ANFIS Construction With Sparse Data via Group Rule Interpolation

Yang, Jing; Shang, Changjing; Li, Ying; Li, Fangyi; Shen, Qiang

Published in:
IEEE Transactions on Cybernetics

DOI:
10.1109/TCYB.2019.2952267

Publication date:
2019

Citation for published version (APA):
Yang, J., Shang, C., Li, Y., Li, F., & Shen, Q. (2019). ANFIS Construction With Sparse Data via Group Rule Interpolation. *IEEE Transactions on Cybernetics*. https://doi.org/10.1109/TCYB.2019.2952267

ANFIS Construction with Sparse Data via Group Rule Interpolation

Jing Yang, Changjing Shang, Ying Li, Fangyi Li and Qiang Shen

***Abstract*—A major assumption for constructing an effective ANFIS (Adaptive-Network-based Fuzzy Inference System) is that sufficient training data is available. However, in many real world applications, this assumption may not hold, thereby requiring alternative approaches. In light of this observation, this research focusses on automated construction of ANFISs in an effort to enhance the potential of Takagi-Sugeno fuzzy regression models, for situations where only limited training data is available. In particular, the proposed approach works by interpolating a group of fuzzy rules in a certain given 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 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. Experimental evaluation on both synthetic and real world datasets is reported, demonstrating that when facing the data shortage situations, the proposed approach helps significantly improve the performance of the original ANFIS modelling mechanism.**

***Index Terms*—ANFIS construction, data shortage, group rule interpolation, transfer learning, rule dictionary, locally linear embedding.**

I. INTRODUCTION

OWING to the simplicity and explainability in representing human knowledge, fuzzy rule-based systems have become one of the most popular tools for finding solutions to various real world problems (e.g., [1], [2], [3]). There are typically two forms of knowledge representation in fuzzy rule-based inference systems: Mandani type models [4] and Takagi-Sugeno (TSK) type models [5]. Mandani models are classical fuzzy systems which have been popular in many real world applications [6], while TSK ones have also played an increasingly important role in such applications, including for example, stock market prediction [7] and EGG signals recognition [8]. The success of TSK type fuzzy inference systems is largely owing to their capability of approximating complex nonlinear functions [9].

ANFIS (Adaptive-Network-based Fuzzy Inference System) with TSK type fuzzy rules is one of the most successful in implementing TSK type fuzzy models, performing fuzzy reasoning with an adaptive network-based architecture. It works by equivalently extracting useful knowledge in terms of a set of fuzzy rules directly from training data, and has proven to be simple but powerful for highly nonlinear problems. In learning such an ANFIS model, a main assumption is that there is sufficient data for training. However, in dealing with certain real situations, it is difficult or even impossible to obtain sufficient data to perform the required training procedure. The shortage of training data significantly restricts the potential of such systems.

In the relevant literature, a typical approach to addressing this practical issue (of training data shortage) is to conduct the learning process through a so-called transfer learning procedure [10], [11], [12], [13], [14]. Such techniques exploit the knowledge accumulated from data in an auxiliary domain (termed source domain, SD) to support predictive modelling in the problem domain at hand (termed target domain, TD). 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 certain SD and a TD via the 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) [15], [16], [17]. In general, FRI techniques work by interpolating new rules from the existing ones, of a sparse rule base that do not match a given observation. 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) [18], [19] has been proposed for improved performance through scale and move transformation. This greatly increases the accuracy of the interpolated rule and also, has led to a number of advanced theoretical and applicational developments over the past decade (e.g., [20], [21], [22], [23], [24], [25], [26], [27]). However, the existing literature of FRI is mainly focussed on Mandani type models, apart from limited initial attempt as reported in [28], [29], the research on FRI with TSK type models is still rather rare.

As an original contribution towards developing a novel approach to use FRI techniques for building TSK type models, this study focusses on the introduction of a transfer-

J. Yang and F. Li are with School of Computer Science, National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, Shaanxi Provincial Key Laboratory of Speech & Image Information Processing, Northwestern Polytechnical University, Xi’an, China and Department of Computer Science, Faculty of Business and Physical Sciences, Aberystwyth University, UK. (e-mail: {jiy6, fal2}@aber.ac.uk).

C. Shang and Q. Shen are with Department of Computer Science, Faculty of Business and Physical Sciences, Aberystwyth University, UK. (e-mail: {cns, qqs}@aber.ac.uk).

Y. Li is with School of Computer Science, National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, Shaanxi Provincial Key Laboratory of Speech & Image Information Processing, Northwestern Polytechnical University, Xi’an, China. (e-mail: lybyp@nwpu.edu.cn).

learning based approach to FRI for ANFIS. 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 TD, the proposed approach seeks the assistance of two neighbouring ANFISs that have been generated from two source domains (where sufficient training data is available). It is different from existing FRI techniques for TSK-type models which are concerned with individual rule interpolation within one fixed problem area, and which are designed to interpolate a rule for an unmatched observation, with just one TSK rule interpolated, using its closest neighbouring rules. This work focusses on the interpolation of an entire fuzzy inference system, where a group of fuzzy rules are interpolated simultaneously for a certain area, using ANFISs learned in the neighbourhood areas. Therefore, this group based work shows a radical departure from the usual focus of FRI, offering a new FRI mechanism with improved abilities in nonlinear adaptation and rapid learning.

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, the present work is established by exploiting the relationships between a TD and its neighbouring SDs, by the use of the locally linear embedding (LLE) algorithm [30]. This is due to the recognition that LLE is a promising manifold learning algorithm, capable of capturing hidden relationships between TD and SDs in a certain rule antecedent space. From this, by adopting similarity-based inference, hidden relationships within the rule consequent space can be subsequently learned. To aid in this learning process, a rule dictionary is generated firstly by extracting fuzzy rules from source ANFISs that are known. Next, a set of new rules are interpolated using LLE to construct an intermediate ANFIS. Finally, the resulting intermediate ANFIS is utilised as an initial network for fine tuning to obtain a more accurate one. Whilst ANFIS is chosen in this work to demonstrate the ideas (owing to its popularity, representativeness and availability), the insights gained from devising and running the ANFIS-based implementation can be extended to other TSK-type fuzzy inference systems.

The rest of this paper is structured as follows. Section II reviews the relevant background on ANFIS and LLE. In Section III, the proposed approach to ANFIS construction is described in detail. Experimental results are presented and analysed in Section IV. Finally, Section V concludes the paper and points out directions for further research.

II. Background

This section presents an overview of the most relevant work that will be used to develop the present research, including ANFIS and LLE.

A. ANFIS – Adaptive Network-based Fuzzy Inference System

ANFIS [31] is a fuzzy inference system that implements approximate reasoning within the general framework of an adaptive network. Thanks to its simplicity and effectiveness, it has been widely applied to various kinds of real world problems (e.g., [32], [33], [34], [35]). The following gives a brief 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.

1) Network Architecture: A general ANFIS network contains five layers of computing elements. For easy understanding, a simple ANFIS with two-input (x_1 and x_2) and one-output (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 and Sugeno's type [36] 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 $y_1 = p_1x_1 + q_1x_2 + r_1$
Rule 2: If x_1 is A_2 and x_2 is B_2,
then $y_2 = p_2x_1 + q_2x_2 + r_2$

The network structure of such a TSK type ANFIS can be shown in Fig. 1, where square nodes stand for adaptive computing units with modifiable parameters, and circle nodes represent those fixed units without parameters. Further details of the individual layers within this ANFIS are outlined below.

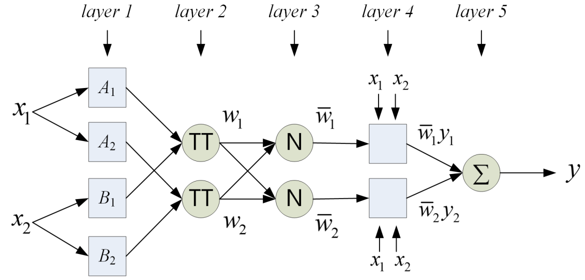


Fig. 1. ANFIS structure representing two TSK rules

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 simplicity, the popularly applied triangular membership functions are adopted here. In general terms, a triangular fuzzy set is defined by

$$\mu_O(x) = \begin{cases} k_1x + b_1 & a_0 \leq x \leq a_1 \\ k_2x + b_2 & a_1 \leq x \leq a_2 \\ 0 & otherwise \end{cases} \tag{1}$$

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 (a_0, a_1, a_2) can be simply defined as $Rep(O) = (a_0 + a_1 + a_2)/3$, 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.

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, acting as the firing strength of the ith rule ($i = 1, 2$):

$$w_i = \mu_{A_i}(x_1) \times \mu_{B_i}(x_2) \tag{2}$$

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:

$$\bar{w_i} = \frac{w_i}{w_1 + w_2} \tag{3}$$

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:

$$\bar{w_i}(p_i x_1 + q_i x_2 + r_i) = \bar{w_i} y_i \tag{4}$$

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,

$$y = \sum_i \bar{w_i} y_i = \bar{w_1} y_1 + \bar{w_2} y_2 \tag{5}$$

2) Parameter Training: The key point for constructing an effective ANFIS is to learn the unknown parameters in the network, including both the premise parameters describing the fuzzy sets and the consequent ones specifying the linear functions. In the original work reported in [31], these parameters are trained using a hybrid learning method combining gradient descent and Least Square Estimation (LSE). More specifically, the hybrid learning algorithm can be divided into forward pass and backward pass. In running the forward pass, the premise parameters are set to be fixed, the input values are fed forward till *Layer 4*, with the errors over the consequent parameters being identified by LSE. In the backward pass, the error rates are propagated backward with the consequent parameters being fixed, and the premise parameters are updated using the gradient descent method. More detailed description of this training process can be found in [31].

B. Locally Linear Embedding

The locally linear embedding algorithm (LLE) [30] is one of the most classical manifold learning methods. Although originally developed for tackling dimensionality reduction problems, it has now been widely utilised in performing different types of machine learning task (e.g., [37], [38]).

LLE generates a neighbourhood preserving mapping between a high-dimensional data space (denoted as D) and a low-dimensional data space (represented by d), 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 high 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:

$$\varepsilon(W) = |X_i - \sum_j w_j X_j|^2 \tag{6}$$

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 D, LLE involves the following implementation steps as summarised in Alg. II-B, in an effort to construct the corresponding data point Y_i in d (thereby reducing the data dimensionality). This summary is given for completeness, but further details regarding this algorithm can be found in [30].

```
Algorithm II-B: Locally Linear Embedding (LLE)
Input:
    $X_i$: Data point in $D$-dimensional data space
    $K$ : Number of closest neighbours
Step 1: Find $K$ nearest neighbours $\{X_j\}$ to $X_i$ in $D$
Step 2: Compute weights $\{w_j\}$ of selected neighbours
by minimising cost function Eqn. (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; and
    2. that $\sum_j w_j = 1$.
Step 3: Compute corresponding data point $Y_i$ regarding
$X_i$ in $d$ using weights $\{w_j\}$ and corresponding
neighbours $\{Y_j\}$ regarding $X_j$ such that
    $Y_i = \sum_j w_j Y_j$
Output:
    $Y_i$: Data point in $d$-dimensional data space
```

Note that as reflected above, being a theoretically well-formed method (as proven in [30]), LLE provides a systematic means for calculating the weights that are required to construct an intermediate ANFIS during the interpolation process. This differs from traditional FRI methods that compute the weights with the ad hoc use of Euclidian distance measures, whilst any other distance metric may be used as an alternative.

III. Proposed Approach

In this section, the main idea underlying the proposed approach and the 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 TD is given, expressed in a collection of input-output pairs $\{(\mathbf{x}, y)\}$, and that two ANFISs (denoted as $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively) are already trained over the source domains *SD*$_1$ and *SD*$_2$. Then, the goal of this work is to generate a new ANFIS $\mathcal{A}_{int}$ over TD, through an innovative way of rule interpolation. Note that here, $\mathbf{x}$ stands for the vector of all input variables.

To be concise, the overall algorithm implementing the proposed approach is shown in Fig. 2, with detailed steps summarised in Alg. III. The entire process of constructing an effective ANFIS with sparse data over TD involves three stages:

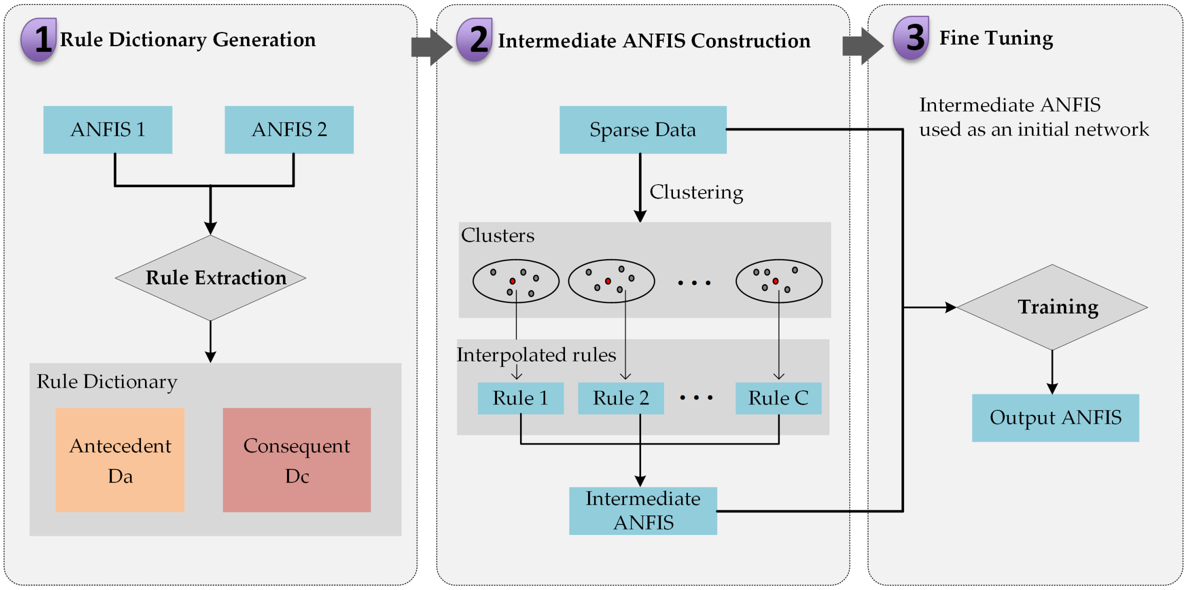


Fig. 2. Flowchart of proposed approach

```
Algorithm III: ANFIS Construction via Group Rule Interpolation
Input:
    1. Two source ANFISs in SDs: $\mathcal{A}_1, \mathcal{A}_2$
    2. Sparse training data in TD
Step 1: Rule Dictionary Generation
    1. Extract fuzzy rules $\{R_i\}$ from $\mathcal{A}_1$ and $\mathcal{A}_2$;
    2. Construct antecedent part of dictionary $D_a$ by Eqn. (9);
    3. Construct consequent part of dictionary $D_c$ by Eqn. (10);
Step 2: Intermediate ANFIS Construction
    1. Divide sparse training data into $C$ clusters, $C$ is decided by Eqn. (11);
    2. For each cluster centre $c^{(k)}$, interpolate new rule $R_k$ using Alg. II-B:
        (a) Select $K$ closest atoms in $D_a$;
        (b) Compute weights $w^{(k)}$ for chosen atoms using Eqn. (13) and (14);
        (c) Generate new rule $R_k$ using weights $w^{(k)}$:
            (c-1) Create rule antecedent using chosen atoms with index set $\mathcal{K}$ in $D_a$ by Eqn. (16);
            (c-2) Create rule consequent using atoms in $D_c$ with the same index set $\mathcal{K}$ by Eqn. (17);
    3. Integrate all interpolated rules.
Step 3: ANFIS Fine-tuning
    Use sparse training data to fine tune intermediate ANFIS.
Output:
    Interpolated ANFIS in TD: $\mathcal{A}_{int}$
```

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

In stage 1, a rule dictionary is firstly generated by separating and reorganising rules extracted from source ANFISs $\mathcal{A}_1$ and $\mathcal{A}_2$. After that, in stage 2, an intermediate ANFIS is interpolated by the following procedure: a) clustering the sparse data of TD 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 below.

A. Rule Dictionary Generation

To support interpolation of (groups of) fuzzy rules that are to be subsequently used for building an intermediate ANFIS (once a certain unmatched observation is given), a rule dictionary 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 $\mathcal{A}_1$ and $\mathcal{A}_2$ consist of n_1 and n_2 rules respectively. Thus, the extracted rules can be expressed in the following format:

$$\begin{aligned} R_i^{\mathcal{A}_1} : if\ x_1\ is\ A_{i1}^{\mathcal{A}_1}\ and \dots and\ x_m\ is\ A_{im}^{\mathcal{A}_1}, \\ then\ y_i = \sum_{j=0}^{m} p_{ij}^{\mathcal{A}_1} x_j \end{aligned} \tag{7}$$

$$\begin{aligned} R_i^{\mathcal{A}_2} : if\ x_1\ is\ A_{i1}^{\mathcal{A}_2}\ and \dots and\ x_m\ is\ A_{im}^{\mathcal{A}_2}, \\ then\ y_i = \sum_{j=0}^{m} p_{ij}^{\mathcal{A}_2} x_j \end{aligned} \tag{8}$$

where m denotes the number of input variables, and $p_{i0}^{\mathcal{A}_t}, t \in \{1, 2\}$ 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:

$$\begin{aligned} D_a =& [d_1^a\ d_2^a\ \cdots\ d_N^a] \\ =& \begin{bmatrix} A_{1,1}^{\mathcal{A}_1} & A_{2,1}^{\mathcal{A}_1} & \cdots & A_{n_1,1}^{\mathcal{A}_1} & A_{n_1+1,1}^{\mathcal{A}_2} & \cdots & A_{N,1}^{\mathcal{A}_2} \\ A_{1,2}^{\mathcal{A}_1} & A_{2,2}^{\mathcal{A}_1} & \cdots & A_{n_1,2}^{\mathcal{A}_1} & A_{n_1+1,2}^{\mathcal{A}_2} & \cdots & A_{N,2}^{\mathcal{A}_2} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ A_{1,m}^{\mathcal{A}_1} & A_{2,m}^{\mathcal{A}_1} & \cdots & A_{n_1,m}^{\mathcal{A}_1} & A_{n_1+1,m}^{\mathcal{A}_2} & \cdots & A_{N,m}^{\mathcal{A}_2} \end{bmatrix} \end{aligned} \tag{9}$$

where each column $d_i^a = [A_{i,1}^{\mathcal{A}_t}\ A_{i,2}^{\mathcal{A}_t} \cdots A_{i,m}^{\mathcal{A}_t}]^T, t \in \{1, 2\}$, (each $A_{i,j}^{\mathcal{A}_t}, j = 1, 2, \dots, m$, contains a fuzzy set value for a given input variable within a certain rule) forming an atom of the dictionary unit D_a, 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

$$\begin{aligned} D_c =& [d_1^c\ d_2^c\ \cdots\ d_N^c] \\ =& \begin{bmatrix} p_{1,0}^{\mathcal{A}_1} & p_{2,0}^{\mathcal{A}_1} & \cdots & p_{n_1,0}^{\mathcal{A}_1} & p_{n_1+1,0}^{\mathcal{A}_2} & \cdots & p_{N,0}^{\mathcal{A}_2} \\ p_{1,1}^{\mathcal{A}_1} & p_{2,1}^{\mathcal{A}_1} & \cdots & p_{n_1,1}^{\mathcal{A}_1} & p_{n_1+1,1}^{\mathcal{A}_2} & \cdots & p_{N,1}^{\mathcal{A}_2} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ p_{1,m}^{\mathcal{A}_1} & p_{2,m}^{\mathcal{A}_1} & \cdots & p_{n_1,m}^{\mathcal{A}_1} & p_{n_1+1,m}^{\mathcal{A}_2} & \cdots & p_{N,m}^{\mathcal{A}_2} \end{bmatrix} \end{aligned} \tag{10}$$

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

B. Intermediate ANFIS Construction

An intermediate ANFIS is a set of new rules interpolated with the assistance of the rule dictionary created as above. The sparse training data given in the form of $\{(\mathbf{x}, y)\}$ within the target domain are partitioned into C clusters on a variable by variable basis by using K-means clustering algorithm. The number of clusters C which is also the expected number of rules in the intermediate ANFIS is decided by Eqn. (11) below:

$$C = \prod_{j=1}^{m} \lfloor \frac{n_1^{(j)} + n_2^{(j)}}{2} \rfloor \tag{11}$$

where $n_1^{(j)}$ is the number of fuzzy sets of the jth variable in $\mathcal{A}_1$, $n_1 = \prod_{j=1}^{m} n_1^{(j)}$; and similarly, $n_2^{(j)}$ is that of $\mathcal{A}_2$, $n_2 = \prod_{j=1}^{m} n_2^{(j)}$.

With respect to the resulting centroid of each cluster, a single newly interpolated rule is generated using LLE through three steps, as outlined in Fig. 3. These three steps are described as follows.

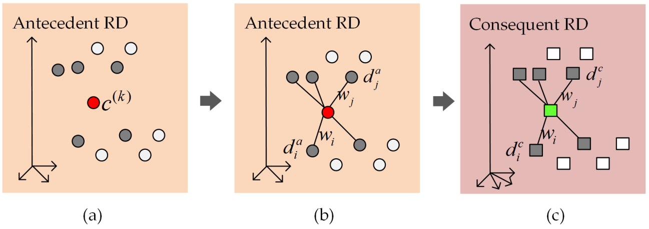


Fig. 3. Illustrative implementation of single rule interpolation: (a) Choosing K closest neighbours. (b) Calculating weights of chosen neighbours. (c) Generating new rule.

1) Choosing K closest neighbours: 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 dictionary D_a, K closest atoms to $c^{(k)}$ is firstly selected using the Euclidian distance metric (though any other distance metric can be utilised as an alternative if preferred):

$$d_i = d(d_i^a, c^{(k)}) = \sqrt{\sum_{j=1}^{m} d(A_{ij}^{\mathcal{A}_t}, c_j)^2} \tag{12}$$

where $d(A_{ij}^{\mathcal{A}_t}, c_j) = |Rep(A_{ij}^{\mathcal{A}_t}) - c_j|, t \in \{1, 2\}$. 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 $\mathcal{K}$.

2) Calculating construction weights: Based on the obtained closest atoms $\{d_i^a | i \in \mathcal{K}\}$, the aim of step 2 is to find the best construction weights that indicate the relative significance of each selected atom. This is achieved by resolving the following optimisation problem:

$$w^{(k)} = \min_{w^{(k)}} ||c^{(k)} - \sum_{i\in\mathcal{K}} Rep(d_i^a) w_i^{(k)}||^2,\ s.t.\ \sum_{i\in\mathcal{K}} w_i^{(k)} = 1 \tag{13}$$

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

$$w^{(k)} = \frac{G^{-1}\mathbf{1}}{\mathbf{1}^T G^{-1}\mathbf{1}} \tag{14}$$

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 $\{d_i^a | i \in \mathcal{K}\}$.

3) Generating new rules: The underlying assumption for generating a new rule is that 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 $\mathcal{K}$ in the antecedent dictionary D_a, then the corresponding consequent can be expressed as the linear combination of the atoms (whose locations are also indexed by $\mathcal{K}$) in the consequent dictionary D_c. Based on this (practically working) presumption, a newly interpolated rule $R_k^{\mathcal{A}'}$ 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:

$$R_k^{\mathcal{A}'}: if\ x_1\ is\ A_{k1}^{\mathcal{A}'}\ and \ldots and\ x_m\ is\ A_{km}^{\mathcal{A}'}, \\ then\ y_k = \sum_{j=0}^{m} p_{kj}^{\mathcal{A}'} x_j \tag{15}$$

where the antecedent part is obtained by

$$A_{kj}^{\mathcal{A}'} = \sum_{i \in \mathcal{K}} w_i^{(k)} A_{ij}^{\mathcal{A}_t},\ j = 1, 2, \cdots, m,\ k = 1, 2, \cdots, C. \tag{16}$$

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

$$p_{kj}^{\mathcal{A}'} = \sum_{i \in \mathcal{K}} w_i^{(k)} p_{ij}^{\mathcal{A}_t},\ j = 0, 1, 2, \cdots, m,\ k = 1, 2, \cdots, C. \tag{17}$$

The above completes the process of producing a newly interpolated rule. By integrating all such interpolated rules, the intermediate ANFIS $\mathcal{A}'$ results. This is operated by putting all the newly generated rules into an ANFIS network, which is an inverse procedure 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 Fig. 4.

C. ANFIS Fine-tuning

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

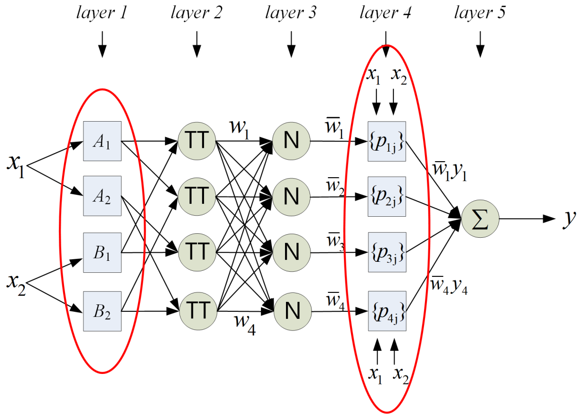


Fig. 4. Integrating newly generated rules as an (intermediate) ANIFS

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 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 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.

D. Distinctions from Existing Work

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 TD. 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 TD.
3) Weights required for performing interpolation are computed differently. For instance, the active set algorithm [39] is employed in the existing work, 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.

E. Complexity Analysis

The time complexity of Alg. III is estimated here. As Step 3 is the same as that for conducting traditional ANFIS training, only the complexity of the procedures regarding rule interpolation (i.e., that of Steps 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

In Step 1 (rule dictionary generation), 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 Step 1 is $n \times O(4m+1) = O(4mn+n)$.

In Step 2 (intermediate ANFIS construction), the sequence of three sub-steps (a), (b) and (c) repeats C times. Particularly, sub-step (a) involves two operations: 1) computing the Euclid distances, of a complexity $O(n)$; and 2) sorting the distances, of a complexity $O(n^2)$. Then, in sub-step (b), the weight is calculated once for each chosen rule, thereby being of a complexity $O(K)$. Finally, sub-step (c) takes a complexity of $O(4m+1)$, in which the weighted average is calculated once for each parameter. Thus, the time complexity for Step 2 is estimated to be $C \times [O(n) + O(n^2) + O(K) + O(4m+1)] = O(Cn^2)$.

Together, the overall time complexity for group rule interpolation bar that required by Step 3 (for running the standard procedure to perform ANFIS fine-tuning with a small number of training data) is estimated to be $O(4mn+n) + O(Cn^2) = O(Cn^2)$. This is practically doable since both n and C are not a very large number.

IV. Experimentation

Both synthetic and real world data are considered in the experiments to qualitatively and quantitatively evaluate the proposed ANFIS construction approach. Section IV-B validates the approach by looking into two synthetic function modelling cases, and in Section IV-C, its effectiveness in dealing with real world situations is shown. Section IV-D discusses the model parameters, and examines the robustness of the proposed approach in response to the use of different amounts of training data. Finally, Section IV-E represents an initial real world application of the proposed method.

A. 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 paper). 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 $\mathcal{A}_1$ on the SD data S_1 is denoted by $E_{\mathcal{A}_1(S_1)}$; that from $\mathcal{A}_2$ on S_2 by $E_{\mathcal{A}_2(S_2)}$; and that from $\mathcal{A}_1$ and $\mathcal{A}_2$ on the TD data T by $E_{\mathcal{A}_1(T)}$ and $E_{\mathcal{A}_2(T)}$ respectively. These RMSEs are computed as below:

$$E_{\mathcal{A}_t(*)} = \sqrt{\frac{\sum_{k=1}^{N_*}(g_k^* - \mathcal{A}_t(x_k^*))^2}{N_*}} \tag{18}$$

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; $\mathcal{A}_t(x_k^*), t \in \{1, 2, ori, int\}$ 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 TD is also generated here for comparison, denoted by $\mathcal{A}_{ori}$. The RMSE of $\mathcal{A}_{ori}$ on the testing data T is denoted as $E_{\mathcal{A}_{ori}(T)}$, and the RMSE of the interpolated ANFIS $\mathcal{A}_{int}$ using the proposed approach on T is denoted by $E_{\mathcal{A}_{int}(T)}$.

B. Experiments on Synthetic Data

Two numerical functions are utilised here to test the proposed approach for working on highly nonlinear 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.

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:

$$y = \frac{sin(2x)}{e^{\frac{x}{5}}} \tag{19}$$

where $x \in [0, 9]$, the shape of this function is plotted in Fig. 5(a).

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 nonlinear function are treated as the data as depicted in Fig. 5(a).

In particular, the left part $x \in [0, 3]$ is used as the first SD for training the first source (or given) ANFIS $\mathcal{A}_1$, and the right part $x \in [6, 9]$ is used as the second SD for training the second source ANFIS $\mathcal{A}_2$. The middle part data $x \in [3, 6]$ forms the TD, which is divided into two sub-parts, with a small portion (20% randomly taken from the TD, 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 II-A. The rules embedded within the trained source ANFISs $\mathcal{A}_1$ and $\mathcal{A}_2$, form the atoms of the rule dictionary. Here, the numbers of membership functions are set to 5 and 6 for $\mathcal{A}_1$ and $\mathcal{A}_2$, 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. 5×5-fold cross validation 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. The mean and standard deviation values of different methods compared are listed in Table I, and the visual results of 5 folds within the total 5×5 folds are shown in Fig. 5(b)-(f).

TABLE I
EXPERIMENTAL RESULTS ON ONE-DIMENSIONAL FUNCTION

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

By examining the experimental results in both Fig. 5 and Table I, it can be seen that the two source ANFISs $\mathcal{A}_1$ and $\mathcal{A}_2$ perform quite well in the corresponding source domains (as $E_{\mathcal{A}_1(S_1)}$ and $E_{\mathcal{A}_2(S_2)}$ are quite small). However, these two ANFISs do not work in the target domain (as $E_{\mathcal{A}_1(T)}$ and $E_{\mathcal{A}_2(T)}$ 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 $\mathcal{A}_{ori}$ 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 $\mathcal{A}_{int}$ improves the results significantly. As shown in the middle part of Fig. 5(b)-(f), its outcome is much closer to the ground truth.

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

$$y = sin(\frac{x_1}{\pi})sin(\frac{x_2}{\pi}) \tag{20}$$

where $x_1 \in [-30, 30], x_2 \in [-10, 10]$. Fig. 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 II
EXPERIMENTAL RESULTS ON TWO-DIMENSIONAL FUNCTION

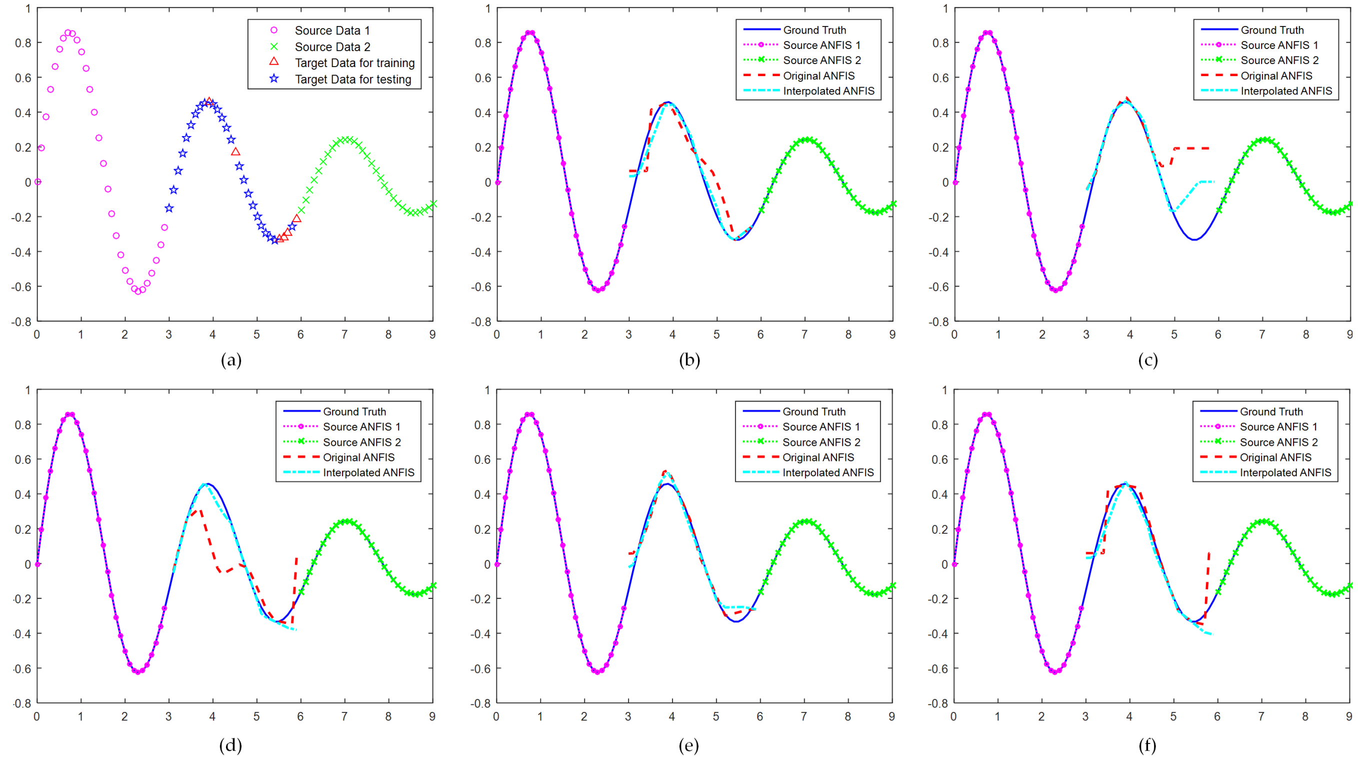
	Mean ± Standard deviation	
	splitting x_1	splitting x_2
$E_{\mathcal{A}_1(S_1)}$	0.024 ± 0.000	0.023 ± 0.000
$E_{\mathcal{A}_2(S_2)}$	0.023 ± 0.000	0.023 ± 0.000
$E_{\mathcal{A}_1(T)}$	1.176 ± 0.029	1.079 ± 0.030
$E_{\mathcal{A}_2(T)}$	1.331 ± 0.022	1.586 ± 0.035
$E_{\mathcal{A}_{ori}(T)}$	0.372 ± 0.063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

The middle column of Table II lists the results of 5×5-fold cross validation for this two dimensional function approximation problem using different ANFISs. As an example, Fig. 6 shows the visual result of one randomly picked fold, where Fig. 6(a) displays the ground truth view of this two-dimensional function in TD; Fig. 6(b) illustrates the result based on the original ANFIS directly learned from the sparse training data; and Fig. 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 SDs and TD defined by splitting the domain of the first input variable x_1. However, similar results can also be obtained via specifying the SDs and TD with regard to the second variable x_2 of this function, as listed in the third, i.e., the right-most column of Table II.

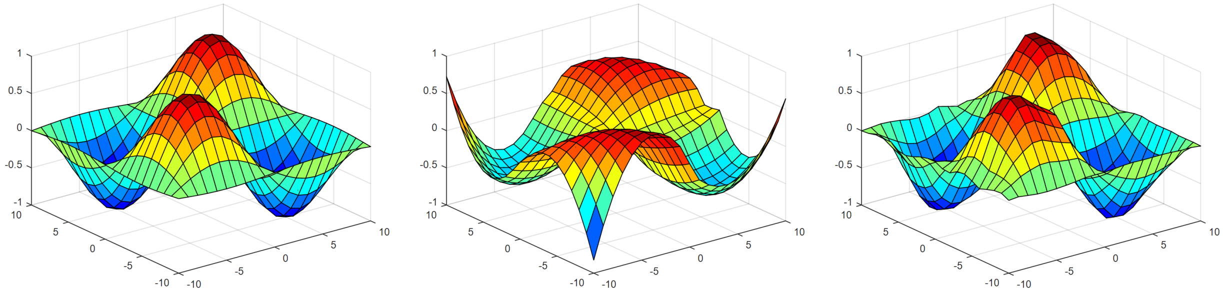
Consider a more general situation where both input variables x_1 and x_2 are divided into three regions. As such there are totally nine subregions, as shown in Fig. 7. This gives rise to a general case in which there may be more than two source



(a) (b) (c)

(d) (e) (f)

Fig. 5. Five folds taken from 5×5-fold cross validation for one-dimensional nonlinear function approximation. (a) Source data and target data used (only one fold of training and testing data is shown here for illustration). (b)-(f) Each of 5 folds cross validation results taken, respectively.



(a) (b) (c)

Fig. 6. One fold of the two-dimensional function-approximation results: (a) Ground truth. (b) Result based on $\mathcal{A}_{ori}$. (c) Result based on $\mathcal{A}_{int}$.

ANFISs for use to interpolate an ANFIS within a given target domain. For instance, if insufficient training data appears in the central subregion of Fig. 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 TD and the other 8 subregions as SDs can be computed, such that the average and standard deviation values of RMSE (over 5×5-fold cross validation) for the original ANIFS $E_{\mathcal{A}_{ori}(T)}$ is 0.215 ± 0.178, whilst that for the interpolated ANFIS is $E_{\mathcal{A}_{int}(T)} = 0.177 \pm 0.081$, showing a remarkable improvement.

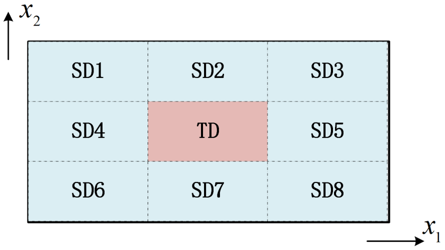


Fig. 7. Function approximation with more than two source domains

3) Situation with no target training data: Consider a further situation where no training data is available in the target domain, using the problem of approximating the function as expressed in Eqn. (19). In the area where $x \in [0.5, 2]$, this one dimensional non-linear function may be interpreted approximately as linear, as shown in Fig. 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 $\mathcal{A}_1$, and that the data within the right part $x \in [1.5, 2]$ is used to train the second source ANFIS $\mathcal{A}_2$. 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 Fig. 8.

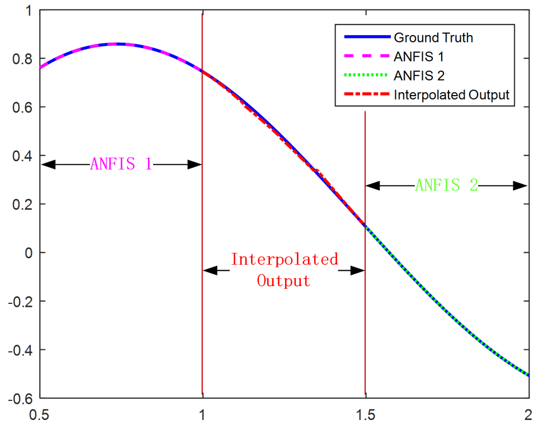


Fig. 8. Function approximation with no target training data

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 Fig. 9. This does not seem to 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 TD and SDs 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.

C. Experiments on Real World Benchmark Data

Twelve popular benchmark regression datasets taken from the KEEL data repository [40] 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 III.

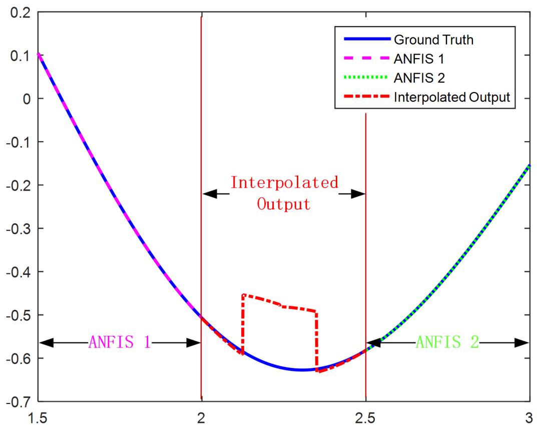


Fig. 9. Performance with no target training data under non-linear situation

TABLE III
Public Dataset Used

Dataset	No.(Attributes)	No.(Instances)
Diabetes	2	1650
Plastic	2	43
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

The generation of the SDs and TD 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 with this variable value being smaller than -40 jointly form the first SD S_1 (of a subtotal of 593 instances), instances with 'Longitude' value larger than 92 form the second SD S_2 (1253 instances), and instances with 'Longitude' value between $[-40, 92]$ form the TD T (332 instances). Similar to the experiments on synthetic data reported earlier, 20% (66 instances) of the data in the TD are used as training data, with the remaining 80% (265 instances) used as testing data. 5×5-fold cross validation results of this experimentation are shown in Table IV. 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 IV, a similar conclusion to what is previously learned from the experiments on synthetic data can be drawn. Both $E_{\mathcal{A}_1(T)}$ and $E_{\mathcal{A}_2(T)}$ are very large, indicating that the ANFISs trained in the SDs are not suitable for the regression problem in the TD. Again, this is not surprising. However, those target domain models $\mathcal{A}_{ori}$

TABLE IV
EXPERIMENTAL RESULTS ON REAL WORLD DATASETS

Datasets	Mean ± Standard deviation of ANFISs			
	$E_{\mathcal{A}_1(T)}$	$E_{\mathcal{A}_2(T)}$	$E_{\mathcal{A}_{ori}(T)}$	$E_{\mathcal{A}_T(T)}$
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

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 $\mathcal{A}_{ori}$ is still not very stable (as indicated by the large standard deviation value in Table IV), 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 $E_{\mathcal{A}_{int}(T)}$ are much smaller than those of $E_{\mathcal{A}_{ori}(T)}$.

D. Experiments with Different Background 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 Fig. 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 [41] (though that piece of existing work is concerned with a weighted approach to FRI).

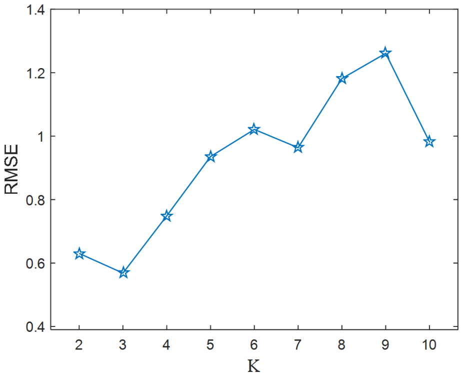


Fig. 10. Performance vs. number of selected closest neighbours

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, Fig. 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 training samples. 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.

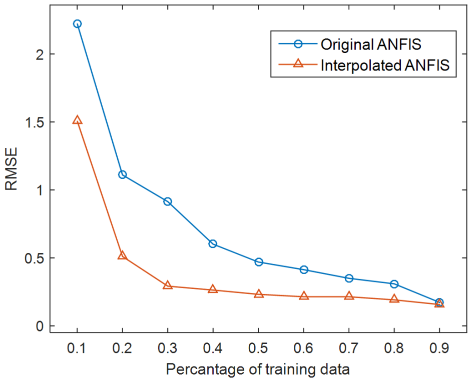


Fig. 11. Performance vs. amount of training data

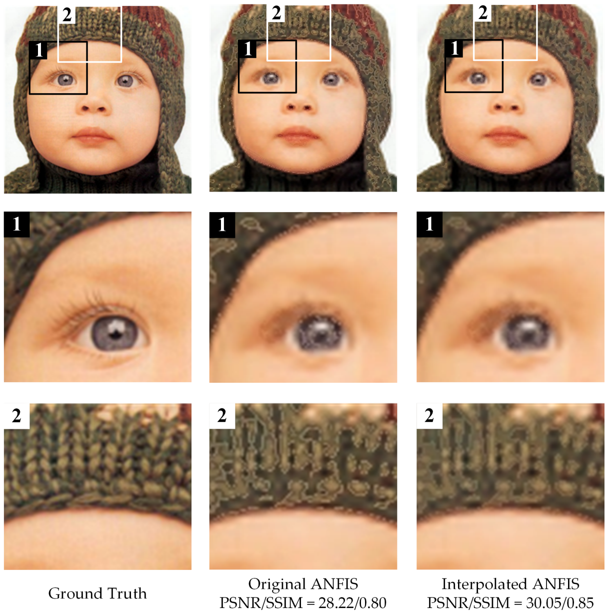
E. Image Super-resolution: An Initial Application

This section presents an initial practical application of the proposed approach on image super resolution (SR). Image SR aims to generate a high resolution (HR) image using a low resolution (LR) one. Amongst various existing SR algorithms, fuzzy rule based SR such as that of [42] learns non-linear mappings from LR images to HR ones through the use of a set of fuzzy rules. To strengthen SR performance, TSK-type ANFISs have recently been utilised [43], where the training images are grouped into a small number of sub-sets, then an individual ANFIS mapping is learned for each sub-set.

In real world, different characteristics of an image may co-exist, be it an LR or HR image. However, the number of training samples between different sub-sets can be rather imbalanced. That is, for certain sub-sets 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. The proposed approach provides an effective solution to such data shortage problems, by choosing two most relevant sub-sets as the SDs and using the corresponding two source ANFISs to support the ANFIS construction in the sub-set (TD) that lacks training samples. The outline of this initial application approach is shown in Alg. IV-E.

```
Algorithm IV-E: Initial application on image SR
Input:
    Training image data set ${\mathbf{Z}}$; Testing LR image $\mathbf{X}$
(Training)
Divide training set into $P$ sub-sets using K-Means;
for $i = 1$ to $i = P$ do
    if $P_i$ contains sufficient data
        Train ANFIS $\mathcal{A}_\mathrm{i}$ with standard training method;
    else
        $S \leftarrow \{S, i\}$
    end if
end for
for each $j \in S$ do
    Choose 2 closest ANFISs as source ANFISs;
    Interpolate ANFIS using proposed approach;
end for
(Training Output: Multiple learned ANFIS models $\{\mathcal{A}_\mathrm{i}\}$)
(Testing)
Pre-processing: Upscale $\mathbf{X}$ by bicubic interpolation;
for each pixel of upscaled image do
    Choose relevant ANFIS model;
    Inference using corresponding ANFIS;
end for
Integrate HR pixels to form HR image $\mathbf{Y}$;
Output:
    HR image $\mathbf{Y}$
```

For simplicity, in this initial experimental study for this real-world application, the number of sub-sets is set to be $P = 3$. That is, Sub-sets 1 and 3 are assumed with sufficient training samples where standard ANFIS learning procedure can be applied, while for Sub-set 2 only sparse training data is available where the corresponding ANFIS is to be constructed using the proposed approach. The visible results are shown in Fig. 12. It can be observed from these results that in the resulting image using the original ANFIS, there are obvious noise and bad edges, and that running the interpolated ANFIS leads to significant improvement. Quantitative performance indices used are those commonly adopted in image super resolution literature, namely, PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structure SIMilarity), whose mathematically definitions are omitted here to save space. The results are given at the bottom of the respective images, conforming to the visual observation.



Ground Truth

Original ANFIS
PSNR/SSIM = 28.22/0.80

Interpolated ANFIS
PSNR/SSIM = 30.05/0.85

Fig. 12. Super-resolution results using different ANFISs.

V. Conclusion

How to construct an effective fuzzy inference system with insufficient training data is a practically important and challenging issue. This paper 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 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.

The proposed approach has been evaluated with two synthetic and twelve real world benchmark datasets, plus a practical application for image super resolution. It would be interesting to extend the experimental investigations with more complicated problems (e.g., consolidating image super resolution application for face recognition). Currently, the simplest K-means algorithm is used for clustering; using an automated data clustering method such as those proposed in [44] would help derive more accurate interpolative results. Also, the present results of devising and running the approach with ANFISs may be extended to other TSK-type fuzzy inference systems. Similarly, instead of using LLE, it is worth examining how other optimisation mechanisms may be adapted for integrated use within the interpolation process. Finally, the current interpolation method terminates once an interpolated ANFIS covering an originally sparse data area is obtained. If the interpolation process iterates, 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 [25] to create novel ANFIS models on the fly, gaining overall inference efficiency as well as effectiveness.

References

[1] J. Qiu, H. Gao, and S. X. Ding, "Recent advances on fuzzy-model-based nonlinear networked control systems: A survey," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1207–1217, 2016.

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

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

[4] E. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Trans. Comput.*, vol. 26, no. 12, pp. 1182–1191, 1977.

[5] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, no. 1, pp. 116–132, 1985.

[6] Y. Chai, L. Jia, and Z. Zhang, "Mamdani model based adaptive neural fuzzy inference system and its application," *Int. J. Comput. Intell.*, vol. 5, no. 1, pp. 22–29, 2009.

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

[8] Y. Jiang, Z. Deng *et al.*, "Recognition of epileptic eeg signals using a novel multiview tsk fuzzy system," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 1, pp. 3–20, 2017.

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

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

[11] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Fuzzy regression transfer learning in takagi–sugeno fuzzy models," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1795–1807, 2017.

[12] Z. Deng, Y. Jiang, F.-L. Chung, H. Ishibuchi, and S. Wang, "Knowledge-leverage-based fuzzy system and its modeling." *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 4, pp. 597–609, 2013.

[13] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Granular fuzzy regression domain adaptation in takagi–sugeno fuzzy models," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 847–858, 2018.

[14] J. Shell and S. Coupland, "Fuzzy transfer learning: methodology and application," *Inf. Sci.*, vol. 293, pp. 59–79, 2015.

[15] D. Tikk, I. Joó, L. T. Kóczy, P. Várlaki, B. Moser, and T. D. Gedeon, "Stability of interpolative fuzzy kh controllers," *Fuzzy Sets Syst.*, vol. 125, no. 1, pp. 105–119, 2002.

[16] L. Kóczy and K. Hirota, "Approximate reasoning by linear rule interpolation and general approximation," *Int. J. Approx. Reason.*, vol. 9, no. 3, pp. 197–225, 1993.

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

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

[19] Z. Huang and Q. Shen, "Fuzzy interpolation and extrapolation: A practical approach," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, pp. 13–28, 2008.

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

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

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

[23] L. Yang and Q. Shen, "Adaptive fuzzy interpolation," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 6, pp. 1107–1126, 2011.

[24] L. Yang, F. Chao, and Q. Shen, "Generalised adaptive fuzzy rule interpolation," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 4, pp. 839–853, 2017.

[25] N. Naik, R. Diao, and Q. Shen, "Dynamic fuzzy rule interpolation and its application to intrusion detection," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 4, pp. 1878–1892, 2018.

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

[27] F. Li, Y. Li, C. Shang, and Q. Shen, "Fuzzy knowledge-based prediction through weighted rule interpolation," *IEEE Trans. Cybern.*, 2019.

[28] 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 Comput.*, vol. 22, no. 10, pp. 3155–3170, 2018.

[29] 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.

[30] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[31] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, 1993.

[32] I. Turkmen, "Efficient impulse noise detection method with anfis for accurate image restoration," *Int. J. Electron. Commun.*, vol. 65, no. 2, pp. 132–139, 2011.

[33] L.-Y. Wei, "A hybrid anfis model based on empirical mode decomposition for stock time series forecasting," *Appl. Soft Comput.*, vol. 42, pp. 368–376, 2016.

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

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

[36] 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.

[37] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proc. 2004 Conf. Comput. Vis. Pattern Recognit.*, vol. 1. IEEE, 2004, pp. I–I.

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

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

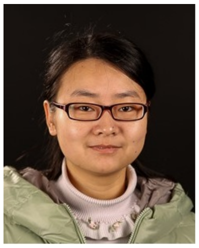
[40] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework." *J. Multi. Valued Log. S.*, vol. 17, 2011.

[41] F. Li, C. Shang, Y. Li, J. Yang, and Q. Shen, "Fuzzy rule based interpolative reasoning supported by attribute ranking," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2758–2773, 2018.

[42] P. Purkait, N. R. Pal, and B. Chanda, "A fuzzy-rule-based approach for single frame super resolution," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2277–2290, 2014.

[43] J. Yang, C. Shang, Y. Li, and Q. Shen, "Single frame image super resolution via learning multiple anfis mappings," in *2017 IEEE Int. Conf. Fuzzy Syst.* IEEE, 2017, pp. 1–6.

[44] T. Boongoen, C. Shang, N. Iam-On, and Q. Shen, "Extending data reliability measure to a filter approach for soft subspace clustering," *IEEE Trans. Syst., Man, Cybern. B*, vol. 41, no. 6, pp. 1705–1714, 2011.

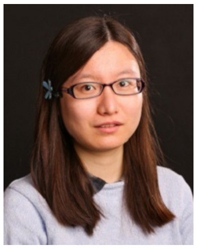


Jing Yang received the B.Sc. degree in information and computing science from Taiyuan Normal University, China, in 2012, and the M.Sc. degree in computational mathematics from Northwestern Polytechnical University, Xian, China, in 2015. She is currently working toward the Ph.D. degree jointly with the School of Computer Science and Engineering, Northwestern Polytechnical University and the Department of Computer Science, Faculty of Business and Physical Sciences, Aberystwyth University, Aberystwyth, U.K. Her research interests include fuzzy systems, machine learning, and image superresolution.

Changjing Shang received the Ph.D. degree in computing and electrical engineering from Heriot-Watt University, Edinburgh, U.K., in 1995.

She is currently a University Research Fellow with the Department of Computer Science, Faculty of Business and Physical Sciences, Aberystwyth University, Aberystwyth, U.K. Prior to joining Aberystwyth, she was with Heriot-Watt, Loughborough and Glasgow Universities. She has published extensively, and supervised more than ten Ph.Ds/PDRAs in the areas of pattern recognition, data mining and analysis, space robotics, and image modeling, and classification.

Ying Li received the Ph.D. degree in circuits and systems from Xidian University, Xi’an, China, in 2002. Since 2003, she has been with the School of Computer Science, Northwestern Polytechnical University, Xi’an, where she is currently a Full Professor. Her current research interests include computation intelligence, image processing and pattern recognition. She has published extensively in the above areas.



Fangyi Li received the B.Sc. degree in Computer Science and Technology from Northwestern Polytechnical University, Xi’an, China, in 2014. She is currently working toward a jointly Ph.D. degree in Computational Intelligence with the School of Computer Science and Engineering, Northwestern Polytechnical University, China, and the Department of Computer Science, Faculty of Business and Physical Sciences, Aberystwyth University, U.K.. Her research interests include fuzzy interpolative reasoning, machine learning and image classification.

Qiang Shen received the Ph.D. degree in computing and electrical engineering from Heriot-Watt University, Edinburgh, U.K., in 1990 and the D.Sc. degree in computational intelligence from Aberystwyth University, Aberystwyth, U.K., in 2013.

He is appointed as the Chair of Computer Science and the Pro Vice-Chancellor Faculty of Business and Physical Sciences, Aberystwyth University, Aberystwyth, U.K. He has authored two research monographs and over 390 peer-reviewed papers.

Dr. Shen was a recipient of the Outstanding Transactions Paper Award from the IEEE.