction using fuzzy rough sets," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 6, pp. 1456–1458, 2009.

[69] R. Jensen, A. Tuson, and Q. Shen, "Finding rough and fuzzy-rough set reducts with sat," *Information Sciences*, vol. 255, pp. 100–120, 2014.

[70] R. Ji, H. Liu, L. Cao, D. Liu, Y. Wu, and F. Huang, "Toward optimal manifold hashing via discrete locally linear embedding," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5411–5420, 2017.

[71] Y. Jiang, F.-L. Chung *et al.*, "Multitask tsk fuzzy system modeling by mining intertask common hidden structure," *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 534–547, 2014.

[72] Y. Jiang, Z. Deng, F.-L. Chung, G. Wang, P. Qian, K.-S. Choi, and S. Wang, "Recognition of epileptic eeg signals using a novel multiview tsk fuzzy system," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 1, pp. 3–20, 2017.

[73] S. Jin, R. Diao, C. Quek, and Q. Shen, "Backward fuzzy rule interpolation," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 6, pp. 1682–1698, 2014.

[74] S. Jin and J. Peng, "Towards hierarchical fuzzy rule interpolation," in *2015 IEEE International Conference on Cognitive Informatics & Cognitive Computing (ICCI & CC)*. IEEE, 2015, pp. 267–274.

[75] Z. C. Johanyák and S. Kovács, "A brief survey and comparison on various interpolation based fuzzy reasoning methods," *Acta Polytechnica Hungarica*, vol. 3, no. 1, pp. 91–105, 2006.

[76] ——, "Survey on various interpolation based fuzzy reasoning methods," *Production Systems and Information Engineering*, vol. 3, no. 1, pp. 39–56, 2006.

[77] A. Kabán and S. A. Pitchay, "Single-frame image recovery using a pearson type vii mrf," *Neurocomputing*, vol. 80, pp. 111–118, 2012.

[78] N. K. Kasabov and Q. Song, "Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, 2002.

[79] R. Kawakami, Y. Matsushita, J. Wright, M. Ben-Ezra, Y.-W. Tai, and K. Ikeuchi, "High-resolution hyperspectral imaging via matrix factorization," in *CVPR 2011*. IEEE, 2011, pp. 2329–2336.

[80] H. B. Kazemian and S. A. Yusuf, "An anfis approach to transmembrane protein prediction," in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2014, pp. 1360–1365.

[81] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44–57, 2002.

[82] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[83] L. T. Koczy and K. Hirota, "Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases," *Information Sciences*, vol. 71, no. 1, pp. 169–201, 1993.

[84] L. Kóczy and K. Hirota, "Approximate reasoning by linear rule interpolation and general approximation," *International Journal of Approximate Reasoning*, vol. 9, no. 3, pp. 197–225, 1993.

[85] S. Kovács, "Fuzzy rule interpolation," in *Encyclopedia of Artificial Intelligence*. IGI Global, 2009, pp. 728–733.

[86] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[87] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 2, pp. 262–267, 2010.

[88] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 624–632.

[89] T. M. Lehmann, C. Gonner, and K. Spitzer, "Survey: Interpolation methods in medical image processing," *IEEE Transactions on Medical Imaging*, vol. 18, no. 11, pp. 1049–1075, 1999.

[90] F. Li, Y. Li, C. Shang, and Q. Shen, "Fuzzy knowledge-based prediction through weighted rule interpolation," *IEEE transactions on cybernetics*, vol. 50, no. 10, pp. 4508–4517, 2019.

[91] F. Li, C. Shang, Y. Li, and Q. Shen, "Interpretable mammographic mass classification with fuzzy interpolative reasoning," *Knowledge-Based Systems*, vol. 191, p. 105279, 2020.

[92] F. Li, C. Shang, Y. Li, J. Yang, and Q. Shen, "Fuzzy rule based interpolative reasoning supported by attribute ranking," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 2758–2773, 2018.

[93] ——, "Interpolation with just two nearest neighbouring weighted fuzzy rules," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 9, pp. 2255–2262, 2020.

[94] ——, "Approximate reasoning with fuzzy rule interpolation: background and recent advances," *Artificial Intelligence Review*, pp. 1–48, 2021.

[95] J. Li, L. Yang, Y. Qu, and G. Sexton, "An extended takagi–sugeno–kang inference system (tsk+) with fuzzy interpolation and its rule base generation," *Soft Computing*, vol. 22, no. 10, pp. 3155–3170, 2018.

[96] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, 2001.

[97] X. Li, Y. Hu, X. Gao, D. Tao, and B. Ning, "A multi-frame image super-resolution method," *Signal Processing*, vol. 90, no. 2, pp. 405–414, 2010.

[98] S. Lijun, X. ZhiYun, and H. Hua, "Image super-resolution based on mca and wavelet-domain hmt," in *2010 International Forum on Information Technology and Applications (IFITA)*, vol. 2. IEEE, 2010, pp. 264–269.

[99] X. Liu, L. Chen, W. Wang, and J. Zhao, "Robust multi-frame super-resolution based on spatially weighted half-quadratic estimation and adaptive btv regularization," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4971–4986, 2018.

[100] L. Loncan, L. B. De Almeida, J. M. Bioucas-Dias, X. Briottet, J. Chanussot, N. Dobigeon, S. Fabre, W. Liao, G. A. Licciardi, M. Simoes *et al.*, "Hyperspectral pansharpening: A review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 3, no. 3, pp. 27–46, 2015.

[101] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[102] G. Lu and B. Fei, "Medical hyperspectral imaging: a review," *Journal of biomedical optics*, vol. 19, no. 1, p. 010901, 2014.

[103] X.-J. Ma, Z.-Q. Sun, and Y.-Y. He, "Analysis and design of fuzzy controller and fuzzy observer," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 1, pp. 41–51, 1998.

[104] N. MacParthalain, R. Jensen, Q. Shen, and R. Zwiggelaar, "Fuzzy-rough approaches for mammographic risk analysis," *Intelligent Data Analysis*, vol. 14, no. 2, pp. 225–244, 2010.

[105] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[106] L. Magdalena and F. Monasterio-Huelin, "A fuzzy logic controller with learning through the evolution of its knowledge base," *International Journal of Approximate Reasoning*, vol. 16, no. 3-4, pp. 335–358, 1997.

[107] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[108] E. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Transactions on Computers*, vol. 26, no. 12, pp. 1182–1191, 1977.

[109] K. C. Mertens, B. De Baets, L. P. Verbeke, and R. R. De Wulf, "A sub-pixel mapping algorithm based on sub-pixel/pixel spatial attraction models," *International Journal of Remote Sensing*, vol. 27, no. 15, pp. 3293–3310, 2006.

[110] F. A. Mianji, Y. Gu, Y. Zhang, and J. Zhang, "Enhanced self-training superresolution mapping technique for hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 671–675, 2011.

[111] S. Mousavi, S. Mirinezhad, and V. Lyashenko, "An evolutionary-based adaptive neuro-fuzzy expert system as a family counselor before marriage with the aim of divorce rate reduction," *Artigence*, vol. 1, pp. 1–16, 05 2018.

[112] N. Naik, R. Diao, C. Quek, and Q. Shen, "Towards dynamic fuzzy rule interpolation," in *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2013, pp. 1–7.

[113] N. Naik, R. Diao, and Q. Shen, "Dynamic fuzzy rule interpolation and its application to intrusion detection," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 1878–1892, 2018.

[114] O. Obajemu and M. Mahfouf, "A dirichlet process based type-1 and type-2 fuzzy modeling for systematic confidence bands prediction," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 9, pp. 1853–1865, 2019.

[115] O. Obajemu, M. Mahfouf, and J. W. Catto, "A new fuzzy modeling framework for integrated risk prognosis and therapy of bladder cancer patients," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 3, pp. 1565–1577, 2017.

[116] S. J. Pan, Q. Yang *et al.*, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[117] M. Papananias, T. E. McLeay, M. Mahfouf, and V. Kadirkamanathan, "An intelligent metrology informatics system based on neural networks for multistage manufacturing processes," *Procedia CIRP*, vol. 82, pp. 444–449, 2019.

[118] K. M. G. Park S C, Park M K, "Super-resolution image reconstruction: a technical overview," *Signal Processing*, vol. 20, no. 3, pp. 21–36, 2003.

[119] R. C. Patel and M. V. Joshi, "Super-resolution of hyperspectral images: Use of optimum wavelet filter coefficients and sparsity regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 1728–1736, 2014.

[120] I. Perfilieva, D. Dubois, H. Prade, F. Esteva, L. Godo, and P. Hoďáková, "Interpolation of fuzzy data: Analytical approach and overview," *Fuzzy Sets and Systems*, vol. 192, pp. 134–158, 2012.

[121] J. Petke, S. O. Haraldsson, M. Harman, W. B. Langdon, D. R. White, and J. R. Woodward, "Genetic improvement of software: a comprehensive survey," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 415–432, 2017.

[122] P. Purkait, N. R. Pal, and B. Chanda, "A fuzzy-rule-based approach for single frame super resolution," *IEEE transactions on Image Processing*, vol. 23, no. 5, pp. 2277–2290, 2014.

[123] S.-E. Qian and G. Chen, "Enhancing spatial resolution of hyperspectral imagery using sensor's intrinsic keystone distortion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 12, pp. 5033–5048, 2012.

[124] J. Qiu, H. Gao, and S. X. Ding, "Recent advances on fuzzy-model-based nonlinear networked control systems: A survey," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1207–1217, 2016.

[125] J. A. Robinson, "A machine-oriented logic based on the resolution principle," *Journal of the ACM (JACM)*, vol. 12, no. 1, pp. 23–41, 1965.

[126] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[127] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[128] J. Shell and S. Coupland, "Fuzzy transfer learning: methodology and application," *Information Sciences*, vol. 293, pp. 59–79, 2015.

[129] Y. Shi, R. Eberhart, and Y. Chen, "Implementation of evolutionary fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 2, pp. 109–119, 1999.

[130] T. Takagi and M. Sugeno, "Derivation of fuzzy control rules from human operator's control actions," *IFAC Proceedings Volumes*, vol. 16, no. 13, pp. 55–60, 1983.

[131] ——, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, no. 1, pp. 116–132, 1985.

[132] D. Tikk and P. Baranyi, "Comprehensive analysis of a new fuzzy rule interpolation method," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 3, pp. 281–296, 2000.

[133] D. Tikk, I. Joó, L. T. Kóczy, P. Várlaki, B. Moser, and T. D. Gedeon, "Stability of interpolative fuzzy kh controllers," *Fuzzy Sets and Systems*, vol. 125, no. 1, pp. 105–119, 2002.

[134] D.-H. Trinh, M. Luong, F. Dibos, J.-M. Rocchisani, C.-D. Pham, and T. Q. Nguyen, "Novel example-based method for super-resolution and denoising of medical images," *IEEE Transactions on Image processing*, vol. 23, no. 4, pp. 1882–1895, 2014.

[135] I. Turkmen, "Efficient impulse noise detection method with anfis for accurate image restoration," *AEU-International Journal of Electronics and Communications*, vol. 65, no. 2, pp. 132–139, 2011.

[136] I. Turksen and Z. Zhong, "An approximate analogical reasoning approach based on similarity measures," *IEEE transactions on systems, man, and cybernetics*, vol. 18, no. 6, pp. 1049–1056, 1988.

[137] G. Vass, L. Kalmár, and L. Kóczy, "Extension of the fuzzy rule interpolation method," in *Proceedings of the International Conference on Fuzzy Sets Theory Applications*, 1992, pp. 1–6.

[138] Z. J. Viharos and K. B. Kis, "Survey on neuro-fuzzy systems and their applications in technical diagnostics and measurement," *Measurement*, vol. 67, pp. 126–136, 2015.

[139] A. Wafa'H, B. Khorsheed, M. Mahfouf, G. K. Reynolds, and A. D. Salman, "An interpretable fuzzy logic based data-driven model for the twin screw granulation process," *Powder Technology*, vol. 364, pp. 135–144, 2020.

[140] X. Wang, A. E. Brownlee, J. R. Woodward, M. Weiszer, M. Mahfouf, and J. Chen, "Aircraft taxi time prediction: Feature importance and their implications," *Transportation Research Part C: Emerging Technologies*, vol. 124, p. 102892, 2021.

[141] Y. Wang, L. Wang, J. Yang, W. An, and Y. Guo, "Flickr1024: A large-scale dataset for stereo image super-resolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[142] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[143] H. Wei and X.-S. Tang, "A genetic-algorithm-based explicit description of object contour and its ability to facilitate recognition," *IEEE Transactions on Cybernetics*, vol. 45, no. 11, pp. 2558–2571, 2014.

[144] L.-Y. Wei, "A hybrid anfis model based on empirical mode decomposition for stock time series forecasting," *Applied Soft Computing*, vol. 42, pp. 368–376, 2016.

[145] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994.

[146] K. W. Wong, D. Tikk, T. D. Gedeon, and L. T. Kóczy, "Fuzzy rule interpolation for multidimensional input spaces with applications: A case study," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 6, pp. 809–819, 2005.

[147] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "Aid: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, 2017.

[148] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2015.

[149] C.-Y. Yang and M.-H. Yang, "Fast direct super-resolution by simple functions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 561–568.

[150] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[151] J. Yang, Y. Li, J. Chan, and Q. Shen, "Image fusion for spatial enhancement of hyperspectral image via pixel group based non-local sparse representation," *Remote Sensing*, vol. 9, no. 1, p. 53, 2017.

[152] J. Yang, C. Shang, Y. Li, F. Li, and Q. Shen, "Generating anfiss through rule interpolation: An initial investigation," in *Advances in Intelligent Systems and Computing*, 2018.

[153] ——, "Anfis construction with sparse data via group rule interpolation," *IEEE Transactions on Cybernetics*, 2019.

[154] J. Yang, C. Shang, Y. Li, and Q. Shen, "Single frame image super resolution via learning multiple anfis mappings," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–6.

[155] L. Yang, F. Chao, and Q. Shen, "Generalised adaptive fuzzy rule interpolation," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 839–853, 2017.

[156] L. Yang and Q. Shen, "Adaptive fuzzy interpolation," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 6, pp. 1107–1126, 2011.

[157] ——, "Closed form fuzzy interpolation," *Fuzzy Sets and Systems*, vol. 225, pp. 1–22, 2013.

[158] S. Yang, M. Wang, Y. Chen, and Y. Sun, "Single-image super-resolution reconstruction via learned geometric dictionaries and clustered sparse coding," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4016–4028, 2012.

[159] D. S. Yeung and E. C. C. Tsang, "A comparative study on similarity-based fuzzy reasoning methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 27, no. 2, pp. 216–227, 1997.

[160] N. Yokoya, T. Yairi, and A. Iwasaki, "Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 2, pp. 528–537, 2011.

[161] Y. Yuan and W. Banzhaf, "Arja: Automated repair of java programs via multi-objective genetic programming," *IEEE Transactions on Software Engineering*, vol. 46, no. 10, pp. 1040–1067, 2018.

[162] Y. Yuan, X. Zheng, and X. Lu, "Hyperspectral image superresolution by transfer learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 5, pp. 1963–1974, 2017.

[163] Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets and Systems*, vol. 69, no. 2, pp. 125–139, 1995.

[164] L. Yue, H. Shen, J. Li, Q. Yuan, H. Zhang, and L. Zhang, "Image super-resolution: The techniques, applications, and future," *Signal Processing*, vol. 128, pp. 389–408, 2016.

[165] L. A. Zadeh, "Quantitative fuzzy semantics," *Information Sciences*, vol. 3, no. 2, pp. 159–176, 1971.

[166] ——, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 1, pp. 28–44, 1973.

[167] ——, "Fuzzy logic," *Computer*, vol. 21, no. 4, pp. 83–93, 1988.

[168] ——, "Fuzzy sets," in *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*. World Scientific, 1996, pp. 394–432.

[169] H. Zhang, Y. Li, Y. Jiang, P. Wang, Q. Shen, and C. Shen, "Hyperspectral classification based on lightweight 3d-cnn with transfer learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5813–5828, 2019.

[170] H. Zhang, L. Zhang, and H. Shen, "A super-resolution reconstruction algorithm for hyperspectral images," *Signal Processing*, vol. 92, no. 9, pp. 2082–2096, 2012.

[171] K. Zhang, D. Tao, X. Gao, X. Li, and Z. Xiong, "Learning multiple linear mappings for efficient single image super-resolution," *IEEE transactions on Image Processing*, vol. 24, no. 3, pp. 846–861, 2015.

[172] P. Zhang and Q. Shen, "A novel framework of fuzzy rule interpolation for takagi-sugeno-kang inference systems," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2019, pp. 1–6.

[173] ——, "Dynamic tsk systems supported by fuzzy rule interpolation: An initial investigation," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2020, pp. 1–7.

[174] Y. Zhang, S. De Backer, and P. Scheunders, "Noise-resistant wavelet-based bayesian fusion of multispectral and hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 11, pp. 3834–3843, 2009.

[175] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[176] Y. Zhao, J. Yang, and J. C.-W. Chan, "Hyperspectral imagery super-resolution by spatial–spectral joint nonlocal similarity," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2671–2679, 2013.

[177] H.-J. Zimmermann, "Fuzzy set theory," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 3, pp. 317–332, 2010.

[178] 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*, vol. 26, no. 2, pp. 847–858, 2018.

[179] ——, "Fuzzy regression transfer learning in takagi–sugeno fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1795–1807, 2017.