Rule-Based Models with Information Granules: Enhancements and Applications

by

Ye Cui

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Rule-based models have enjoyed a great deal of interest in the previous decades as one of the development paradigms of intelligent systems. They are regarded as a fundamental vehicle of knowledge representation, serving as a computational platform supporting an array of design practices and analysis of knowledge-based systems and their applications. With the rapidly growing complexity of real-world systems and their ensuing models, and diversity and quality of distributed sources of data, the quest for designing advanced rule-based models becomes evident. In the design of rule-based models, there are two challenging issues. The first is about the scalability of rule-based models when we are faced with high-dimensional data. The equally important task when building rule-based models is to endow such models with a sound measure of quality with which one can efficiently assess the relevance of the results produced by the rules.

In this study, some key design objectives are formulated and pursued. When faced with high-dimensional data, some fundamental limitations such as the concentration effect hamper the design of high-quality rules or even make the design of monolithic models infeasible. To alleviate this problem, an idea of distributed fuzzy rule-based models is formulated, instead of a single monolithic (multivariable) rule-based model. In its realization, a slew of low-dimensional models is built and aggregated. The aggregation is realized by some linear linkage transformation. Such ensembles of models help us avoid a negative effect of the concentration effect. Next, a novel concept of the granular rule-based model is investigated. We show that granular models quantify the relevance of the original rule-based architectures and deliver a granular format of results, namely prediction intervals. The granular results are optimized by engaging the criteria of coverage and specificity of information granules. Subsequently, in order to improve the quality of the models, a comprehensive and systematic way of ranking alternatives in the environment of multicriteria group decision making is proposed by introducing information granules. The underlying decision process is realized with the use of the analytical hierarchy process (AHP), resulting in information granules (fuzzy sets) quantifying degrees of preference and relevance of the weights. A series of experiments are carried out to examine the feasibility of the proposed methods.

Preface

The research in this thesis was performed by Ye Cui under the supervision of Professor Witold Pedrycz.

Chapter 3 of this thesis includes the material published as Y. Cui, H. E, W. Pedrycz, Z. Li. "Designing distributed fuzzy rule-based models" *IEEE Transactions on Fuzzy Systems* (2020). I was responsible for the experimental development, data analysis, and the manuscript composition. W. Pedrycz was the supervisory author and was involved with the concept formation and manuscript composition. H. E and Z. Li were involved with the manuscript composition.

Chapter 4 of this thesis includes the material published as Y. Cui, H. E, W. Pedrycz, Z. Li. "Augmentation of rule-based models with a granular quantification of results" *Soft Computing* (2019). I was responsible for the experimental development, data analysis, and the manuscript composition. W. Pedrycz was the supervisory author and was involved with the concept formation and manuscript composition. H. E and Z. Li were involved with the manuscript composition.

Chapter 5 of this thesis includes the material submitted to *Renewable Energy* as Y. Cui, H. E, W. Pedrycz, A. R. Fayek. "A granular multicriteria group decision making in renewable energy planning problems". I was responsible for the experimental development, data analysis, and the manuscript composition. W. Pedrycz was the supervisory author and was involved with the concept formation and manuscript composition. H. E and A. R. Fayek were involved with the manuscript composition.

Acknowledgements

I am grateful that I could be a student of Professor Witold Pedrycz. His perseverance and broad knowledge in the research are admirable, encouraging and helping me overcome the difficulties step by step.

I would like to express my gratitude to many people, including Dr. Marek Reformat, Dr. Venkata Dinavahi, Dr. Jie Han, Dr. Edmond Lou, Dr. Alireza Sadeghian, Dr. Petr Musilek, Dr. Zhiwu Li, Dr. Aminah Robinson Fayek, Dr. Ming Zuo, Dr. Ergun Kuru and Dr. Witold Krzymien. Their vivid courses and constructive suggestions further improve interest in my Ph. D. study.

In addition, I would like to thank my lab mates for their research discussions and great friends for their company in the daily life, they make my foreign life colorful and impressive.

Last but not least, my whole family, boyfriend Hanyu E, and his parents deserve respect and gratitude for their love and support. For both beloved parents, we always keep in touch by WeChat to share our lives such that we feel like staying around each other. As for Hanyu, thanks for his great care and encouragement, which makes me to be a better girl.

Ye Cui
University of Alberta
June 2022

Table of Contents

Chapter 1. Introduction 1

1.1 Motivation.. 2

1.2 Objectives and originality 3

1.3 Organization.. 5

Chapter 2. Preliminaries 7

2.1 Clustering... 7

2.2 Rule-based models 13

2.3 The principle of justifiable granularity 19

2.4 Prediction interval 21

2.5 Particle swarm optimization 23

2.6 Concentration effect... 25

2.7 Performance indexes.. 26

2.8 Conclusions... 27

Chapter 3. Enhancement of rule-based models—designing distributed models 28

3.1 Distributed rule-based model and its development 28

3.2 From local models to a global model.. 31

3.3 Design of the distributed rule-based model.. 33

3.4 Experimental studies 36

3.5 Conclusions... 47

Chapter 4. Augmentation of rule-based models—with a granular quantification of results 48

4.1 Rule-based models: concise structural and design considerations 49

4.2. Design of prediction intervals for rule-based model 50

4.2.1 Boolean rules 50

4.2.2 Fuzzy rule-based model 52

4.3 Evaluation of quality of information granules of results of rule-based models 53

4.4 Experimental studies 54

4.5 Conclusions 71

Chapter 5. A granular multicriteria group decision making in renewable energy planning problems 72

5.1 Problem formulation – an overall of design process 74

5.2 Literature review 77

5.2.1 The analytic hierarchy process (AHP). 77

5.2.2 Arithmetic of information granules 79

5.2.3 Ranking information granules 82

5.2.4 Information granularity 84

5.3 An overall method 86

5.4 Experimental study 88

5.5 Conclusions 95

Chapter 6. Conclusions & future studies 97

6.1 Conclusions 97

6.2 Future studies 98

6.2.1 From numeric to granular rule-based model 98

6.2.2 Analysis and design of distributed granular rule-based models 99

6.2.3 Augmentation of rule-based models—distributed models with a biclustering algorithm 99

Bibliography 100

Appendix 116

List of Tables

Table 3.1 Development strategies of one-dimensional rule-based model. 29

Table 3.2 *RMSE* obtained for the distributed rule-based model and the monolithic rule-based model involving selected numbers of their rules. 39

Table 3.3 Dataset information. 41

Table 3.4 *RMSE* for the distributed and monolithic rule-based model involving selected numbers of rules. 42

Table 3.5 Improvement (*RMSE*) of distributed one-dimensional rule-based model 1 (compared to entire feature space) for testing data. 44

Table 3.6 Accuracy (%) obtained for the distributed rule-based model and FDSS.......... 46

Table 4.1 Performance of the model reported for selected values of *c* and *α*=0.05. 55

Table 4.2 Performance of the model reported for selected values of *c* with optimized *m*.56

Table 4.3 Comparing performance of the model for one-dimensional data. 58

Table 4.4 Performance of the model reported for selected values of *c* for *α*=0.05........... 59

Table 4.5 Performance of the model reported for selected values of *c* with optimized *m*.60

Table 4.6 Performance of the models for two-dimensional data. 62

Table 4.7 Performance of the model reported for selected values of *c* and *α*=0.05. 63

Table 4.8 Performance of the model reported for selected values of *c* with optimized *m*.64

Table 4.9 IRIS data-comparative analysis. 66

Table 4.10 Performance of the model reported for selected values of *c* and *α*=0.05. 67

Table 4.11 Performance of the model reported for selected values of *c* with optimized *m*. 68

Table 4.12 Comparing performance of the model produced for Banknote data. 70

Table 5.1 The eigenvectors and values of inconsistency index of pairwise comparison matrices. 89

Table 5.2 Maximal eigenvectors and inconsistency index of pairwise comparison matrix. 91

Table 5.3 Ranking results obtained using different ranking methods. 94

Table 5.4 Ranking results with different methods. 95

List of Figures

Figure 1.1 Roadmap of the research. 4

Figure 2.1 Characteristic and membership functions with (*a*) clustering: *K*-means (*b*) fuzzy clustering: *FCM*. 15

Figure 2.2 The performance indexes as a function of the upper bound. 21

Figure 2.3 Confidence and prediction interval as a function of *x*. Solid curve: confidence bounds, dashed curve: prediction bounds. 22

Figure 2.4 Fitness function. 23

Figure 2.5 The distribution of distances for various number of variables. 26

Figure 3.1 Overall architecture of distributed rule-based model. 31

Figure 3.2 Average performance index for single input rule-based models: (*a*) optimal number of rules (*b*) corresponding values of *RMSE* index. 38

Figure 3.3 Input-output plots of one-dimensional models with superimposed data, where the stars are the original points and the piecewise linear is described in the right bottom part: (*a*) the 13th variable (*b*) the 2nd variable. 38

Figure 3.4 *RMSE* values versus *c* for the rule-based model obtained for the (*a*) the 13th variable and (*b*) the 2nd variable. 39

Figure 3.5 Radar plot of errors shown for the individual data (testing set); the circle is the average value of error. (*a*) distributed one-dimensional model (*b*) two-input rule-based model; monolithic model: (*c*) *c*=3, (*d*) *c*=5, and (*e*) *c*=8. 40

Figure 4.1 Prediction bounds produced by rule-based models for selected values of *c*: (*a*) *c*=1 (*b*) *c*=3 (*c*) *c* =9. 51

Figure 4.2 Prediction bounds produced by rule-based models for selected values of *c*: (*a*) *c*=3 (*b*) *c*=9. 53

Figure 4.3 An overall design of granular rule-based model. 54

Figure 4.4 Performance of the model as a function of *m*; shown are curves for different values of *c*. 57

Figure 4.5 *V* as a function of *c*. 57

Figure 4.6 Two-dimensional synthetic data. 59

Figure 4.7 Performance of the model as a function of *m*; shown are curves for different values of *c*. 61

Figure 4.8 *V* regarded as a function of *c*. 61

Figure 4.9 Prediction intervals vs. numeric outputs for *c*=9. 62

Figure 4.10 Performance of the model as a function of *m*; shown are curves for different values of *c*. 65

Figure 4.11 *V* as a function of *c*. 65

Figure 4.12 Prediction intervals vs. numeric outputs with cluster 8. 66

Figure 4.13 Performance of the model as a function of *m*; shown are curves for different values of *c*. 68

Figure 4.14 *V* regarded as a function of *c*. 69

Figure 4.15 Prediction intervals vs. numeric outputs with cluster 9. 70

Figure 5.1 Criteria-decision-makers (DM$_j$) array. 74

Figure 5.2 Elevation of type of information granule reflective with the movement from individual DMs to a group decision-making. 75

Figure 5.3 Difference of results (consistency, distance between fuzzy sets and rankings) between original and modified *R* as a function of ε.. 79

Figure 5.4 Addition and multiplication of triangular fuzzy numbers. 82

Figure 5.5 Example of ranking. 83

Figure 5.6 Example of resulting information granular (interval-valued or triangular fuzzy set)... 85

Figure 5.7 Visualization of $\boldsymbol{e}^{[i,j]}$... 90

Figure 5.8 Pairwise comparison matrices. 92

Figure 5.9 Resulting fuzzy sets (*a*) interval-valued (*b*) weighted sum of triangular fuzzy sets for 3 alternatives. 93

Figure 5.10 Ranking of fuzzy sets: (*a*) interval-valued (*b*) weighted sum of triangular fuzzy sets... 95

Chapter 1. Introduction

In two-valued logic, there are only the concepts of false and true. However, the introduction of ambiguity (fuzzy set) makes it possible to describe the logic with weights, breaking the traditional regulation. Fuzzy sets form a class of objects with a continuum of grades of membership coming from the range of values in [0,1] [1]. Fuzzy rule-based models are proposed based on the concept of fuzzy sets [2], [3]. Fuzzy rule-based model is a system that describes the relationships among input and output variables expressed at the level of information granules (fuzzy sets). Within the realm of rule-based modeling, there has been a plethora of methodologies, designs, and analyses supporting the development of the models [4]–[10]. Fuzzy rule-based modes have been studied, developed, and applied to control, data mining marketing, and decision making, among others [11]–[14].

Granular Computing (GrC) [15]–[17] has emerged to process information, especially in the area of computational intelligence and human-centric systems [18]. In GrC, the variables are represented in an abstract way and then the resulting information granules, formalized as fuzzy sets, interval-valued sets, rough sets, etc., are used in complex problem solving [19]. It is beneficial to design a comprehensive framework to develop and process information granules with an appropriate methodology. Then, the rule-based model is presented as that well-structured framework since it can handle numeric and linguistic information simultaneously.

1.1 Motivation

In the design of rule-based models, there are two aspects we need to pay attention to. One is that with the emergence of big data in recent decades, fuzzy rule-based models meet two other challenges, namely accuracy and computing overhead. There is no doubt that the big data challenge comes hand in hand with the computing complexity and during the design process of local models, the distance between values will be influenced by the concentration effect while processing the high-dimensional data [20]. In order to address these two problems, distributed fuzzy rule-based models are applied. That is, assign partitioned input data into modular models such that only a few variables are used for the construction of each fuzzy rule-based model. This issue has not received enough attention so far and thus calls for more investigations.

The other aspect worthy of much attention is that the rule-based models and fuzzy rule-based models in a way they have been constructed (optimized), validated, and used, are in essence regarded as numeric constructs. Thus, the results produced by any model do not fully coincide with the experimental data. It becomes highly desirable, if not imperative, to come up with models augmented by the quantification of their quality and results. From a perspective of the structure of the rule-based model, one can regard it as a collection of local regressions augmented with some aggregation mechanism. In these local models, an estimation of the parameters is carried out in a "standard" way (typically by considering a least square error optimization method) and there are statistical means to assess the quality of the model by supplying the corresponding confidence intervals and prediction intervals for the output generated by the model. In linear regression models, the concepts and algorithms supporting a formation of prediction intervals are well documented and widely utilized. This avenue of model evaluation has not been investigated and pursued in terms of algorithmic developments in fuzzy rule-based modeling. As a result, there are a number of open problems with regard to prediction intervals in rule-based models. Hence their formal formulation and ensuing algorithms deserve careful attention.

1.2 Objectives and originality

Objectives

In the presence of high dimensional data, we are faced with some algorithmic challenges. In addition, there is no ideal rule-based model that can fully coincide with the original data. We propose distributed and granular rule-based models to address the above stated problems, respectively. The design of the distributed model is based on the partitioned data, which effectively mitigate the issue caused by large-sized data. The proposed granular model brings the rule-based model to a new conceptual and algorithmic level, producing the interval-valued sets to further quantify the performance of the model and improve its credibility. When it comes to the concept of information granules, their combination with decision making helps solve an actual problem (the site selection of renewable energy sources), and the sites are evaluated by the granular preference results. The objectives and relationships among the main pursuits of this study are displayed in Figure 1.1.

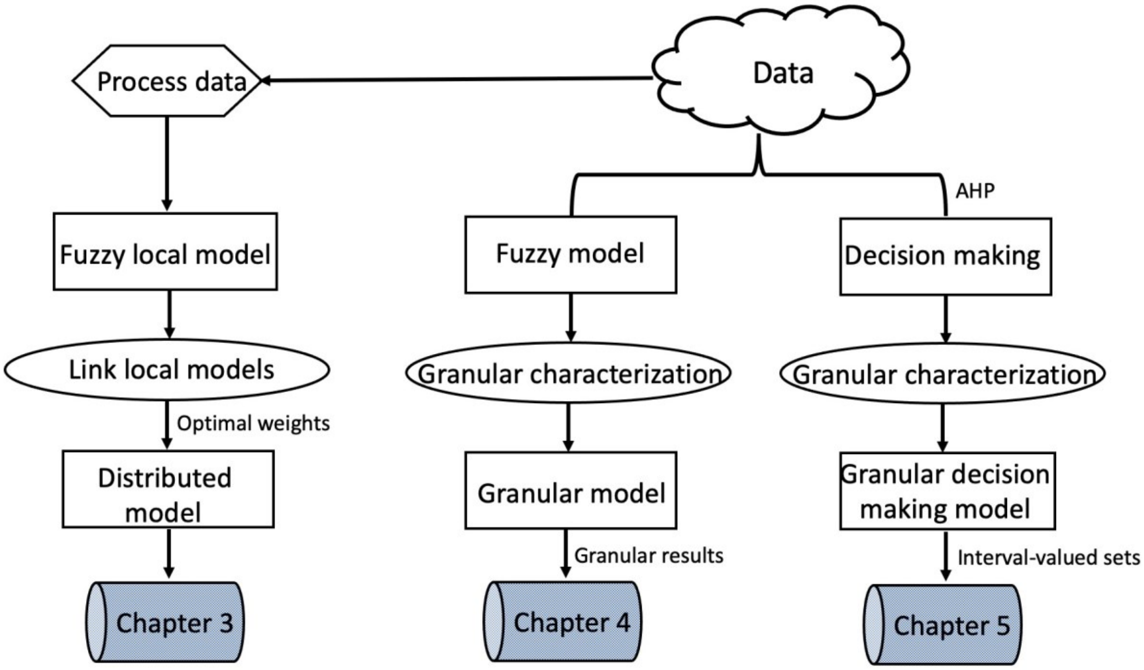


Figure 1.1 Roadmap of the research.

- For high-dimensional data, we improve the design methodology and quality of fuzzy rule-based models by introducing a design of one-dimensional local model (rule-based model) aiming at dealing with the high dimensionality of the input space and improving the computing efficiency. Then the distributed local models are aggregated.
- The (fuzzy) rule-based models are improved by the association of numeric results produced by the rule-based model with the granular characterization (in the form of prediction intervals) and then we introduce the mechanism of the performance assessment. That is, the quality of the granular model is evaluated by the concepts of coverage and specificity, and the combination of these two measures.
- The principle of justifiable granularity and the concept of fuzzy sets applied to the Analytic Hierarchy Process (AHP) algorithm generates the granular (interval-valued) results for assessing the alternative sites of solar energy. Then, the quality of granular results has to be evaluated following the ranking fuzzy sets method.

Originality

The originality of the study exhibits several facets.

- The development of a novel distributed rule-based model avoids the issue of the concentration effect; during the design process of distributed models, the membership grades as the input are transformed into the output space by the optimal linkage matrix, which creates a new avenue for the design of rule-based models.
- The degree of granularity of the rules is proposed and optimized. This problem, although of practical relevance, has not been studied in the past.
- The concept of the prediction interval is combined with rule-based models, resulting in interval information granules to improve the quality of the model.
- A novel granular group decision-making model is developed, which builds upon the combination of information granules and the AHP algorithm. The preference for alternatives is evaluated based on the obtained granular results.

1.3 Organization

The organization of the thesis is as follows:

In Chapter 2, we provide a critical literature review, including the concepts and the application of clustering algorithms, optimization algorithms, rule-based models and information granules, etc. They are the fundamental elements in this study.

In Chapter 3, we focus on the augmentation mechanism of rule-based models for dealing with large or high-dimensional data. The data are separated into local models, generating partition matrices that are then mapped to the distributed rule-based models. Finally, the results are aggregated.

Chapter 4 elaborates upon the combination of rule-based models and granular quantifications, which forms the granular rule-based model. The numeric output of the basic rule-based model is augmented by the granular output space, which further improves the model reliability.

In Chapter 5, a granular multicriteria group decision making is presented to solve the renewable energy planning problem. Compared with the traditional decision-making model, our model produces granular results. Thus, the plan is assessed based on both the granular preference degrees of alternatives with respect to the set of criteria and the weights of the corresponding criteria.

Finally, the overall summary is provided and the future research tasks are outlined in Chapter 6.

Chapter 2. Preliminaries

In this chapter, we introduce some fundamental methods related to our research. Specifically, the clustering algorithms are covered in Section 2.1. In Section 2.2, we introduce the key model-rule-based model. Subsequently, in Sections 2.3 and 2.4, ideas of information granules and prediction intervals are summarized to lay the foundation for the studies reported in Chapters 4 and 5, respectively. Then a commonly used optimization method (Particle Swarm Optimization) is introduced in Section 2.5. Next, the concentration effect is introduced in Section 2.6. Finally, some evaluation methods are summarized in Section 2.7.

2.1 Clustering

Clustering is an unsupervised learning method, devoted to dividing data into a number of groups (clusters), where the data points in the same group are similar to each other while dissimilar to the points in other groups. In our study, two clustering algorithms, namely *K*-Means and Fuzzy *C*-Means (*FCM*) are reported.

A. *K*-Means

K-Means method [21], [22] is commonly used because of its simplicity [23]–[26]. It is an iterative algorithm to partition data $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ in the $\boldsymbol{R}^n$-dimensional space, into a collection of *c* disjoint subsets (clusters). It aims at minimizing the objective function in the following form to find the optimal cluster centers for the data.

$$J = \sum_{i=1}^{c} \sum_{x_k \in S_i} \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 \tag{1}$$

where S_i is the i-th cluster and $\boldsymbol{v}_i$, $i = 1,2,\dots,c$, is the center of S_i. $\|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2$ is the distance defined as

$$\|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 = \sum_{j=1}^{n} \frac{\left(x_{kj} - v_{ij}\right)^2}{\sigma_j^2} \tag{2}$$

where σ_j is the standard deviation of the j-th feature of data.

The minimization of the above objective function is done in an iterative way. First, we set the number of c clusters and select cluster centers randomly. Next, we calculate the distance between data points and cluster centers. For the k-th data, it is assigned to i-th cluster if this point is closer to i-th cluster than to the other centers, as shown below,

$$S_i = \left\{\boldsymbol{x}_k \middle| \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 \leq \left\|\boldsymbol{x}_k - \boldsymbol{v}_j\right\|^2, \forall j \in [1,c], j \neq i\right\} \tag{3}$$

Then we compute the average value of the points in each cluster as the new updated cluster center (prototype).

$$\boldsymbol{v}_i = \frac{\sum_{x_k \in S_i} \boldsymbol{x}_k}{N_i} \tag{4}$$

where N_i is the number of data in i-th cluster. The process of calculating distance and determining to which cluster the data points belong is terminated when the new cluster center is close to the previous one. The detailed process of K-Means clustering is shown as follows:

```
Input:  data, the number of clusters $c$, maximal number of iterations $iteration$ and
        threshold $\tau$
Output: prototypes

Initialize prototypes $\boldsymbol{v}^0, \boldsymbol{v}^1$ randomly, $iter = 0$
While $\left|\boldsymbol{v}_i^{iter+1} - \boldsymbol{v}_i^{iter}\right| > \tau; i \in [1,c]$
   {
    Group the data into corresponding cluster
    $S_i = \left\{ \boldsymbol{x}_k \middle| \left\|\boldsymbol{x}_k - \boldsymbol{v}_i^{iter}\right\|^2 \le \left\|\boldsymbol{x}_k - \boldsymbol{v}_j^{iter}\right\|^2, \forall j \in [1,c], j \neq i \right\}$
    Compute new prototypes with
    $\boldsymbol{v}_i^{iter+1} = \frac{\sum_{x_k \in S_i} x_k}{N}; i = 1,2,\dots,c$
    $iter = iter + 1$
    If $iter > iteration$
      {
        break
      }
}
```

B. Fuzzy C-Means

Fuzzy *C*-Means [27]–[30] is a clustering algorithm in which a dataset is grouped into *c* clusters with every data point belonging to each cluster to a certain degree. It was first reported in the literature by Dunn in 1973 [31] and then generalized by Bezdek in 1981 [32]. The objective of the *FCM* algorithm is to minimize the weighted Euclidean distance between data points and clustering centers, namely prototypes. As before, we also assume the data in $\boldsymbol{R}^n$. The objective function to be minimized comes in the following form

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 \tag{5}$$

where u_{ik} stands for the degree of membership of k-th data clustered into the i-th cluster; the partition matrix includes a collection of membership grades u_{ik}, see $U = [u_{ik}]$, $i = 1,2,\dots,c; k = 1,2,\dots,N$. The partition matrix satisfies two requirements: $u_{ik} \in [0,1]$ and $\sum_{i=1}^{c} u_{ik} = 1$, $k = 1,2,\dots,N$. The distance $\|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2$ is the same as in (2). The fuzzification coefficient *m* is greater than 1, commonly its value is selected to be 2. $\boldsymbol{v}_i$, $i =$

$1,2,\ldots,c$, are a collection of prototypes. For the *k*-th input data, the result of being clustered to *i*-th cluster, in terms of the membership grades [33]–[35], is computed as

$$u_{ik}=\frac{1}{\sum_{j=1}^{c}\left(\frac{\|\boldsymbol{x}_k-\boldsymbol{v}_i\|}{\|\boldsymbol{x}_k-\boldsymbol{v}_j\|}\right)^{2/(m-1)}} \tag{6}$$

where the distance between data point and prototype is in the form (2). The *i*-th prototype is expressed in the form

$$\boldsymbol{v}_i=\frac{\sum_{k=1}^{N}u_{ik}^{m}\boldsymbol{x}_k}{\sum_{k=1}^{N}u_{ik}^{m}} \tag{7}$$

The *FCM* algorithm is realized as a sequence of steps where the partition matrix and prototypes are updated until the minimized value of the objective function reaches the predefined threshold.

```
Input:  data, the number of clusters c, the fuzzification coefficient m, maximal number
        of iterations $iteration$ and threshold $\tau$
Output: partition matrix U, prototypes $\boldsymbol{v}_i$; $i = 1,2,\ldots,c$
Initialize prototypes randomly, $iter = 0$
Compute the partition matrix $u_{ik}$ and the value of objective function $J$ with
$u_{ik}=\frac{1}{\sum_{j=1}^{c}\left(\frac{\|x_k-v_i\|}{\|x_k-v_j\|}\right)^{2/(m-1)}}$;$i,j = 1,2,\ldots,c; k = 1,2,\ldots,N$
$J=\sum_{i=1}^{c}\sum_{k=1}^{N}u_{ik}^{m}\|\boldsymbol{x}_k-\boldsymbol{v}_i\|^2$
While the value of objective function> $\tau$
   {
     Obtain a new partition matrix and prototypes as
     $u_{ik}=\frac{1}{\sum_{j=1}^{c}\left(\frac{\|x_k-v_i\|}{\|x_k-v_j\|}\right)^{2/(m-1)}}$ ; $i,j = 1,2,\ldots,c; k = 1,2,\ldots,N$
     $\boldsymbol{v}_i=\frac{\sum_{k=1}^{N}u_{ik}^{m}\boldsymbol{x}_k}{\sum_{k=1}^{N}u_{ik}^{m}}$; $i = 1,2,\ldots,c$
```

```
        Calculate the value of the objective function $J$
        $iter = iter + 1$
        If $iter > iteration$
          {
            break
          }
    }
```

There were some improvements to the *FCM* algorithm. The Gustafson-Kessel (GK) clustering algorithm proposed by Gustafson-Kessel in [36] could be viewed as a *FCM* variant employing an adaptive norm distance to find ellipsoidal shaped clusters. The distance (2) is defined as $\|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 = (\boldsymbol{x}_k - \boldsymbol{v}_i)^T M_i(\boldsymbol{x}_k - \boldsymbol{v}_i)$, where M_i is a symmetric positive definite matrix. The improvement of GK algorithm is illustrated by clustering two classes that had some degree of overlap. A feature weight-based *FCM* (WFCM) algorithm was proposed in [37], which is similar to GK, but the symmetric positive definite matrix M_i is fixed for different clusters. Subsequently, the gradient-based Fuzzy *C*-Means (GBFCM) algorithm was proposed in [38] to present one datum at a time to the network, and the minimization of the objective function is completed using the gradient descent method. However, the minimization process of *FCM* is proceeded by solving two equations alternatively in an iterative way and in each iteration, all the data are used. Compared to *FCM*, GBFCM is very competitive in terms of speed and stability of convergence for the data. In [39], the Fuzzy *C*-Means with Focal Point algorithm (FCMFP) was proposed and applied to bearing fault diagnosis, the function (1) is rewritten by the introduction of a regularization term with a focal point $\boldsymbol{p}$ as $J = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m}\|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 + \zeta\sum_{i=1}^{c}\|\boldsymbol{p} - \boldsymbol{v}_i\|^2$. If $\boldsymbol{p}$ is far enough from the data points, as the *i*-th cluster approaches $\boldsymbol{p}$ there is no data belonging to it and thus the membership values will tend to zero. In practice, this is equivalent to moving the prototype $\boldsymbol{v}_i$. This allows one to obtain different clusters, depending on the different focal points. In [40], a new hybrid method, namely FCM-ELPSO was proposed, which combined *FCM* with an improved version of PSO and then used a special index and the objective function value as cluster validity indexes to evaluate the clustering effect. In general, *FCM* and its improved versions are relatively easy to implement and as such they serve as important tools to form the condition part of rule-based models.

C. Biclustering algorithm

It was first introduced by Hartigan in 1972 [41]. Unlike clustering, it aims to simultaneously cluster both rows and columns for a data matrix. The biclustering algorithm can be realized in different ways. In general, they can be categorized into five directions; that is, Iterative row and column clustering combination [42], [43], divide and conquer [41], greedy iterative search [44], exhaustive bicluster enumeration [45], and distribution parameter identification [46]. We briefly introduce the 'Blocking clustering' method as an example, which is the representative of the divide and conquer category.

Blocking clustering was first proposed by Hartigan [41], which is called 'direct clustering'. Its main idea is to find *K* appropriate clusters and in each cluster all data are equal. Duffy *et al.* made it known as 'block clustering' [47] and an improvement was reported by Tibshirani *et al.* in 1999 [48], where the entire data are split into *M* (*M*>*K*) biclusters that are then recombined. The detailed process in [48] is as follows:

1. Begin with the entire data (*X*) into one block.
2. Sort the rows (or columns) by row (resp. column) mean.
3. At each stage, find the row or column split of all existing blocks into two pieces, choosing the one that produces largest reduction in the total performance index, the index is sum of squares, say

$$Q = \sum_{p} \sum_{i,j \in B_p} \left(x_{ij} - b_p \right)^2$$

   where B_p is the p-th cluster (block), p=1, 2, …, M, and b_p is the average value of x_{ij} in the cluster B_p.
4. The splitting is continued until *M* blocks are obtained. And then blocks that have approximative b_p are recombined until the required *K* blocks are obtained.

2.2 Rule-based models

Rule-based architectures are commonly used for system modeling [3], [49]. Their essence is to describe the potential relationship between input and output in the form of if-then rules.

$$\text{If } \textit{antecedent} \text{ then } \textit{consequent} \tag{8}$$

There are two main categories: Mamdani [13] and Takagi-Sugeno rule-based models [10]. In this thesis, we consider the rule-based models in the Takagi-Sugeno structure. Assume that we have c if-then rules and the combined input-output pairs of data are positioned in the $\boldsymbol{R}^{n+1}$ dimensional space, e.g. [$\boldsymbol{x}_k$ $target_k$], $k = 1,2,\dots,N$. For the *i*-th rule, it is shown as follows

$$\text{If } \boldsymbol{x} \text{ is } A_i(\boldsymbol{x}), \text{then } y \text{ is } L_i(\boldsymbol{x}) \tag{9}$$

where $i = 1,2,\dots,c,$ usually the condition part $A_i(\boldsymbol{x})$ is a set or fuzzy set defined in the multi-dimensional input space. The conclusion part $L_i(\boldsymbol{x})$ is a certain function, such as a constant function, linear function, polynomial function, etc., where constant function and linear function are commonly used. Researchers usually set the prototypes of the output space ω_i as the constant function $L_i(\boldsymbol{x}) = \omega_i$ and choose $L_i(\boldsymbol{x}) = a_{i0} + \boldsymbol{a}_i^T\boldsymbol{x}$ as linear function. The rationale behind our choice is that the Takagi-Sugeno rule-based model can be regarded as a combination of linguistic description (in the antecedent part) and function (in the consequent part) modeling. Since the architecture of the (fuzzy) rule-based model is determined, for any input data $\boldsymbol{x}$, the corresponding output ($\widehat{\boldsymbol{y}}$) of the model is the linear combination of the conclusions of the rules

$$\widehat{\boldsymbol{y}} = \frac{\sum_{i=1}^{c} A_i(\boldsymbol{x}) L_i(\boldsymbol{x})}{\sum_{i=1}^{c} A_i(\boldsymbol{x})} \qquad (10)$$

The set $A_i(\boldsymbol{x})$ stands for the degree of activation of the rule, $i = 1,2,\dots,c,$ Thus, the sum of c activation results is 1, saying $\sum_{i=1}^{c} A_i(\boldsymbol{x}) = 1$, so that the output can be rewritten as

$$\widehat{\boldsymbol{y}} = \sum_{i=1}^{c} A_i(\boldsymbol{x}) L_i(\boldsymbol{x}) \qquad (11)$$

A. Design of information granules of the antecedent part of the rules

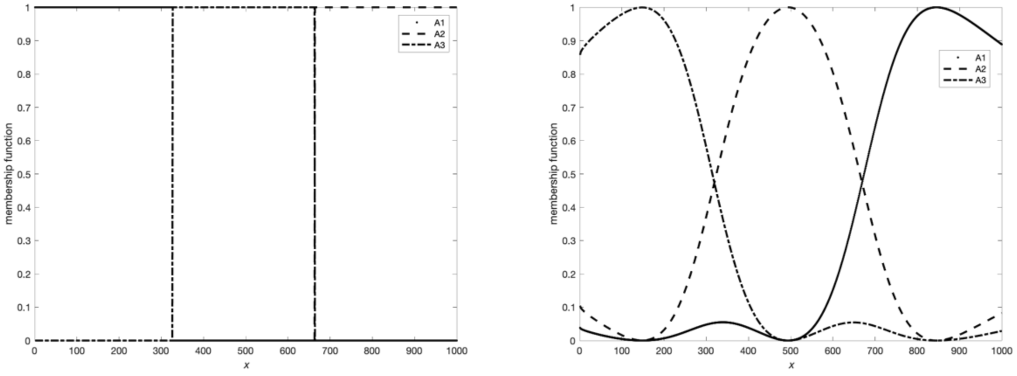
As mentioned above, we have input-output data in the $\boldsymbol{R}^{n+1}$ dimensional space. The commonly used way to form information granules A_i is made through clustering or fuzzy clustering, which can be carried out in the input space or combined input-output space. Cluster these data into c clusters and the results are given in the form of c prototypes whose coordinates $\boldsymbol{v}_i$ is located in the input space and ω_i is positioned in the output space, respectively.

Two types of information granules are formed:

(i) *set information granules*: Here we consider the use of *K*-Means. The clustering method returns *c* centers and splits the data into *c* sets.

(ii) *fuzzy set information granules*: Here *FCM* algorithm is considered. The generated characteristic functions are generalized into a collection of fuzzy sets. An additional fuzzification coefficient *m* offers some flexibility modifying the shape of the obtained membership functions.

In what follows, we show an example of information granules (membership functions) generated by clustering and fuzzy clustering (*m*=2), see Figure 2.1. In comparison with the characteristic functions *K*-means, *FCM* produces membership functions. The shape of these membership functions is impacted by the values of the fuzzification coefficient.



(*a*) (*b*)

Figure 2.1 Characteristic and membership functions with (*a*) clustering: *K*-means (*b*) fuzzy clustering: *FCM*.

B. Estimation of optimal parameters of the consequent part of the rules

Considering the constant function, we could use the prototypes ω_i or the weighted values of output space as the constants, saying

$$L_i(\boldsymbol{x}) = \frac{\sum_{k=1}^{N} A_i(\boldsymbol{x}_k) target_k}{\sum_{k=1}^{N} A_i(\boldsymbol{x}_k)} \tag{12}$$

When it comes to the linear function, it can be described in different ways

$$L_i(\boldsymbol{x}) = \omega_i + \boldsymbol{a}_i^T(\boldsymbol{x} - \boldsymbol{v}_i) \tag{13}$$

or

$$L_i(\boldsymbol{x}) = a_{i0} + \boldsymbol{a}_i^T \boldsymbol{x} \tag{14}$$

where $\boldsymbol{a}_i$ is a vector of parameters of the *i*-th local function. The formulation of the local model makes it directly linked with the *i*-th rule. It is evident that the function (13) passes through the cluster point $(\boldsymbol{v}_i, \omega_i)$.

Here we focus on the estimation of the parameters standing in $L_i(\boldsymbol{x})$; refer to the linear function (13). To determine the parameters $\boldsymbol{a}_i$ for the i-th rule, let us introduce the following notation describing a transformation of the original data

$$\boldsymbol{z}_k = \boldsymbol{x}_k - \boldsymbol{v}_i \tag{15}$$

$$target_k^{'} = target_k - \omega_i \tag{16}$$

The model can be expressed now as

$$h(\boldsymbol{z}_k) = \boldsymbol{a}^T \boldsymbol{z}_k \tag{17}$$

This entails that the data are now considered as the pairs $(\boldsymbol{z}_k, target_k')$, which are formed on the basis of (13). In other words, $L_i(\boldsymbol{x}_k) = \omega_i + h(\boldsymbol{z}_k)$.

The minimized performance index *Q* (distance between the data output and model output) reads as follows

$$Q = \frac{1}{N}\sum_{k=1}^{N}(target_k{}' - h(\boldsymbol{z}_k))^2 \tag{18}$$

We organize the data in the vector- matrix format

$$Z = \begin{bmatrix} \boldsymbol{z}_1^T \\ \boldsymbol{z}_2^T \\ \cdots \\ \boldsymbol{z}_N^T \end{bmatrix}; \quad \boldsymbol{target'} = \begin{bmatrix} target_1{}' \\ target_2{}' \\ \cdots \\ target_N{}' \end{bmatrix} \tag{19}$$

The optimized vector of parameters of $\boldsymbol{a}_i$ is computed following the Least Square Error algorithm:

$$\boldsymbol{a}_i = (Z^T Z)^{-1} Z^T \boldsymbol{target'} \tag{20}$$

Then we consider another linear function in the form (14). Let us arrange the parameters of the model into a vector form

$$\boldsymbol{a}_i = \begin{bmatrix} a_{i0} \\ a_{i1} \\ \vdots \\ a_{in} \end{bmatrix} \tag{21}$$

Then the output of the rule-based model (14) is calculated by

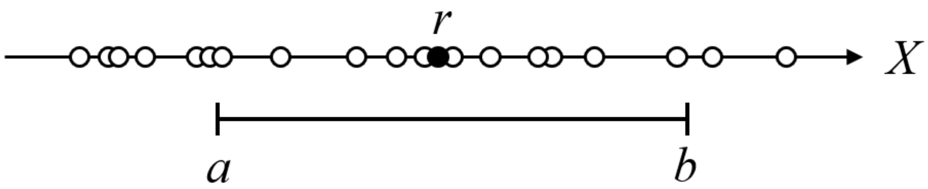
$$\hat{\boldsymbol{y}} = \sum_{i=1}^{c} A_i(\boldsymbol{x})\, \boldsymbol{a}_i^T \begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix} = \sum_{i=1}^{c} \boldsymbol{a}_i^T \begin{bmatrix} A_i(\boldsymbol{x}) \\ A_i(\boldsymbol{x})\boldsymbol{x} \end{bmatrix} \tag{22}$$

Similarity, we optimize the parameters $\boldsymbol{a}_i$, $i = 1,2, \ldots, c$, by minimizing the distance between data output and model output.

As mentioned above, Takagi-Sugeno model is more popular since it realizes the transformation from linguistic information to function. In the past few decades, it has been studied and applied in several aspects. A novel methodology for the extraction of a hierarchical Takagi-Sugeno fuzzy rule-based architecture from data was proposed in [50] to reduce the number and complexity of involved fuzzy rules. The proposed hierarchical architecture takes the form of a cascading topology in which the predicted result computed at the previous layer is considered in the output part of the fuzzy rules. Subsequently, an effective approach to data-based fuzzy modeling of high-dimensional systems was developed in [51], in which redundant rules are removed based on a fuzzy similarity measure and a genetic algorithm and the gradient method are introduced to optimize the structure and parameters of the model. In [52], a prediction model that combines interval type-2 TS fuzzy neural network model optimized by extended Kalman filter and SOM was proposed, which improves the prediction accuracy by 20% compared with traditional algorithms. Soares developed an explainable machine learning model based on the fuzzy rule-based model to significantly reduce the computation time in [53]. The main idea is to approximate the deep reinforcement learning (DRL) model with a set of if-then rules that provide an alternative interpretable model, which is further enhanced by visualizing the rules. The proposed approach includes a learning engine composed of zero-order fuzzy rules, which generalize locally around the prototypes by using multivariate function models. The adjacent prototypes, which correspond to the same action, are further grouped and merged into the so-called MegaClouds reducing significantly the number of fuzzy rules. Zhang designed a novel three-dimensional (3-D) fuzzy modeling framework without model reduction. It is a new 3-D fuzzy modeling method based on clustering and support vector regression. The advantages of this method are linguistic interpretability and no reliance on model reduction [54].

2.3 The principle of justifiable granularity

The terms of information granules and information granularity carry various meanings. One can refer to artificial intelligence in which case information granularity is central to a way of problem solving through problem decomposition where various subtasks could be formed and solved individually [55]. Zadeh coined an informal yet highly descriptive notion of an information granule [56]. In a general sense, by information granule, one regards a collection of elements drawn together by their closeness (resemblance, proximity, functionality, etc.) articulated in terms of some useful spatial, temporal, or functional relationships. The granules are formally described in various ways including fuzzy sets, intervals, rough sets, among others. They are constructed based upon the concept of generalised constraints, coverage and specificity, while 'coverage' quantifies the extent of the experimental data covered by the granule, and the 'specificity' means how specific the granule is. The optimal information granules are realized on higher values of coverage and specificity. Subsequently, Pedrycz and Homenda [57] proposed the principle of justifiable granularity, which intends to design a justifiable granule in presence of experimental evidence. The justifiable granule is formed in such a way that it has highly legality (justifiability) and specific enough meaning. Here we depict an illustrative example to offer a better insight into the essence of this model. Given is a collection of one-dimensional data $\boldsymbol{x} = \{x_1, x_2, \dots, x_N\}$. Our objective is to construct an interval information granule [a, b]. The optimization of the interval, namely the determination of its bounds (a and b) is split into two parts. First, we consider the numeric representative of data, denoted by r, which could be the median, mean, etc. Then the values of a and b are determined separately.



For the left part [a, r], we consider the coverage as the following expression

$$cov = \frac{1}{N}\sum_{k=1}^{N} \mathrm{ind}(x_k, [a, r]) \tag{23}$$

where *ind*() is an indicator function, defined as

$$\mathrm{ind}(c, [d, e]) = \begin{cases} 1, \text{ if } c \in [d, e] \\ 0, \text{ otherwise} \end{cases} \tag{24}$$

and the specificity is

$$sp = \frac{1}{N}\sum_{k=1}^{N} max\left(0{,}1 - \frac{|r - a|}{\mathrm{range}}\right) \tag{25}$$

where range is the range of $\boldsymbol{x}$, it could be written as $\left|max_{k=1,2,\dots,N}x_k - min_{k=1,2,\dots,N}x_k\right|$.

It is evident that with the increase of the interval, *cov* is an increasing function and *sp* is a decreasing function. For these two indexes, the higher their values, the better. However, they are of conflicting nature. Thus, to obtain a justifiable interval, the objective function can be realized as the product of the coverage and specificity.

The upper bound *b* is determined in the similar way. For a fixed r, we show the influence of *b* on the criteria cov, sp and their product in Figure 2.2. The experimental results further show that with the increase of upper bound *b*, the coverage and specificity are in conflict.

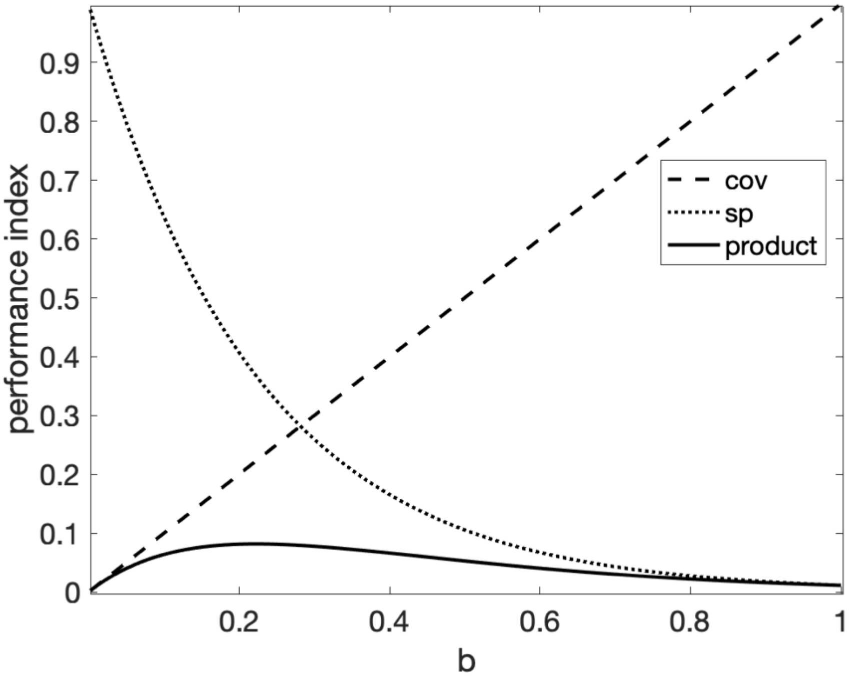


Figure 2.2 The performance indexes as a function of the upper bound.

2.4 Prediction interval

Linear regression model is a linear approach for modeling the relationship between dependent variables and one-dimensional or multi-dimensional independent variables. It is well-known that the regression model produces numeric results, which is usually augmented by the confidence interval or prediction interval. The confidence interval displays the $(1-\alpha)\%$ upper and lower confidence limits for the expected value of the predicted regression results, while the prediction interval [58]–[60] requests the $(1-\alpha)\%$ upper and lower confidence limits for an individual predicted regression value with α being the confidence level. Thus, with the increase of α, the intervals become narrow. To make it easy to follow, we show a regression problem with one-dimensional data, $(x_k, y_k), k = 1,2,\dots,N$. The linear regression model is defined as $\hat{y} = \beta x$, where β is a constant value estimated from the data. Denote the numeric result of the model by $\hat{y}$. Thus, the confidence interval is expressed by [61]

$$\hat{y} \pm t_{N-1}^{\alpha/2} \sigma_y \sqrt{\frac{1}{N} + \frac{(x-\bar{x})^2}{(N-1)\sigma_x^2}} \tag{26}$$

and the prediction interval is

$$\hat{y} \pm t_{N-1}^{\alpha/2} \sigma_y \sqrt{1 + \frac{1}{N} + \frac{(x-\bar{x})^2}{(N-1)\sigma_x^2}} \tag{27}$$

where σ_x and σ_y are the standard deviations of x and y, respectively, t denotes t-distribution and $\bar{x}$ is the mean value of the input. In Figure 2.3, the solid curve is the confidence interval, the dashed curve is the prediction interval, and the star describes the position of the mean value of *x*. It is easy to conclude that the prediction interval is wider than the confidence interval and both intervals become narrow when the corresponding x is close to $\bar{x}$.

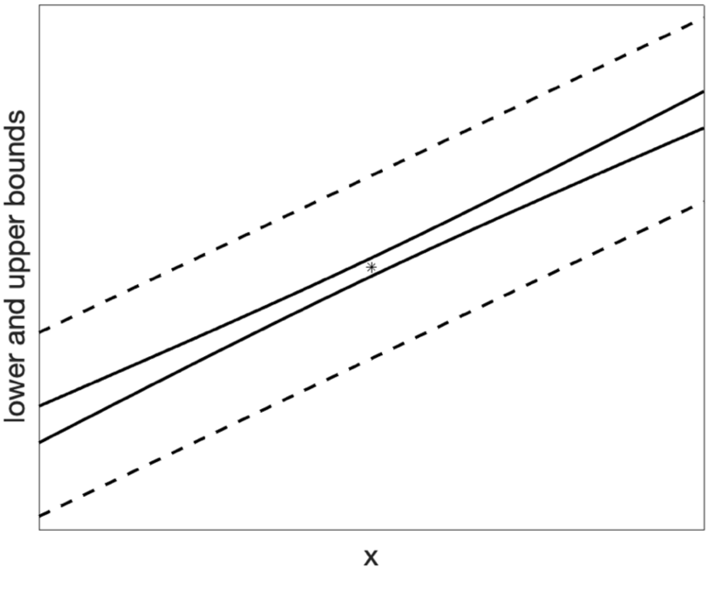


Figure 2.3 Confidence and prediction interval as a function of *x*. Solid curve: confidence bounds, dashed curve: prediction bounds.

2.5 Particle swarm optimization

Particle Swarm Optimization (PSO) was proposed by Eberhart and Kennedy in 1995 [62]. It is a stochastic optimization technique based on the movement of swarms, observing and imaging the effect of the information held by individuals on the population. Suppose that we have a swarm of particles of size *N* moving around in the domain. Each particle is characterized by its own position and velocity. A fitness function is used to describe the goodness of the position of the particle. There is an example in Figure 2.4, the PSO algorithm is introduced to find the maximum value of the function $y = e^{-\frac{(x_1-0.5)^2}{2}-\frac{(x_2-0.5)^2}{2}}$.

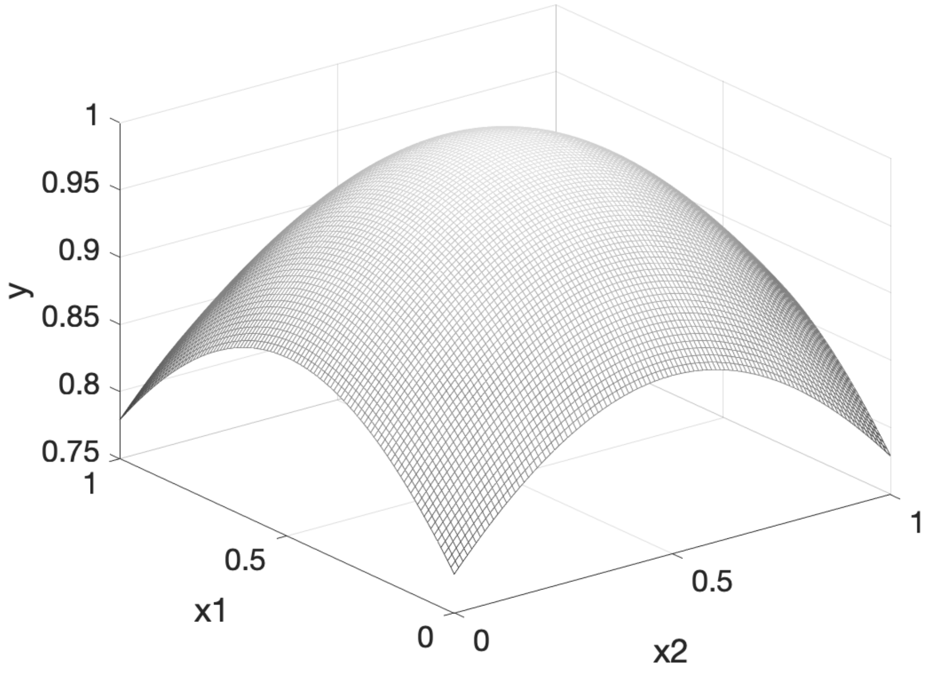


Figure 2.4 Fitness function.

The detailed process of the PSO algorithm is as follows. In the search space, for the i-th particle, its position and velocity are $\boldsymbol{po}_i$ and $\boldsymbol{ve}_i$, respectively. The particle comes with its personal best position visited so far (namely, $\boldsymbol{pbest}_i$) and the global best position within the entire swarm (namely, $\boldsymbol{gbest}$). Then the velocity and position are updated in the following form:

For updated velocity:

$$\boldsymbol{ve}_i^k = \lambda \boldsymbol{ve}_i^{k-1} + c_1 \boldsymbol{r}_1 \left(\boldsymbol{pbest}_i^{k-1} - \boldsymbol{po}_i^{k-1}\right) + c_2 \boldsymbol{r}_2 \left(\boldsymbol{gbest}^{k-1} - \boldsymbol{po}_i^{k-1}\right) \qquad (28)$$

where $i = 1,2, \dots, N$, the multiplication of vectors is completed coordinatewise.

The position is updated as follows:

$$\boldsymbol{po}_i^k = \boldsymbol{po}_i^{k-1} + \boldsymbol{ve}_i^{k-1} \qquad (29)$$

The position is clipped in the search space. For instance, in Figure 2.4, the range is [0,1]. In (28) (29), k stands for the index of iterations; i is the index number of the particle; λ is the non-negative inertia weight and c_1 and c_2 are the acceleration constants used for adjusting the learning rate; $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are the random vectors being generated by the uniform distribution over [0,1]. The updates of velocity and position as described by (28)-(29) are carried out until the global best position has been achieved.

The biggest advantage of the PSO lies in its simplicity and effectiveness. Compared with some traditional optimization algorithms, such as gradient descent, the PSO optimization is completed by searching the space without computing the gradients, which makes it easy to optimize complex problems, especially the non-differentiable or non-continuous problems. The Genetic algorithm (GA) is another optimization algorithm similar to PSO, both of which simulate the fitness of individual populations on the basis of natural characteristics. The GA algorithm is used to find the optimum solution with a series of evolution operations, such as crossover and mutation.

In addition, there are some studies on the improvement of the PSO algorithm. For example, the Elite Particle Swarm Optimization with Mutation (EPSOM) is proposed in [63], where the bad particles are replaced by the same number of elite particles, generating a new swarm. Another improved PSO is proposed by integrating the particle swarm optimizer, dynamic linkage discovery, and recombination operator, called particle swarm optimization with recombination and dynamic linkage discovery (PSO-RDL) [64]. Now the improved PSO algorithm has been commonly used to solve optimization problems [65]–[76]. The PSO-XGboost model is proposed to improve the classification accuracy of the model [77]. The PSO algorithm used in the density peaks clustering improves the global search ability of the classifier [78]. The combination of Particle Swarm Optimization and Support Vector Machine (PSO-SVM) is developed for damage identifications. The proposed approach is inspired by the effective searching capability of PSO, which can eliminate the redundant input parameters and a robust SVM technique is used to classify damage locations effectively [79].

2.6 Concentration effect

The curse of dimensionality was first proposed by Bellman [80], which is a general term associated with high-dimensional data. Typically, the curse of dimensionality includes concentration effects, combination explosions, etc. The concentration effect means that in a high-dimensional space, the difference in distance between data points tends to become smaller. To make it easy to follow, we elaborate upon an experiment with uniformly distributed random numbers in the range of [0,1] (with different dimensions). In the experiment, we compute the distances between any two data points, as shown in the following figure. The *x*-coordinate is the value of distance and *y*-coordinate shows the number of the data pairs generating the corresponding distance. It is obvious that with the increase of the dimensionality, the spread of distances between any two data points becomes concentrated, losing the diversity, resulting in high data similarity.

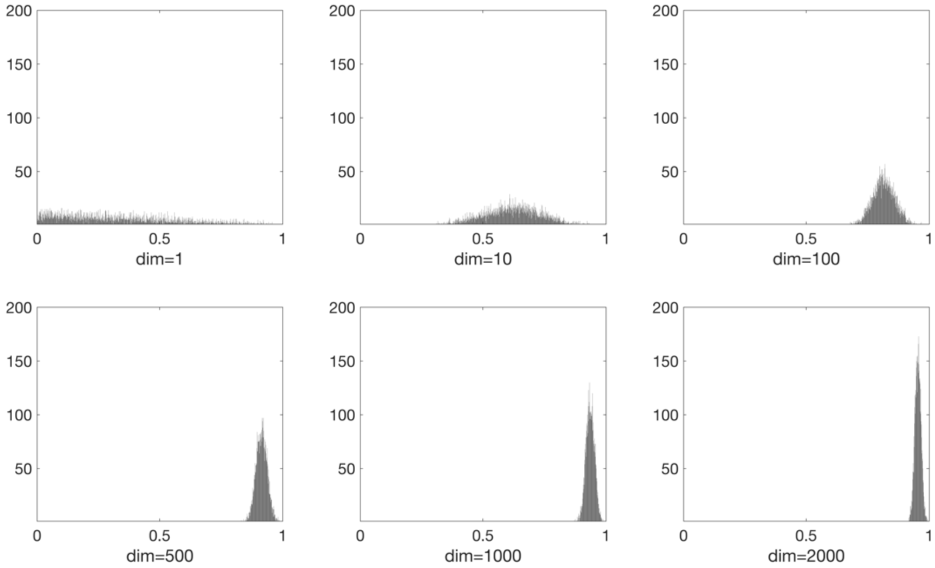


Figure 2.5 The distribution of distances for various number of variables.

2.7 Performance indexes

In view of the numeric results of the rule-based models, the evaluation criteria (Root Mean Squared Error) *RMSE* [81] is commonly used:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(target_k - \hat{y}_k)^2} \tag{30}$$

The other commonly used alternatives are Mean Squared Error (*MSE*) [82]:

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(target_k - \hat{y}_k)^2 \tag{31}$$

Both *RMSE* and *MSE* can intuitively show the error between the predicted value and the actual value. In addition, Mean Absolute Error (*MAE*) [83] is a measure of errors between paired observations expressing the same phenomenon

$$MAE = \frac{1}{N}\sum_{k=1}^{N}|target_k - \hat{y}_k| \tag{32}$$

and R Squared R^2 [84] is a statistical measure that represents the proportion of the variance for a dependent variable, that is explained by an independent variable or variables in a regression model.

$$R^2 = 1 - \frac{MSE(target, \hat{y})}{var(target)} \tag{33}$$

The evaluation criteria are close to each other, exhibiting no essential difference. Thus we just use RMSE as the criterion to measure the difference between the original output and numeric results.

2.8 Conclusions

In this section, fuzzy rule-based models and clustering are introduced which are the basic models and algorithms of this research. Besides, the concentration effect is described. Subsequently, the concept of granularity is used for granulating the model for evaluation, and an advanced decision-making model based on granulation is developed. Then we briefly introduce the prediction interval. Intervals are used to enhance the performance of our models. Finally, the basic principles and development of particle swarm optimization (PSO) are explained and the performance indexes are introduced to optimize and evaluate the model.

Chapter 3. Enhancement of rule-based models—designing distributed models

When it comes to the large or high-dimensional data, the problem concentration effect arises, see Section 2.6. The distributed rule-based models can avoid the appearance of them, since for the distributed low-dimensional data, a clustering algorithm works well. The term distributed fuzzy rule-based models can be found in the literature [85]–[88]. Especially the work in [86] proposes a distributed fuzzy decision tree learning scheme to generate binary and multiway fuzzy decision trees from dataset following the MapReduce programming model. In this study, the proposed distributed rule-based model architecture is that the combination of membership grades as input is transformed to output space through the optimization of the linkage matrix or both a linkage matrix and prototypes of the output part. Also, the development of the model is completed in a supervised mode in the presence of input-output pairs of data $(\boldsymbol{x}_k, target_k)$ mentioned as before. The performance index Q expressing the quality of the model is shown in Section 2.7.

3.1 Distributed rule-based model and its development

In anticipation of dealing with a large number of input variables, our intent is to develop a single input rule-based model associated with a minimal design effort, implying a minimal computing overhead. Such models (more specifically, their condition parts) will be used as the components of the distributed fuzzy model.

We consider *c* rules with the simplest form of the conclusion part (0th order TS model), see equation (10). The design of the rules (viz. their condition and conclusion parts) can be realized in various ways depending upon the involvement of the optimization mechanisms. A concise summary of the existing alternatives is presented in Table 3.1.

Table 3.1 Development strategies of one-dimensional rule-based model.

Condition/conclusion	Conclusion (no optimization)	Conclusion (optimization)
Condition (no optimization)	uniformly distributed membership functions in the input space and uniformly distributed numeric representatives in the output space	uniformly distributed membership functions in the input space and optimization of numeric representatives in the output space (e.g., through clustering)
Condition (optimization)	optimized membership functions in the input space (e.g., obtained through clustering) and uniformly distributed numeric representatives in the output space	Combined optimization of the membership functions and numeric representatives in the output space (e.g., through fuzzy clustering applied to data in both spaces)

In the first variant identified in Table 3.1, no optimization schemes have been engaged. The fuzzy sets formed in the input space have triangular membership functions uniformly distributed across the input space. The conclusion part is formed as taking the weighted outputs that fall within the realm of the corresponding fuzzy set in the input space. In other words, the constant result is a context-dependent (conditional) average of y_k computed within the context expressed by A_i. It is worth noting that in this way of building the rules, from the input-output perspective, the fuzzy rule-based model realizes a piecewise linear mapping. One may anticipate that with the increase in the number of rules, this approximation produces better results in terms of smaller values of Q. The number of rules c can be selected by observing the change of Q regarded as a function of c: we determine such a number c_{opt} when a further increase in the number of rules does not result in significant improvements of the model. In other words, Q decreases with the increase of c, but once c becomes bigger than c_{opt}, Q decreases slightly, e.g. $Q(c_{opt}+1)/Q(c_{opt}) \rightarrow 1$. Using $Q(c_{opt}+1)/Q(c_{opt})$ with fixed τ such that we can obtain a more precise value of c_{opt} than plotting Q as a function of c. Once c_{opt} has been determined, further calculations for the higher number of clusters are not considered, which contributes to a reduced computing overhead. In a quantitative way, we consider the ratio $Q(c+1)/Q(c)$, trace its values over successive values of c and determine the smallest c for which the following relationship holds

$$Q(c+1)/Q(c) < \tau, \tau < 1 \qquad (34)$$

where τ is a certain threshold whose value has been predefined. When it comes to the optimization alternatives outlined in Table 3.1, the refinements of the membership functions in terms of their shapes and parameters are considered. They typically engage the clustering of input and output data. It is likely that such models could (will) result in better performance, however some computing overhead is encountered. As usual, one needs a critical assessment of the gains in the performance and the associated optimization effort.

We also consider two-input distributed rule-based models [89], [90], where the rules are constructed in the same way as their single-input computation process. Note that now the number of rules is c^2 in comparison with c rules for single input models.

3.2 From local models to a global model

We are concerned with a distributed collection of data in the sense that individual data are described in the individual feature spaces $\boldsymbol{F}_1$, $\boldsymbol{F}_2$, …, $\boldsymbol{F}_r$. These feature spaces can be disjoint or overlap. In particular, the spaces could be one-dimensional, i.e., card ($\boldsymbol{F}_i$) =1. The buildup of such one-dimensional models is motivated by some experimental findings reported in [91]. The architecture is composed of several main functional modules as outlined in Figure 3.1.

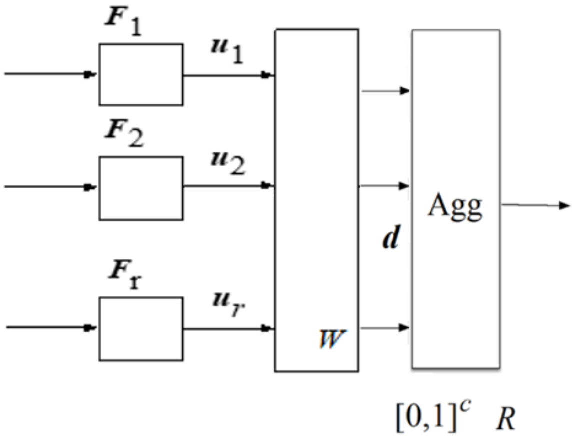


Figure 3.1 Overall architecture of distributed rule-based model.

A. Construction of combined vector of membership grades

The condition parts (fuzzy sets) defined in one-dimensional input spaces transform any scalar input into a vector of membership grades; this is done individually for each variable. The number of fuzzy sets in these spaces is determined following the procedure described in Section 3.1. Thus, we obtain vectors of membership grades of dimensionality of $c_1, c_2, \ldots, c_r$ respectively. Denote these vectors as $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r$. Let $c' = \sum_{i=1}^{r} c_i$. We concatenate these vectors into a single c'-dimensional vector $\boldsymbol{u}$, i.e., $\boldsymbol{u} = [\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \cdots \quad \boldsymbol{u}_i \quad \cdots \quad \boldsymbol{u}_r]^T$. For two-input rule-based models, we obtain the vectors of membership grades $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r$ from the corresponding feature space with the dimensionality of ${c_1}^2, {c_2}^2, \ldots, {c_r}^2$ respectively; thus we have $c' = \sum_{i=1}^{r} {c_i}^2$.

Transformation of membership grades through a linkage matrix to the output space. Before proceeding with the detailed formulas, we introduce a concise matrix notation,

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1c'} \\ w_{21} & w_{22} & \cdots & w_{2c'} \\ \vdots & \vdots & \ddots & \vdots \\ w_{c1} & w_{c2} & \cdots & w_{cc'} \end{bmatrix} U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ u_{21} & u_{22} & \cdots & u_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{c'1} & u_{c'2} & \cdots & u_{c'N} \end{bmatrix} \tag{35}$$

Then the mapping from the c'–dimensional vector of membership degrees positioned in the input space to the output space is completed in the following standard manner

$$\hat{y}_k = \sum_{i=1}^{c} d_{ik} \bar{y}_i \tag{36}$$

where $\bar{y}_i$ is the numeric representative (see Section 3.1) in the output space; d_{ik} is computed with the aid of the linkage (association) matrix W as follows:

$$d_{ik} = \sum_{j=1}^{c'} u_{jk} w_{ij} \tag{37}$$

where u_{jk} and w_{ij}, $i = 1,2, \ldots, c$, are the corresponding elements in (35). In a matrix notation, one has

$$d = Wu \tag{38}$$

Thus, ***d*** becomes a linear combination of ***u*** and some unknown W, where ***u*** stands for the corresponding column of matrix *U*.

3.3 Design of the distributed rule-based model

In the process of designing the model, we are concerned with the determination of the linkage matrix *W* and the collection of prototypes in the output space. We assume that the prototypes in the individual input space are formed as discussed in Section 3.1.

A. Optimization of the linkage matrix

The prototypes (representatives) located in the output space are formed by being uniformly distributed across the output space. Here we envision building c-2 clusters. We also add the extreme values encountered in the output space; in total this yields *c* prototypes; denote them as $\bar{y}_{1,}\bar{y}_2,\ldots,\bar{y}_c$, where $\bar{y}_1$ is the minimal value of the output data and $\bar{y}_c$ is the maximal value of the output data. $\bar{y}_2,\ldots,\bar{y}_{c-1}$ are distributed in a uniform way.

The objective is to minimize the sum of distances between $target_k$ and $\hat{y}_k$, $k = 1,2,\ldots,N$, with respect to the unknown matrix of linkages (associations) W, namely

$$Q(W) = \sum_{k=1}^{N}(target_k - \hat{y}_k)^2 \qquad (39)$$

and

$$W_{opt} = arg(\,min_W\,Q\,(W)) \qquad (40)$$

Let us proceed with the detailed estimation process. The original output data, output of the model and the numeric prototypes are arranged in the following vector format

$$\boldsymbol{target} = \begin{bmatrix} target_1 \\ target_2 \\ \vdots \\ target_N \end{bmatrix}, \boldsymbol{\hat{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{bmatrix}, \boldsymbol{\bar{y}} = \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \bar{y}_c \end{bmatrix} \tag{41}$$

Furthermore, using the matrix notation (35), one has $\hat{y} = (WU)^T\bar{y}$. Next, we rewrite the optimization problem (39) in the following concise format

$$min_W\|\boldsymbol{target} - \boldsymbol{\hat{y}}\|^2 = min_W(\boldsymbol{target} - (WU)^T\boldsymbol{\bar{y}})^T(\boldsymbol{target} - (WU)^T\boldsymbol{\bar{y}}) \tag{42}$$

Proceeding with the detailed calculations, one has

$$\begin{aligned} \|\boldsymbol{target} - \boldsymbol{\hat{y}}\|^2 = \boldsymbol{y}^T\boldsymbol{target} + (\boldsymbol{\bar{y}}^T WUU^T W^T\boldsymbol{\bar{y}}) - (\boldsymbol{\bar{y}}^T WU)\boldsymbol{target} \\ -\boldsymbol{target}^T(U^T W^T\boldsymbol{\bar{y}}) \end{aligned} \tag{43}$$

Taking the gradient of (43), we have the following expression

$$\nabla_W\|\boldsymbol{target} - \boldsymbol{\hat{y}}\|^2 = 2\boldsymbol{\bar{y}}\boldsymbol{\bar{y}}^T WUU^T - 2\boldsymbol{\bar{y}}\boldsymbol{target}^T U^T = 0 \tag{44}$$

Finally, the analytical solution to (44) comes in the form

$$\begin{aligned} 2\boldsymbol{\bar{y}}\boldsymbol{\bar{y}}^T W_{opt}UU^T = 2\boldsymbol{\bar{y}}\boldsymbol{target}^T U^T \\ W_{opt} = (\boldsymbol{\bar{y}}\boldsymbol{\bar{y}}^T)^{-1}\boldsymbol{\bar{y}}\boldsymbol{target}^T U^T(UU^T)^{-1} \end{aligned} \tag{45}$$

B. Optimization of the linkage matrix and prototypes in the output space

To design the distributed model, we also consider another alternative, where both the linkage matrix *W* and the vector of prototypes $\bar{y}$ are optimized, namely *Q* is a function of *W* and $\bar{y}$; the minimum of *Q* with respect to $\bar{y}$ is expressed as

$$min_{\bar{y}}\|\boldsymbol{target} - \boldsymbol{\hat{y}}\|^2 = min_{\bar{y}}(\boldsymbol{target} - (WU)^T\boldsymbol{\bar{y}})^T(\boldsymbol{target} - (WU)^T\boldsymbol{\bar{y}}) \tag{46}$$

where *W* and *U* are defined in (35). By using the notation $A = WU$, we have

$$\begin{aligned} (\boldsymbol{target} - (WU)^T\boldsymbol{\bar{y}})^T(\boldsymbol{target} - (WU)^T\boldsymbol{\bar{y}}) = (\boldsymbol{target} - A^T\boldsymbol{\bar{y}})^T(\boldsymbol{target} - A^T\boldsymbol{\bar{y}}) \\ = \boldsymbol{target}^T\boldsymbol{target} + (\boldsymbol{\bar{y}}^T AA^T\boldsymbol{\bar{y}}) - (\boldsymbol{\bar{y}}^T A)\boldsymbol{target} - \boldsymbol{target}^T(A^T\boldsymbol{\bar{y}}) \end{aligned} \tag{47}$$

The gradient of (47) is

$$\nabla_{\bar{y}}(y - A^T\bar{\boldsymbol{y}})^T(\boldsymbol{target} - A^T\bar{\boldsymbol{y}}) = 2AA^T\bar{\boldsymbol{y}} - 2A\boldsymbol{target} = 0 \tag{48}$$

Proceeding with details, one has

$$\begin{gathered} AA^T\bar{\boldsymbol{y}}_{opt} = A\boldsymbol{target} \\ \bar{\boldsymbol{y}}_{opt} = (AA^T)^{-1}A\boldsymbol{target} \\ \bar{\boldsymbol{y}}_{opt} = (WU(WU)^T)^{-1}(WU)\boldsymbol{target} \end{gathered} \tag{49}$$

As W and $\bar{y}$ are subject to the optimization of $Q(W,\bar{y})$, the formulas (45) and (49) are used iteratively by proceeding with some initial condition, considering a uniformly distributed output prototypes, and successively updating the values of W and $\bar{y}$.

In what follows, we present a general process to build the construction of the distributed models.

```
1. for variable (feature)= 1 to r (r is the number of variables)
   {
     for cluster (c)= the range of clusters
     {
        Generate c prototypes uniformly distributed, calculate the corresponding
        triangular membership grades $\boldsymbol{u}$, and calculate the corresponding $\bar{y}$ so that
        $Q(c) = \|u^T\bar{y} - target\|$ is obtained.
     }
     Using the ratio $Q(c+1)/Q(c) < \tau$ to determine optimal c and the
     corresponding $\boldsymbol{u}$ and $\bar{y}$.
   }
2. Based on step 1, we obtain $\boldsymbol{u}$ and c for each variable; then we combine all $\boldsymbol{u}$ into U
   and generate the prototypes ($\bar{y}$) of output space uniformly distributed across the
   output space where the number of prototypes we set to the mean of all c, through
   formula (41) to determine the linkage matrix W.
3. $Q = \|U^TW^T\bar{y} - target\|$.
```

C. Computing overhead

The analysis of computing complexity encounters two main calculation procedures, namely fuzzy clustering and matrix inversions. Two extreme situations (a monolithic model and a family of one-dimensional rule-based models) are investigated. The time complexity predominantly associated with the fuzzy clustering [92] is $O(c^2NnI)$, where I is the number of iterations of the clustering method (in the case of the monolithic fuzzy model) and for matrix inversion, the associated complexity is $O(c'^3) + O(c^3)$ [26], where usually $c' \gg c$ holds in the case of distributed rule-based models.

3.4 Experimental studies

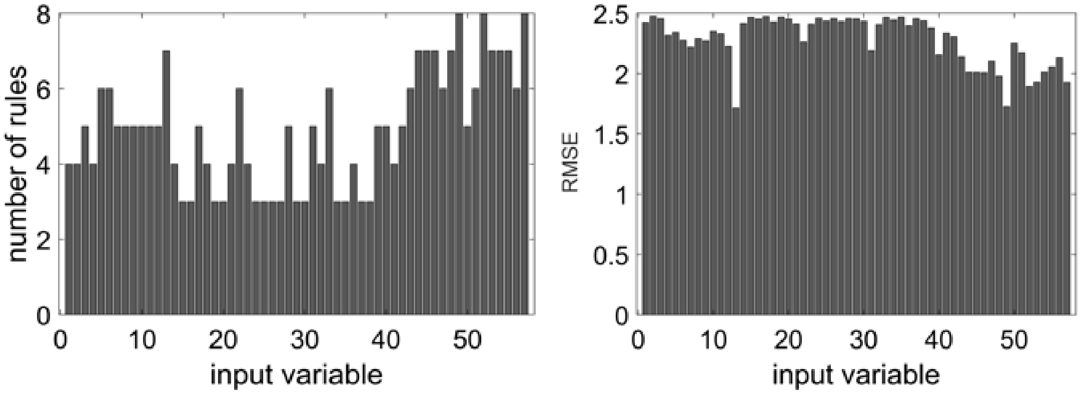
The experimental studies reported in this section are concerned with the publicly available data from https://archive.ics.uci.edu/ml/datasets.php, https://datahub.io/m-achine-learning, https://sci2s.ugr.es/keel/semisupervised.php and https://www.kaggle.com/. The objective is to quantify the performance of the model and deliver some comparative analysis involving a monolithic multivariable fuzzy model by contrasting the performance expressed in terms of the corresponding *RMSE* values and the development time. The experiments were carried out on a PC with AMD Ryzen Threadripper 2990WX 32-Cores 64-Threads 3.5GHz CPU and 64GB RAM running the MATLAB 2019a in the same environment. The experiment is repeated 30 times provided that the data are split into the training and testing sets (70-30% split), and both the average values of the performance index and their associated standard deviations are reported.

The monolithic approach is based on the *FCM* algorithm which is used in its generic version with the weighted Euclidean distance (including the variance of the variables as the weights) while the fuzzification coefficient is set to 2.0. The concatenated input-output variables are clustered. The algorithm runs for 1,000 iterations; it has been found that this number of iterations is sufficient for the method to converge. Besides, in the monolithic fuzzy rule-based model (the same structure shown in (9)), the condition part A_i comes from partition matrix in the FCM algorithm and L_i comes from the prototype $\bar{y}_i$ in the output space produced by the FCM algorithm. For the one-dimensional models with the optimal matrix W, we determine the number of rules based on (34) with the value of τ set to 0.95. When the one-dimensional models contain two parameters (the matrix W and vector $\bar{\boldsymbol{y}}$), we use iterate formula (45) and (49) 100 times.

Crop Data Challenge 2018 (Maize) (no. #3394, 58 including 57 input variables) (https://www.kaggle.com/).

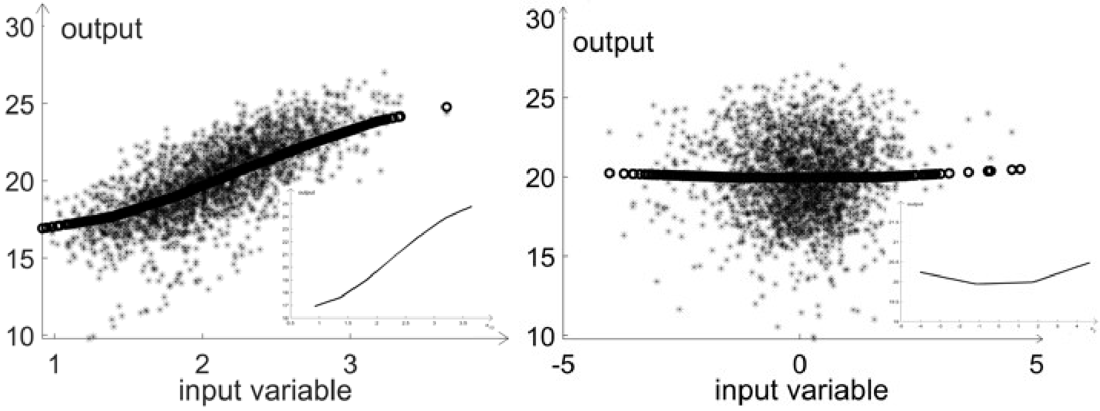
First, we focus on the experiment involving one-dimensional rule-based models. To build a collection of 57 single input models, we determine the number of rules and the obtained results are displayed in Figure 3.2. The constants in the corresponding conclusions are generated by (10). The corresponding values of the performance index (*RMSE*) are included in Figure 3.3(*b*). There is a quite visible difference between the number of rules for each model ranging from three to eight. The average number of rules is five. As the triangular fuzzy sets are used, the input-output characteristics are piecewise linear. The plots of them with superimposed data for the best single input model (the 13th variable) and the worst single input model (the 2nd variable) are included in Figure 3.3, where the piecewise linear nature of the mapping is visible.

As a matter of curiosity, we also design the single-input rules by invoking the optimization mechanism of the *FCM* clustering with the fuzzification coefficient set to two. The input-output data are clustered, yielding pairs of prototypes in the input and output space, respectively. The constant conclusions are the prototypes generated by the clustering algorithm. Consider two variables: the 13th one and the 2nd one; the corresponding *RMSE* values are displayed in Figures 3.4 (*a*) and (*b*).



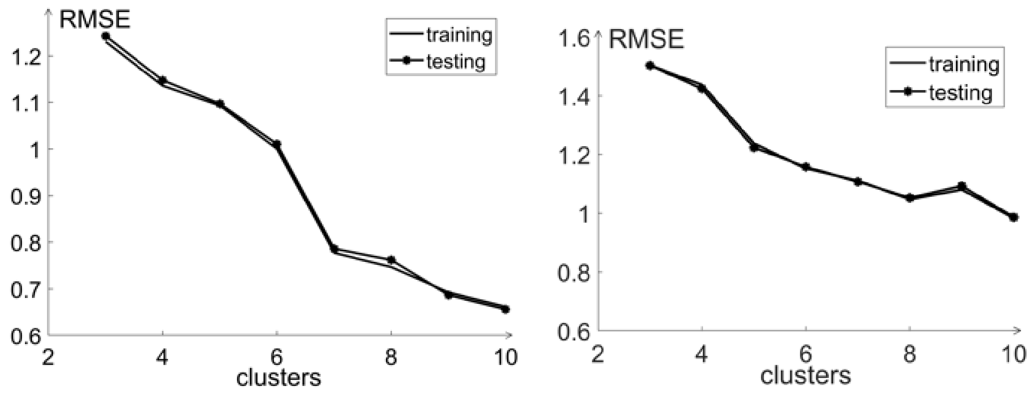
(*a*) (*b*)

Figure 3.2 Average performance index for single input rule-based models: (*a*) optimal number of rules (*b*) corresponding values of *RMSE* index.



(*a*) (*b*)

Figure 3.3 Input-output plots of one-dimensional models with superimposed data, where the stars are the original points and the piecewise linear is described in the right bottom part: (*a*) the 13th variable (*b*) the 2nd variable.



(*a*) (*b*)

Figure 3.4 *RMSE* values versus c for the rule-based model obtained for the (a) the 13th variable and (b) the 2nd variable.

Realizing the distributed models (with the value of $c = 3,4,\ldots,8$ for one-dimensional model and $c = 3$ for two-input model), the obtained results for *Crop Data Challenge* 2018 dataset are covered in Table 3.2. Both the *RMSE* values and the computing times are presented. The monolithic model is run for several selected values of *c* such as the minimal, maximal and average ones.

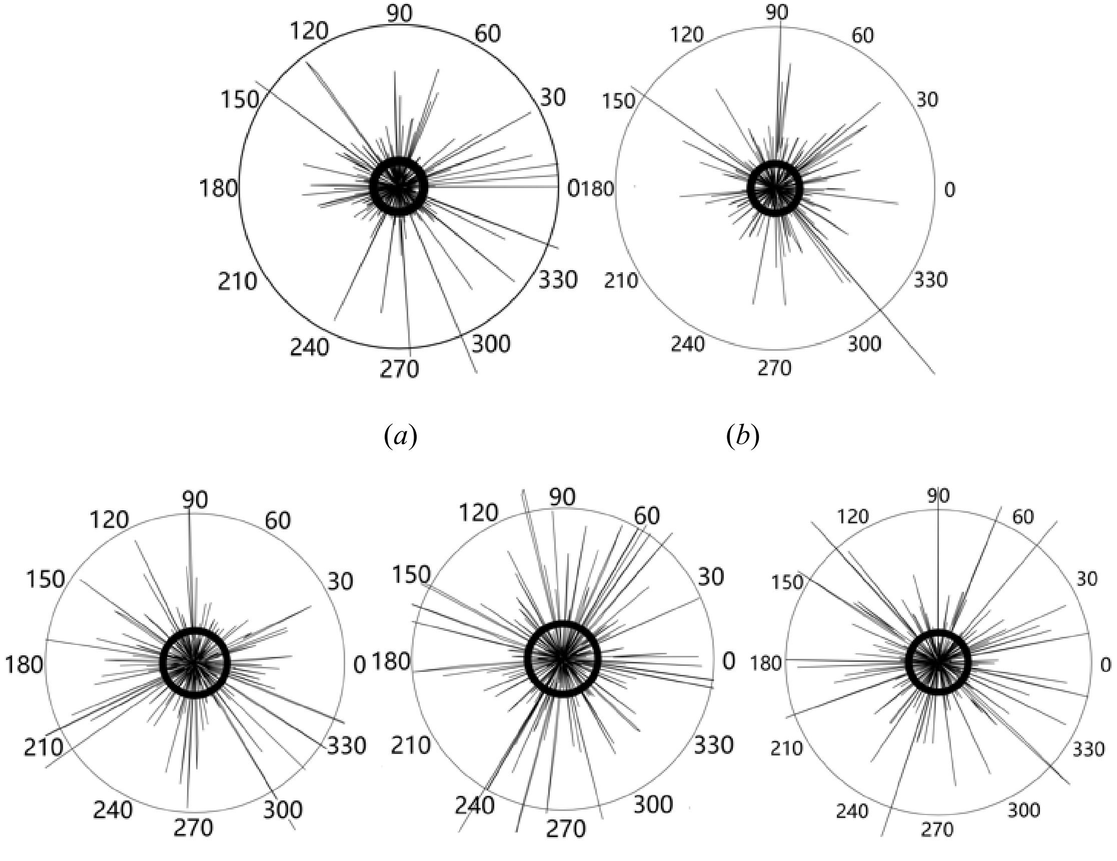
Table 3.2 *RMSE* obtained for the distributed rule-based model and the monolithic rule-based model involving selected numbers of their rules.

				Monolithic model		
	Distributed model 1*	Distributed model 2*	Two input feature space	$min(c_r)$ =3	mean(c_r) =5	$max(c_r)$ =8
Q_{train}	0.4639± 0.0056	0.4667± 0.0055	0.6038± 0.0105	2.2932± 0.4905	2.2310± 0.2038	2.0484± 0.2737
Q_{test}	0.6313± 0.0390	0.6159± 0.0285	0.9026± 0.0453	2.3041± 0.5114	2.2314± 0.2053	2.0613± 0.2791
T(sec)	0.0423s	3.9345s	0.0660s	0.6070s	6.6957s	10.7159s

* Distributed model 1 is one-dimensional rule-based model with optimal linkage matrix *W*.

* Distributed model 2 is one-dimensional rule-based model with optimal linkage matrix W and prototypes $\bar{y}_i$ of output space.

To contrast the results for the individual data, the values of squared differences are displayed in the form of the radar plot, as shown in Figure 3.5 (as the number of data is large, here we only show the first 360 data points).



(*a*) (*b*)

(*c*) (*d*) (*e*)

Figure 3.5 Radar plot of errors shown for the individual data (testing set); the circle is the average value of error. (*a*) distributed one-dimensional model (*b*) two-input rule-based model; monolithic model: (*c*) *c*=3, (*d*) *c*=5, and (*e*) *c*=8.

The experiments are completed for the datasets shown in Table 3.3.

Table 3.3 Dataset information.

Dataset	number of data	dimensionality
Wine Quality	1,599	13
Electrical Grid stability simulated	10,000	14
Appliances energy prediction	19,735	26
Concrete Compressive Strength	1,030	9
Satimage	6,435	36
Texture	5,500	40

(*a*), (*b*), (*c*), (*d*) come from (https://archive.ics.uci.edu/ml/datasets.php), while (*e*) comes from (https://datahub.io/machine-learning), and (*f*) is from (https://sci2s.ugr.es/keel/semisupervised.php). *RMSE* values and resulting computing overhead are shown Table 3.4. Based on the experimental results produced for one-dimensional rule-based models, one can conclude that the distributed model performs better than the monolithic model in terms of the values of the performance index and the associated computing overhead. When it comes to single-input or two-input distributed models, the corresponding values *Q* are close to each other, however, the performance of a single-input distributed model is the better one for all datasets except for the Electrical grid stability dataset.

Table 3.4 *RMSE* for the distributed and monolithic rule-based model involving selected numbers of rules.

	Distributed model 1*	Distributed model 2*	Two input feature space (*c*=3)	Monolithic model		
				min(c_r) =3	mean(c_r) =5	max(c_r) =8
Q_{train}	0.6117± 0.0111	0.6114± 0.0077	0.6337± 0.0096	0.9300± 0.1343	0.9329± 0.1718	0.9965± 0.1699
Q_{test}	0.7832± 0.4430	0.7250± 0.1540	0.8902± 0.0102	0.9313± 0.1316	0.9364± 0.1865	1.0022± 0.1797
T(sec)	0.0050s	0.3203s	0.0112s	0.0776s	0.1215s	0.2976s

(*a*)

	Distributed model 1*	Distributed model 2*	Two input feature space (*c*=3)	Monolithic model		
				min(c_r) =3	mean(c_r) =4	max(c_r) =9
Q_{train}	0.0171± 1.147e-4	0.0172± 9.9405e-5	0.0167± 6.8289e-5	0.0379± 7.1544e-4	0.0375± 6.0172e-4	0.0374± 3.4394e-4
Q_{test}	0.0173± 2.159e-4	0.0172± 2.1961e-4	0.0169± 1.6159e-4	0.0382± 6.5696e-4	0.0376± 5.9866e-4	0.0375± 5.0096e-4
T(sec)	0.0081s	0.8020s	0.0112s	3.8997s	8.6779s	14.7697s

(*b*)

	Distributed model 1*	Distributed model 2*	Two input feature space (c=3)	Monolithic model		
				min(c_r) =3	mean(c_r) =7	max(c_r) =9
Qtrain	14.4411± 0.0304	14.4439± 0.0324	14.4443± 0.0309	17.9374± 0.4089	17.3980± 0.3773	16.9975± 0.4646
Qtest	14.5580± 0.0729	14.5604± 0.0697	14.6126± 0.0704	17.9375± 0.3882	17.4226± 0.3813	16.9963± 0.4405
T(sec)	0.0558s	7.4562s	0.0822s	2.3861s	6.5639s	29.0864s

(*c*)

	Distributed model 1*	Distributed model 2*	Two input feature space (c=3)	Monolithic model		
				min(c_r) =3	mean(c_r) =6	max(c_r) =8
Q_{train}	6.1375± 0.1630	6.0964± 0.1515	7.9323± 0.1313	19.4042± 2.0824	19.6644± 3.2476	18.6308± 2.0706
Q_{test}	11.4269± 1.5050	10.4462± 1.3412	12.0546± 1.1059	19.5849± 2.3595	19.7241± 3.5755	18.7089± 1.9708
T(sec)	0.0021s	0.1454s	0.0023s	0.0685s	0.1876s	0.2586s

(*d*)

	Distributed model 1*	Distributed model 2*	Two input feature space (c=3)	Monolithic model		
				min(c_r) =3	mean(c_r) =7	max(c_r) =9
Q_{train}	19.1414± 0.5748	18.0270± 0.9805	19.2766± 0.4256	33.2661± 11.0861	36.1910± 6.4121	36.3031± 6.7764
Q_{test}	19.5449± 1.3399	18.5377± 1.0204	20.6933± 1.4696	33.2979± 11.0444	36.5775± 6.1629	36.5870± 7.1875
T(sec)	0.0420s	4.7678s	0.0373s	0.8106s	6.1633s	25.0040s

(*e*)

	Distributed model 1*	Distributed model 2*	Two input feature space (c=3)	Monolithic model		
				min(c_r) =6	mean(c_r) =8	max(c_r) =9
Q_{train}	0.0123± 1.2783e-4	0.0123± 1.2463e-4	0.0134± 0.4256	0.5830± 0.0303	0.5801± 0.0271	0.5810± 0.0301
Q_{test}	0.2489± 0.6031	0.1645± 0.4877	0.3066± 1.4696	0.5841± 0.0318	0.5802± 0.0277	0.5824± 0.0305
T(sec)	0.0536s	5.8437s	0.0530s	3.5084s	9.7296s	12.4635s

(*f*)

* Distributed model 1 is one-dimensional rule-based model with optimal linkage matrix W.

* Distributed model 2 is one-dimensional rule-based model with optimal linkage matrix W and prototypes $\bar{y}_i$of output space.

Table 3.5 Improvement (*RMSE*) of distributed one-dimensional rule-based model 1 (compared to entire feature space) for testing data.

dataset	% improvement			
		min(c)	average(c)	max(c)
Crop Data Challenge 2018 (*Maize*)	Q_{test}	72.60%	71.71%	69.37%
	Time	93.03%	99.37%	99.61%
Wine Quality	Q_{test}	15.90%	16.36%	21.85%
	Time	93.56%	95.88%	98.32%
Electrical Grid stability simulated	Q_{test}	54.71%	53.99%	53.87%
	Time	99.79%	99.91%	99.95%
Appliances energy prediction	Q_{test}	18.84%	16.44%	14.35%
	Time	97.66%	99.15%	99.81%
Concrete Compressive Strength	Q_{test}	41.65%	42.07%	38.92%
	Time	96.93%	98.88%	99.19%
Satimage	Q_{test}	41.30%	46.57%	46.58%
	Time	94.82%	99.32%	99.83%
Texture	Q_{test}	57.39%	57.10%	57.26%
	Time	98.47%	99.45%	99.57%

We also do the experiments for Condition monitoring of hydraulic systems CE Dataset. Compared with the monolithic model with average c, the distributed model improves 63.3% for testing results and 96.2% of computing overhead.

Higher-dimensional models (say, those involving pairs of input variables) are constructed in an analogous manner as in the case of single input models. One has become of the fact of the rapid increase of the number of rules. For instance, for a two-dimensional input space and c_i fuzzy sets distributed across the individual variables, one encounters c_i^2 rules in comparison with c_i rules present in the one-dimensional model. This gives rise to the increasing dimensionality of the combined vector of membership degrees used in the construction of the linkage matrix W.

In summary, the distributed rule-based models produce visible improvements for all datasets, especially the improvement expressed in terms of the resulting computing overhead. For most datasets, the distributed rule-based model with the optimal linkage matrix and prototypes in the output space shows improvement over the performance achieved when only the linkage matrix has been optimized.

As for the monolithic models, there is an optimal value of the number of clusters for each dataset. However, when we compare the result of the monolithic model performed with the optimal cluster to that of the distributed models, the former model is worse by 15.90%. Distributed models exhibit good performance for all datasets in the experiments, especially for the *Crop Data Challenge 2018 data* which contains 57 inputs. This particular data demonstrates that a monolithic model is not suitable for high-dimensional data.

Then we show the comparison between the FDSS algorithm in [93] and the distributed models in our paper, where data we use are Wine and WDBC.

Table 3.6 Accuracy (%) obtained for the distributed rule-based model and FDSS.

	Distributed model 1*	FDSS
Wine	56.54±0.08	55.82±3.17
WDBC	98.97±0.03	92.09±6.7

Through comparing the accuracy for these two methods, it is transparent that the distributed model performs better, especially for *WDBC* data; our distributed model improves the performance by 7.5%.

In addition, we also added the experiment of distributed model based on PSO algorithm, that is, we do the experiments as before. For each feature space, when a fuzzy rule-based model is used, we need to generate the prototypes and then calculate the membership grades for the condition part. In our paper, we mentioned that the prototypes are generated by uniformly distribution, but for the distributed model with PSO algorithm. The difference is that prototypes are obtained by the PSO algorithm by minimizing the objective function (error between original and model output). The performance index and the computing overhead are computed for some datasets, where the resulting performance (the average result values of all feature spaces) is a function of the number of clusters. Compared with the distributed model, for *Crop Challenge Data*, the distributed model designed with the PSO algorithm gives the largest improvement of the performance index at the level of 0.86%. However, the computing overhead increases 366 times. As for the Wine Quality data, the performance index improves by 2.82% and the running time has the 262-fold increased overhead.

From the above comparisons, we conclude that once the PSO algorithm has been added, no matter what the number of the cluster is selected, the resulting *RMSE* is slightly better than the one obtained for the distributed model. However, the computing overhead is substantially increased.

3.5 Conclusions

This study has focused on the development of distributed fuzzy rule-based models. The presence of high-dimensional data calls for a change in the design strategy by moving from the monolithic nature of the model to a distributed form of the architecture composed of a number of models of low dimensionality. In particular, the distributed architecture could be composed of one-dimensional fuzzy sets followed by the aggregation mechanism. The conclusions drawn from the series of experiments point out the advantages of the distributed format of the models of their monolithic character, which are manifested through lower *RMSE* values and lower computing overhead. Based on the experimental evidence gained in this study, compared with a monolithic model, the one-dimensional rule-based model shows improvements: especially in distributed model 1.

The distributed architecture of the model provides a sound starting position for further investigations. The experiments presented here concentrate on the one or two-dimensional space of conditions. However, it could be interesting to study the situations of low dimensional feature spaces whose content in the sense of variables involved there has become optimized. The linkage matrix and the way of transformation carried out from the input membership space to the output space studied here form a simple linear mapping; other alternatives including logic-oriented and nonlinear mappings are worth investigating.

Chapter 4. Augmentation of rule-based models—with a granular quantification of results

As we mentioned above, the rule-based model has a weak point in terms of the numeric output space. For the sake of completeness of our considerations, we introduce prediction intervals formed for each rule present in the rule-based models. Their aggregation follows the aggregation rule applied to the original model.

Confidence intervals and prediction intervals are the tangible and compelling manifestation and full acknowledgement of the limited quality of models. There are no ideal models. The likelihood that a numeric result of prediction coincides with the real data is zero. This has been the driving force behind the concept of prediction intervals as a vehicle to make the prediction outcomes more aligned with real-world phenomena. One has to acknowledge that the result of prediction is an information granule. The commonly employed formalism uses intervals. Typically, the determination of prediction intervals is cast in the probabilistic setting with the notion of confidence level playing a pivotal role. Depending upon the nature of the model and the detailed assumptions made in advance, the building of the prediction intervals can be straightforward or could entail a carefully structured detailed optimization procedure.

In linear regression, prediction intervals come with a well–defined and commonly used assumptions as to the character of noise and the linearity of the model. Under such conditions, the intervals are easily determined with the formulas available in numerous textbooks, see [94], [95]. There are numerous refinements made to the generic regression model and those come with the associated prediction intervals.

In nonlinear models, e.g., neural networks, the formation of prediction intervals is more complicated and quite often associated with additional assumptions and optimization involved. The studies reported in [96]–[101] elaborate on ways of forming and optimizing prediction intervals for neural networks. In [63], Khosravi proposed to limit the spatial search of neural network structures by prediction intervals, thereby reducing potential candidates by 77%, and improving prediction accuracy.

Surprisingly, in fuzzy models [101], there are no investigations as to the probabilistic underpinnings of the results (where in fact the rules are built by involving local regression analyses), which in the sequence do not yield any prediction intervals. There are quite limited probabilistic-inclined analyses as reported in [102], [103]; however, they do not address the quality of results in terms of their associated interval characterization.

4.1 Rule-based models: concise structural and design considerations

In this section, we briefly recall the architectural aspects of rule-base models along with the commonly encountered design practices. This brief recollection is helpful as being directly linked to the buildup of the prediction intervals. In this chapter, we consider the structure of rules with linear function (11). The linear aggregation of the local models with the activation levels is shown in (14). The design of a rule-based model is carried as a two-phase process: (i) formation of information granules of the condition part of the rules, and

(ii) estimation of optimal parameters of the local linear functions forming the conclusions of the corresponding rules, see the detailed process in Chapter 2.1.

4.2. Design of prediction intervals for rule-based model

Individual regression local models forming in the conclusion parts of the rules can be associated with the confidence and prediction intervals; their development follows the standard construct known in regression analysis. An important step is to build (aggregate) prediction intervals coming from individual rules.

4.2.1 Boolean rules

Considering a certain single rule for the model constructed with the use of the *K*-Means clustering algorithm, a prediction interval $\hat{Y}$ for any input $\boldsymbol{x}$ and some confidence level is expressed in the form

$$\begin{gathered}\hat{Y} = [\hat{y}^-, \hat{y}^+] = [\omega + \hat{\boldsymbol{a}}^T(\boldsymbol{x}-\boldsymbol{v}) - t_{N-n}\sqrt{\hat{\sigma}^2[1+\boldsymbol{z}^T(Z^TZ)^{-1}\boldsymbol{z}]}, \\ \omega + \hat{\boldsymbol{a}}^T(\boldsymbol{x}-\boldsymbol{v}) + t_{N-n}\sqrt{\hat{\sigma}^2[1+\boldsymbol{z}^T(Z^TZ)^{-1}\boldsymbol{z}]}]\end{gathered} \tag{50}$$

where

$$\hat{\sigma}^2 = \frac{(\boldsymbol{target'} - Z\hat{\boldsymbol{a}})^T(\boldsymbol{target'} - Z\hat{\boldsymbol{a}})}{N-n} \tag{51}$$

and

$$\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{v} \tag{52}$$

t stands for t-distribution, which is popular in the statistical analyses. The shape of t-distribution is related to the degrees of freedom ($N-n$). Now considering that we have *c* rules and for each of them we have constructed the model as above with the optimal parameters, and they are the aggregation of the models (as a matter of fact, only a single model is invoked for any $\boldsymbol{x}$). The rules assume the form

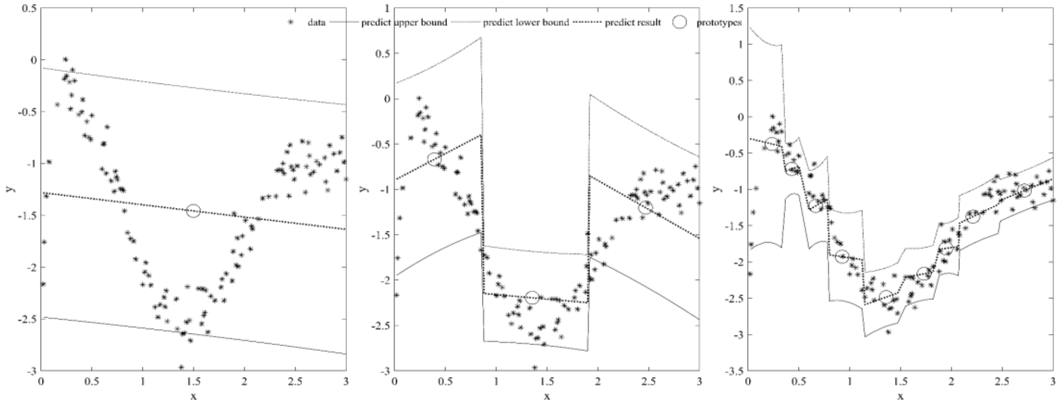
$$\text{-if } \boldsymbol{x} \text{ is } A_i \text{ then } y = \omega_i + \boldsymbol{a}_i^T(\boldsymbol{x} - \boldsymbol{v}_i) \tag{53}$$

where A_i are sets formed by the K-Means algorithm. More specifically, we have the prediction result $\hat{y} = \omega_{i0} + \widehat{\boldsymbol{a}}_{i0}^T(\boldsymbol{x} - \boldsymbol{v}_{i0})$, where $i_0 = \text{argMin}_{i=1,2,\dots,c}\|\boldsymbol{x} - \boldsymbol{v}_i\|^2$ and the corresponding prediction interval is formed as $\widehat{Y}_{i0}$. In a nutshell, the prediction intervals are implied by the interval associated with a single rule.

Use one-dimensional data as an example, it is governed by the nonlinear function

$$y = f(x) + noise$$
$$f(x) = \frac{1}{x} sin(3x) + 0.2^x + log(0.1x) \tag{54}$$

where *noise* is a normally distributed random variable $N(0, 0.15)$. This formula is popular in the literature. We pick up randomly N = 300 input-output pairs of data. Figure 4.1 visualizes the performance of the models in terms of the generated prediction bounds; here the results are reported for the selected values of the rules as 1, 3, 9.



(*a*) (*b*) (*c*)

Figure 4.1 Prediction bounds produced by rule-based models for selected values of *c*: (*a*) *c*=1 (*b*) *c*=3 (*c*) *c* =9.

4.2.2 Fuzzy rule-based model

In the case of the rules constructed by the *FCM* clustering [104], as several rules are fired at the same time with the corresponding degrees of firing (membership), we determine the prediction interval associated with the individual rule and then complete their aggregation. The calculations of the prediction intervals for each rule follow the same way as presented before for the Boolean rules. Because of the membership grades, the overall prediction interval resulting from all *c* rules is expressed as follows

$$\hat{Y} = \sum_{i=1}^{c} A_i(\boldsymbol{x})\hat{Y}_i \tag{55}$$

where $\hat{Y}_i = [\hat{y}_i^-, \hat{y}_i^+]$. In terms of the detailed computing (note that interval calculus is engaged in [105]), the above expression reads as follows,

--lower bound of the prediction interval

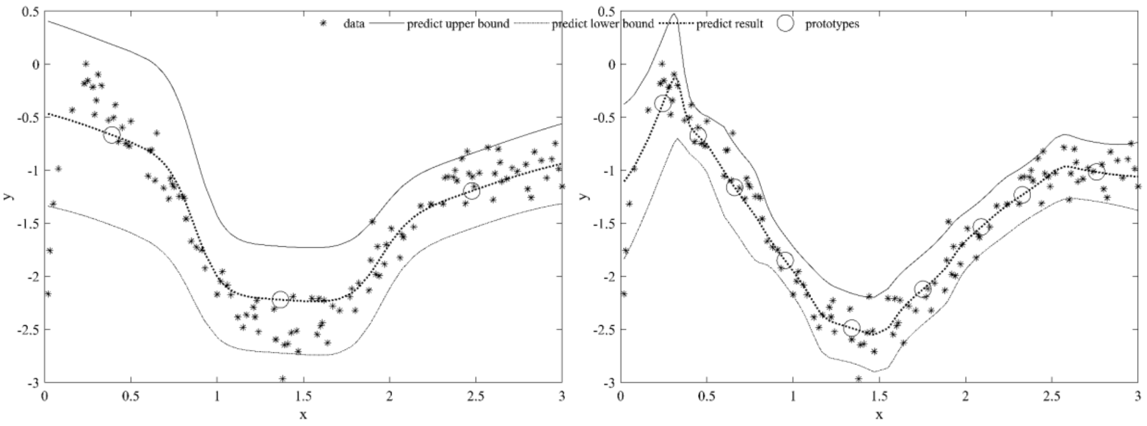
$$\hat{y}^- = \sum_{i=1}^{c} A_i(\boldsymbol{x})\hat{y}^- \tag{56}$$

--upper bound of the prediction interval

$$\hat{y}^+ = \sum_{i=1}^{c} A_i(\boldsymbol{x})\hat{y}_i^+ \tag{57}$$

In essence, structurally the above formula follows the same expression as being used to aggregate numeric values coming from the rules.

Similarity, we use the one-dimensional data mentioned above to do the illustration. Figure 4.2 depicts the prediction intervals with optimized parameters *m*, where (*a*) *c*=3 and (*b*) *c*=9.



(*a*) (*b*)

Figure 4.2 Prediction bounds produced by rule-based models for selected values of *c*: (*a*) *c*=3 (*b*) *c*=9.

4.3 Evaluation of quality of information granules of results of rule-based models

As the results produced now are in the form of information granules, the quality of the model can be expressed by means of the two performance indexes which are pertinent to the granular nature of the output, viz. coverage and specificity [106], [107]. The detailed formula is shown in Chapter 2.3 (23) and (25), where the lower and upper bound are generated by prediction intervals $\hat{y}^{-}$ and $\hat{y}^{+}$, respectively.

Both the coverage and specificity have to be as high as possible. Because of their conflicting nature, an alternative is to assess the quality of prediction intervals (viz. the granular manifestation of the rule-based model). Here we consider the product of the coverage and specificity,

$$V = cov * sp \tag{58}$$

An overall development of the granular model showing the main phases is displayed in Figure 4.3. We train the rule-based model with training data to obtain the optimal parameters c, m and centers $\boldsymbol{a}$, which are used for testing data to generate a partition matrix and then produce predicted output space; the quality of the model is evaluated by *RMSE* between data target and predicted output. Next, based on the predicted output, the prediction interval is produced and assessed by the granular nature *V*.

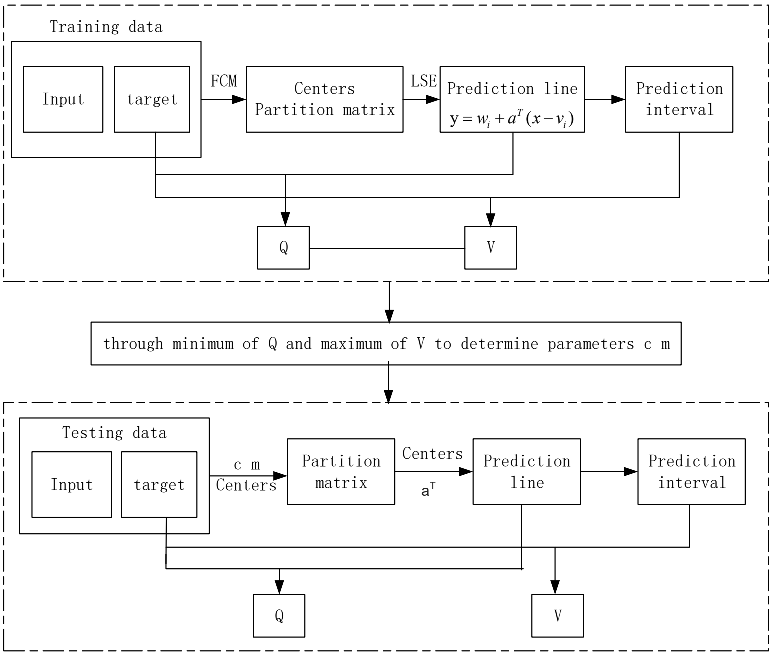


Figure 4.3 An overall design of granular rule-based model.

4.4 Experimental studies

In the series of experiments, we demonstrate the approach considering both synthetic data and those publicly available. As the *FCM* is being used for the construction of the rule-based model, the number of rules (c) and the fuzzification coefficient (m) are experimented to assess their impact on the performance of the model. Several selected values of the confidence level (α) [108] are considered; in particular in the following experiments, we assume the value of α is set to 0.05.

Synthetic data

The one-dimensional data mentioned in Chapter 4.2.1. The training and testing data are divided by 70-30%. Proceeding with the *K*-means clustering, the results are reported for the varying number of rules; as a reference we also consider *c* =1 (viz. a single linear regression model). The performance is reported in terms of the sum of squared errors (*Q*), coverage (*Cov*), specificity (*Sp*) and the product (*V*), as seen in Table 4.1.

Table 4.1 Performance of the model reported for selected values of *c* and *α*=0.05.

c	1	2	3	4	5	6	7	8	9
$Q_{train,}$	0.48	0.12	0.07	0.05	0.05	0.03	0.03	0.02	0.02
Q_{test}	0.50	0.13	0.17	0.18	0.15	0.13	0.14	0.12	0.07
Cov_{train}	0.96	0.94	0.93	0.90	0.91	0.92	0.92	0.94	0.92
Cov_{test}	0.95	0.94	0.93	0.93	0.93	0.94	0.94	0.94	0.95
Sp_{train}	0.15	0.60	0.72	0.73	0.73	0.77	0.78	0.82	0.83
Sp_{test}	0.20	0.60	0.53	0.56	0.57	0.60	0.58	0.62	0.68
V_{train}	0.15	0.56	0.62	0.65	0.67	0.71	0.72	0.77	0.77
V_{test}	0.19	0.56	0.49	0.52	0.53	0.56	0.54	0.58	0.64

When the confidence level is fixed, with the increase of clusters, the product of coverage and specificity is larger, and when *c* is 9, the performance yields the highest value.

The same data (along with their split into the training and testing part) are used in the construction of fuzzy rule-based model (when using the *FCM* algorithm); the results are reported in the same way as before. However, we encounter an additional parameter, namely a fuzzification coefficient (*m*). Here the range of *m* is from 1.05 to 3, and the step is equal to 0.05. Its impact on the performance of the model is investigated.

Table 4.2 shows the results regarded as a function of *c* and *m*, with the increase of clusters, *V* increases quickly.

Table 4.2 Performance of the model reported for selected values of *c* with optimized *m*.

m \c	1.1\\2	1.6\\3	1.45\\4	1.3\\5	1.95\\6	1.7\\7	1.65\\8	1.6\\9
$Q_{train},$	0.08	0.06	0.03	0.03	0.03	0.03	0.02	0.02
Q_{test}	0.09	0.06	0.04	0.03	0.04	0.04	0.03	0.02
Cov_{train}	0.98	0.97	0.95	0.96	0.97	0.96	0.96	0.96
Cov_{test}	0.98	0.94	0.94	0.94	0.97	0.96	0.97	0.98
Sp_{train}	0.62	0.67	0.72	0.77	0.77	0.81	0.83	0.84
Sp_{test}	0.63	0.67	0.70	0.76	0.71	0.74	0.75	0.77
V_{train}	0.61	0.65	0.68	0.74	0.74	0.78	0.80	0.80
V_{test}	0.61	0.63	0.66	0.71	0.69	0.71	0.73	0.76

Figure 4.4(*a*) shows that *V* (product of coverage and specificity) varies from *m* (fuzzification coefficient) for training data. Figure 4.4(*b*) shows that *V* varies from *m* (fuzzification coefficient) for testing data. Figure 4.4(*c*) is the relationship of training data between *Q* and *m.* Figure 4.4(*d*) is the relationship for testing data. Figure 4.5 shows the change of *V* with *c.*

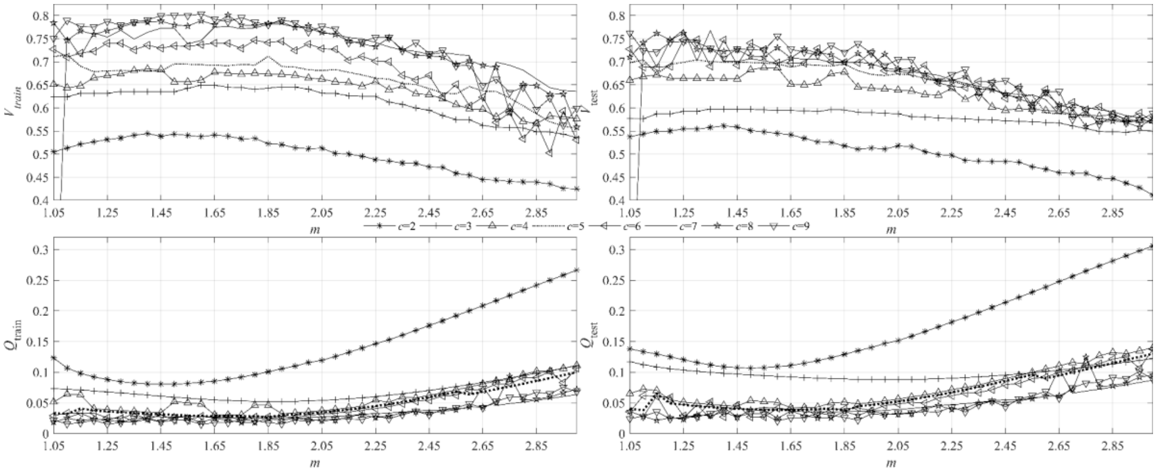


Figure 4.4 Performance of the model as a function of *m*; shown are curves for different values of *c*.

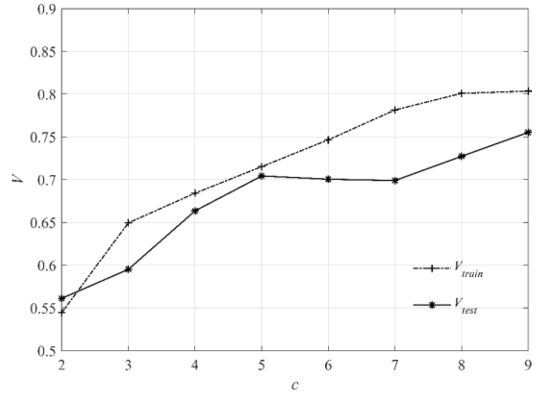


Figure 4.5 *V* as a function of *c.*

According to Figures 4.4-4.5, it is shown that for both training and testing data, the larger the cluster is, the larger *V* is. There is an opposite trend for *Q* with the increase of *c*. Especially in Figure 4.5, apart from clusters, *m* has little influence on *Q*. Usually when *m* is around 1.6, the parameter *Q* gets the minimum value, but *V* decreases with the increase of *m*.

After implementing the process for the same data with *K*-means and Fuzzy *C*-Means, we will compare the results. In Table 4.3, we show the examples for clusters 3 and 9.

Table 4.3 Comparing performance of the model for one-dimensional data.

clusters	performance	*K*-means	*FCM*	clusters	performance	*K*-means	*FCM*
c=3	Q_{train}	0.07	0.06	c=9	Q_{train}	0.02	0.02
	Q_{test}	0.17	0.06		Q_{test}	0.07	0.02
	Cov_{train}	0.93	0.97		Cov_{train}	0.92	0.96
	Sp_{train}	0.72	0.67		Sp_{train}	0.83	0.84
	V_{train}	0.62	0.65		V_{train}	0.77	0.80
	Cov_{test}	0.93	0.94		Cov_{test}	0.95	0.98
	Sp_{test}	0.53	0.67		Sp_{test}	0.68	0.77
	V_{test}	0.49	0.63		V_{test}	0.64	0.76

Through comparing the result shown in Table 4.3, *FCM* performs better than *K*-means when we choose a suitable fuzzification coefficient. Especially *V* increases obviously.

Synthetic two-dimensional data

The well-known two-dimensional synthetic data are described in the form

$$y = f(x_1, x_2) + noise, f(x_1, x_2) = x_1 e^{-{x_1}^2 - {x_2}^2} + sin({x_1}^2) + cos({x_2}^2) \qquad (59)$$

where *noise* is normally distributed noise, *N*(0, 0.15). Again *N* =300 data pairs are selected randomly. The nonlinear function is displayed in Figure 4.6.

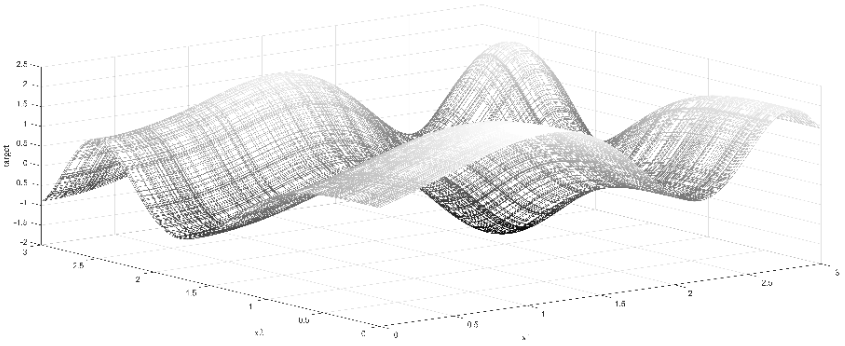


Figure 4.6 Two-dimensional synthetic data.

Again proceeding with the *K*-means clustering, the results are reported for the varying number of rules, saying the sum of squared errors, coverage, and specificity, as shown in Table 4.4.

Table 4.4 Performance of the model reported for selected values of *c* for *α*=0.05.

c	1	2	3	4	5	6	7	8	9
Q_{train},	0.79	0.48	0.40	0.28	0.26	0.18	0.14	0.13	0.10
Q_{test}	1.37	1.29	0.67	0.54	0.32	0.19	0.21	0.19	0.15
Cov_{train}	0.91	0.91	0.92	0.92	0.92	0.93	0.94	0.93	0.44
Cov_{test}	0.05	0.16	0.43	0.48	0.71	0.75	0.82	0.87	0.41
Sp_{train}	0.30	0.46	0.51	0.58	0.60	0.67	0.71	0.73	0.92
Sp_{test}	0.99	0.92	0.80	0.80	0.66	0.70	0.68	0.67	0.88
V_{train}	0.28	0.42	0.47	0.54	0.55	0.63	0.66	0.68	0.41
V_{test}	0.05	0.15	0.34	0.39	0.47	0.53	0.56	0.58	0.36

Concluding from Table 4.4, when we consider cluster 8, the performance is the best for these two-dimensional data, which means that for *K*-means, the prediction result is the best when cluster 8 is considered.

Then the same data are also used in the construction of the fuzzy rule-based model, and we also give two parameters, namely a fuzzification coefficient (*m*) and clusters (*c*). Table 4.5 shows the results regarded as a function of *c* and *m*. *Q* has an obvious decrease, and *V* has the opposite trend.

Table 4.5 Performance of the model reported for selected values of *c* with optimized *m*.

m \c	1.35\2	1.15\3	1.35\4	1.35\5	1.3\6	1.25\7	1.45\8	1.25\9
$Q_{train,}$	0.37	0.28	0.17	0.18	0.14	0.10	0.09	0.07
Q_{test}	0.43	0.35	0.18	0.19	0.15	0.13	0.12	0.09
Cov_{train}	0.94	0.95	0.96	0.97	0.96	0.96	0.96	0.96
Cov_{test}	0.93	0.94	0.94	0.94	0.98	0.96	0.98	0.95
Sp_{train}	0.47	0.58	0.61	0.63	0.67	0.71	0.72	0.75
Sp_{test}	0.39	0.51	0.57	0.59	0.62	0.65	0.65	0.69
V_{train}	0.44	0.55	0.59	0.62	0.64	0.68	0.69	0.71
V_{test}	0.37	0.48	0.54	0.56	0.61	0.63	0.63	0.66

Figure 4.7(*a*) shows that *V* varies from *m* for training data. Figure 4.7(*b*) shows that *V* varies from *m* for testing data. Figure 4.7(*c*) is the relationship of training data between *Q* and *m*. Figure 4.7(*d*) shows the relationship for testing data, while Figure 4.8 depicts the change of *V* with *c*.

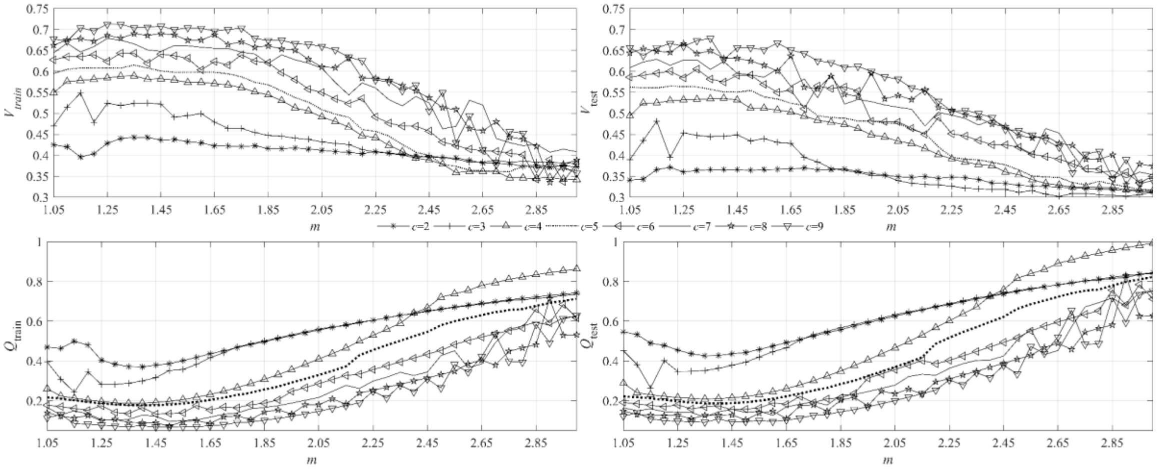


Figure 4.7 Performance of the model as a function of *m*; shown are curves for different values of *c*.

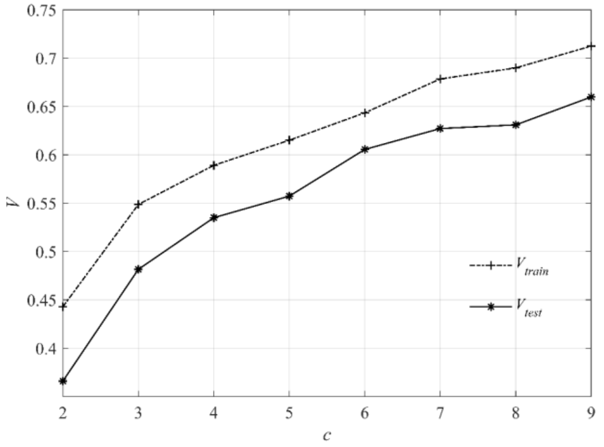


Figure 4.8 *V* regarded as a function of *c*.

Through the above figures and tables one can conclude that the larger the cluster is, the bigger *V* is, but the smaller *Q* is. *Q* and *V* exhibit the opposite trend with the increase of *m*. *V* decreases when *m* becomes bigger, but *Q* decreases firstly, and then increases when *m* is bigger than 1.4.

In Figure 4.9, we only report the prediction intervals obtained for 9 clusters with its suitable *m,* and as the data are two-dimensional, the prediction intervals are plotted versus the original outputs.

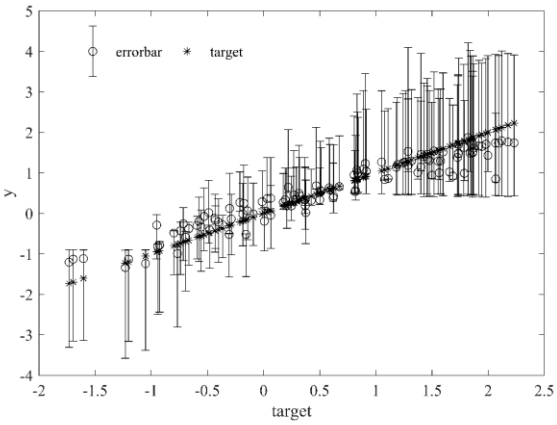


Figure 4.9 Prediction intervals vs. numeric outputs for *c*=9.

Table 4.6 Performance of the models for two-dimensional data.

clusters	performance	*K*-means	*FCM*	clusters	performance	*K*-means	*FCM*
c=3	Q_{train}	0.40	0.28	*c*=9	Q_{train}	0.10	0.07
	Q_{test}	0.67	0.35		Q_{test}	0.15	0.09
	Cov_{train}	0.92	0.95		Cov_{train}	0.44	0.96
	Sp_{train}	0.51	0.58		Sp_{train}	0.92	0.75
	V_{train}	0.47	0.55		V_{train}	0.41	0.71
	Cov_{test}	0.43	0.94		Cov_{test}	0.41	0.95
	Sp_{test}	0.80	0.51		Sp_{test}	0.88	0.69
	V_{test}	0.34	0.48		V_{test}	0.36	0.66

Then we will compare the results for the same data with *K*-means and Fuzzy *C*-means in Table 4.6 and we show the examples in the case of the clusters 3 and 9. All the results of

FCM are bigger than that of *K*-mean*s*, and especially when cluster is 9, *V* significantly increases.

Machine Learning data

Iris Data description:

The data set contains three classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter two are not linearly separable from each other, see https://archive.ics.uci.edu/ml/datasets/iris.

Name	Objects	Attributes	Categories
Iris	150	4	3

Again, proceeding with the *K*-means clustering for Iris data, the results are reported for the varying number of rules. Here we put clusters are from 2 to 8, and then give the performance in terms of the sum of squared errors, coverage, and specificity, see Table 4.7.

Table 4.7 Performance of the model reported for selected values of *c* and *α*=0.05.

c	2	3	4	5	6	7	8
Q_{train},	0.27	0.14	0.06	0.05	0.05	0.04	0.04
Q_{test}	0.68	0.53	0.46	0.38	0.28	0.19	0.23
Cov_{train}	0.90	0.92	0.92	0.96	0.95	0.94	0.97
Cov_{test}	0.29	0.43	0.56	0.64	0.86	0.86	0.87
Sp_{train}	0.48	0.62	0.74	0.52	0.77	0.79	0.78
Sp_{test}	0.89	0.85	0.80	0.73	0.43	0.47	0.62
V_{train}	0.44	0.57	0.68	0.49	0.73	0.74	0.76
V_{test}	0.26	0.36	0.45	0.47	0.37	0.40	0.55

From Table 4.7, the performance is the best when cluster 8 is considered. Thus, the prediction result is best when iris data are divided into 8 clusters.

Then the same data are also used in the construction of fuzzy rule-based model, and Table 4.8 shows the results regarded as a function of *c* and *m* like before. And the range of *m* is from 1.04 to 3, and step is 0.1.

Table 4.8 Performance of the model reported for selected values of *c* with optimized *m*.

m \c	2.74\2	1.64\3	1.84\4	1.84\5	1.84\6	1.34\7	1.34\8
Q_{train},	0.15	0.08	0.04	0.04	0.04	0.03	0.03
Q_{test}	0.18	0.10	0.07	0.08	0.08	0.09	0.09
Cov_{train}	0.93	0.93	0.94	0.94	0.94	0.95	0.96
Cov_{test}	0.93	0.92	0.93	0.93	0.96	0.97	0.96
Sp_{train}	0.58	0.70	0.76	0.77	0.80	0.70	0.81
Sp_{test}	0.54	0.67	0.66	0.65	0.66	0.63	0.65
V_{train}	0.54	0.65	0.72	0.73	0.75	0.67	0.77
V_{test}	0.51	0.61	0.61	0.61	0.64	0.61	0.63

Figure 4.10(*a*) shows that *V* varies from *m* for training data. Figure 4.10(*b*) shows that *V* varies from *m* for testing data. Figure 4.10(*c*) is the relationship of training data between *Q* and *m.* Figure 4.10(*d*) is the relationship for testing data. Figure 4.11 shows the change of *V* with *c.*

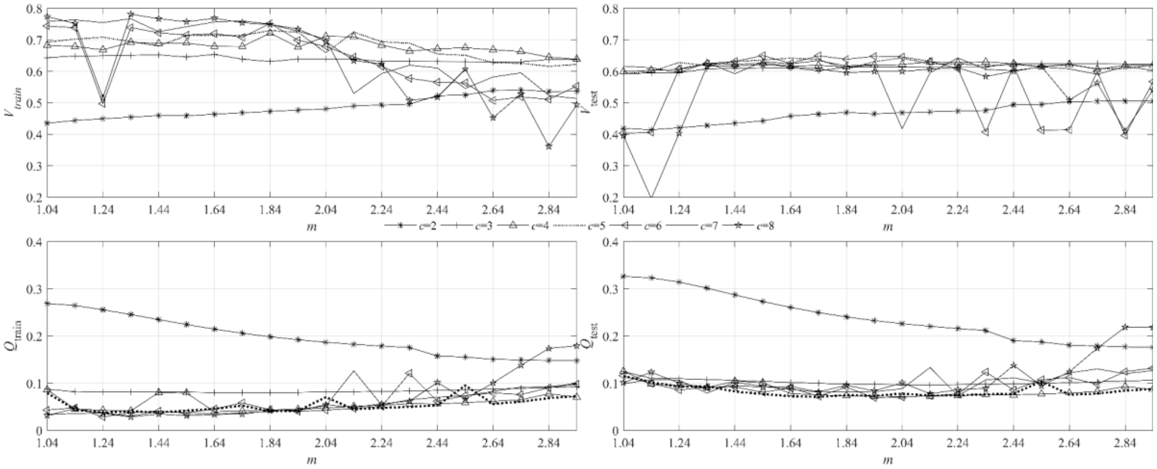


Figure 4.10 Performance of the model as a function of *m*; shown are curves for different values of *c*.

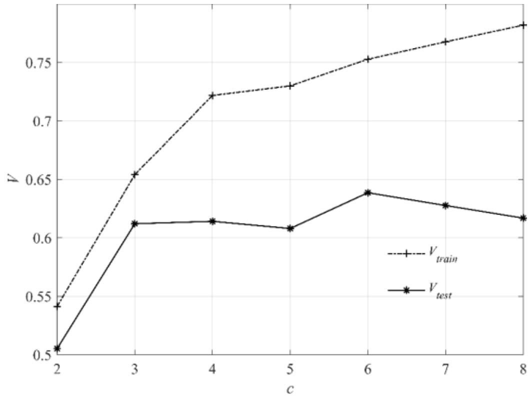


Figure 4.11 *V* as a function of *c*.

Through the above figures and tables, we conclude that the larger the cluster is, the bigger *V* of training data is, but the smaller *Q* is. For the testing data, when cluster is 6, the model gets its maximum value. *Q* and *V* have the opposite trend with the increase of *m*. For training data, *V* decreases with the increase of *m*, but *Q* increases; for testing data, *V* and *Q* usually change a little except cluster 2.

Next, we only give the prediction interval for 8 clusters with the suitable *m,* and as the data are multi-dimensional, the prediction intervals are plotted versus the original outputs, see Figure 4.12.

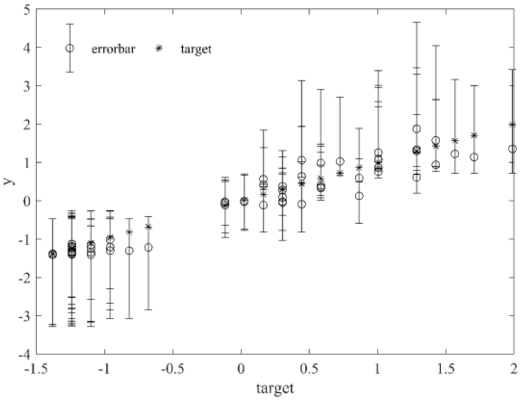


Figure 4.12 Prediction intervals vs. numeric outputs with cluster 8.

Then we will compare the results for the same data with *K*-means and Fuzzy *C*-means as shown in Table 4.9, we show the performance indices when cluster is 3 and 8.

Table 4.9 IRIS data-comparative analysis.

clusters	performance	*K*-means	*FCM*	clusters	performance	*K*-means	*FCM*
c=3	Q_{train}	0.14	0.08	*c*=8	Q_{train}	0.04	0.03
	Q_{test}	0.53	0.10		Q_{test}	0.23	0.09
	Cov_{train}	0.92	0.93		Cov_{train}	0.97	0.96
	Sp_{train}	0.62	0.70		Sp_{train}	0.78	0.81
	V_{train}	0.57	0.65		V_{train}	0.76	0.77
	Cov_{test}	0.43	0.92		Cov_{test}	0.87	0.96
	Sp_{test}	0.85	0.67		Sp_{test}	0.62	0.65
	V_{test}	0.36	0.61		V_{test}	0.55	0.63

Banknote data description:

Data were extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400x400 pixels. The object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained. Wavelet Transform tool were used to extract features from images, see https://archive.ics.uci.edu/ml/datasets/banknote+authentication.

Attribute Information:

1. variance of Wavelet Transformed image (continuous)

2. skewness of Wavelet Transformed image (continuous)

3. curtosis of Wavelet Transformed image (continuous)

4. entropy of image (continuous)

Name	Objects	Attributes	Categories
Banknote authentication	1372	4	2

Again proceeding with the *K*-means clustering for Banknote data, the results (the sum of squared errors, coverage, and specificity) are reported for the varying number of rules, where clusters are from 2 to 9, as shown in Table 4.10.

Table 4.10 Performance of the model reported for selected values of *c* and *α*=0.05.

c	2	3	4	5	6	7	8	9
Q_{train},	0.38	0.28	0.25	0.23	0.20	0.18	0.17	0.16
Q_{test}	0.73	0.60	0.52	0.49	0.40	0.30	0.18	0.16
Cov_{train}	0.93	0.90	0.89	0.90	0.91	0.90	0.90	0.90
Cov_{test}	0.32	0.42	0.53	0.60	0.71	0.78	0.82	0.90
Sp_{train}	0.61	0.67	0.69	0.70	0.72	0.74	0.75	0.75
Sp_{test}	0.85	0.83	0.81	0.78	0.76	0.75	0.74	0.72
V_{train}	0.57	0.60	0.61	0.64	0.66	0.67	0.67	0.68
V_{test}	0.27	0.35	0.43	0.47	0.54	0.59	0.61	0.65

From Table 4.10, the performance is the best when cluster is 9. Thus, the prediction result is the best when Banknote data are divided into 9 clusters. Then the same data are also used in the construction of fuzzy rule-based model. Table 4.11 shows the results regarded as a function of *c* and *m* as before. The values of *m* vary from 1.04 to 3, with the step of 0.1.

Table 4.11 Performance of the model reported for selected values of *c* with optimized *m*.

m \c	1.14\\2	1.24\\3	1.24\\4	1.44\\5	1.44\\6	1.44\\7	1.34\\8	1.24\\9
Q_{train},	0.35	0.23	0.23	0.17	0.15	0.12	0.11	0.11
Q_{test}	0.37	0.25	0.23	0.18	0.16	0.13	0.12	0.13
Cov_{train}	0.94	0.91	0.91	0.94	0.94	0.94	0.94	0.94
Cov_{test}	0.94	0.91	0.92	0.94	0.95	0.94	0.94	0.93
$Spec_{train}$	0.61	0.67	0.69	0.70	0.73	0.75	0.76	0.76
$Spec_{test}$	0.55	0.65	0.65	0.68	0.71	0.73	0.74	0.76
V_{train}	0.57	0.61	0.62	0.64	0.68	0.70	0.71	0.71
V_{test}	0.52	0.59	0.60	0.64	0.67	0.68	0.70	0.70

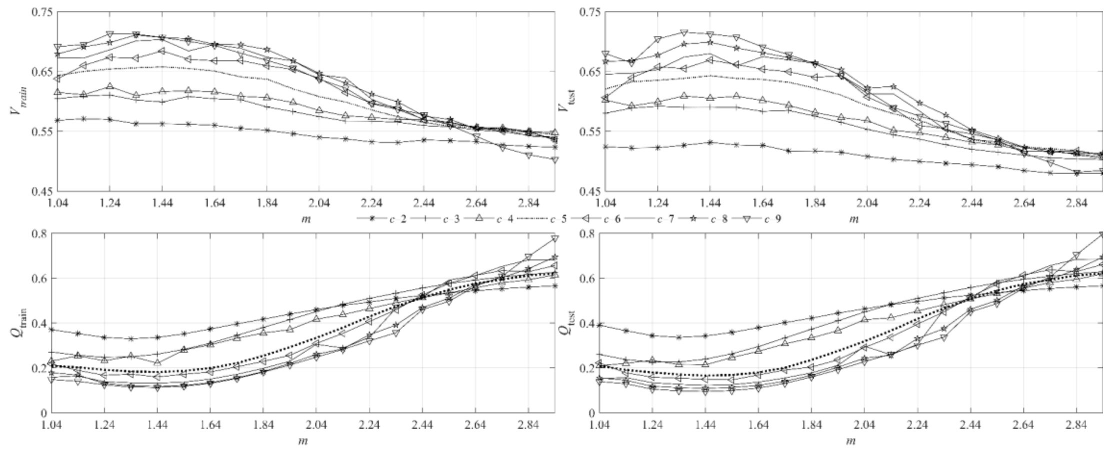


Figure 4.13 Performance of the model as a function of *m*; shown are curves for different values of *c*.

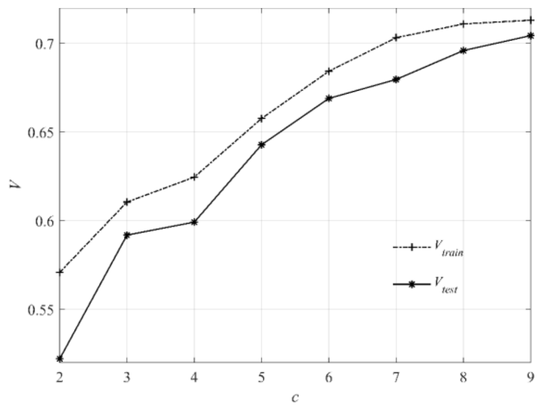


Figure 4.14 *V* regarded as a function of *c*.

Figure 4.13(*a*) shows that *V* varies from *m* for training data. Figure 4.13(*b*) shows that *V* varies from *m* for testing data. Figure 4.13(*c*) is the relationship of training data between *Q* and *m.* Figure 4.13(*d*) is the relationship for testing data. Figure 4.14 shows the change of *V* for varying values of *c.*

Concluded from the above figures and table, the larger the cluster is, the bigger *V* is, but the smaller *Q* is. And *V* has a decrease trend with the increase of *m*, but when *m* varies from 1.04 to around 1.5, *Q* decreases, and then it increases quickly.

Next, we give the prediction interval for 9 clusters with the suitable *m*. As the data are multi-dimensional, the prediction intervals are plotted versus the original outputs, see Figure 4.15.

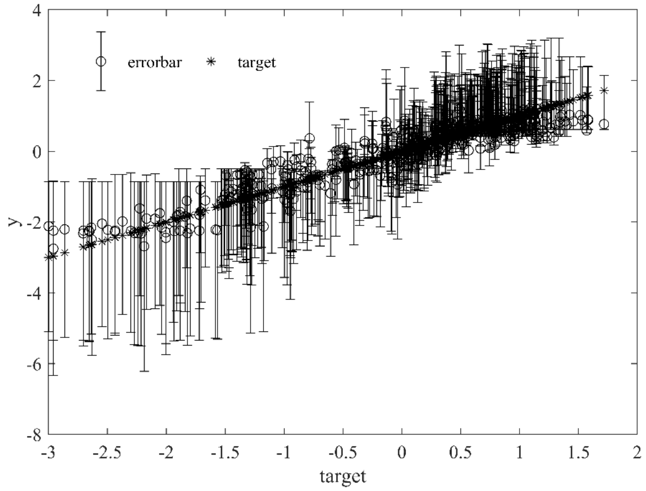


Figure 4.15 Prediction intervals vs. numeric outputs with cluster 9.

Next, we compare the results for the same data with *K*-means and Fuzzy *C*-means, as shown in Table 4.12.

Table 4.12 Comparing performance of the model produced for Banknote data.

clusters	performance	*K*-means	*FCM*	clusters	performance	*K*-means	*FCM*
c=3	Q_{train}	0.28	0.23	*c*=9	Q_{train}	0.16	0.11
	Q_{test}	0.60	0.25		Q_{test}	0.16	0.13
	Cov_{train}	0.90	0.91		Cov_{train}	0.90	0.94
	Sp_{train}	0.67	0.67		Sp_{train}	0.75	0.76
	V_{train}	0.60	0.61		V_{train}	0.68	0.71
	Cov_{test}	0.42	0.91		Cov_{test}	0.90	0.93
	Sp_{test}	0.83	0.65		Sp_{test}	0.72	0.76
	V_{test}	0.35	0.59		V_{test}	0.65	0.70

Until now, we have finished all the experiments, and the results of Machine Learning data are similar to that of Synthetic data. Comparing the results shown in above tables, whether it is *K*-means and *FCM*, *Q* decreases and the values of *V* increase. When the number of clusters and the fuzzification coefficient are fixed, the performance of *FCM* is better than that of the results produced by the *K*-means clustering.

4.5 Conclusions

This study raised an important issue of the inherent granularity of results produced by rule-based models, both Boolean and fuzzy set-based. It is shown that prediction intervals are impacted by the essential parameters of the clustering techniques, such as the number of clusters and the fuzzification coefficient.

As a result of comparative analysis, it becomes apparent that fuzzy rule-based models exhibit better performance than rule-based models in terms of the coverage criterion. There are visible jumps between successive prediction intervals implied by the lack of continuity of clustering results generated by the *K*-Means algorithm. From this point of view, there is a tangible advantage of the *FCM* algorithm.

Chapter 5. A granular multicriteria group decision making in renewable energy planning problems

In this chapter, we consider the application of information granules. That is, the combination of information granules and group decision making. To position the formulated problem in a broader context and highlight the multi-criteria and group decision nature of decision problems in the area of renewable energy, in what follows, we complete a concise focused literature review.

A selection of a suitable location for renewable energy facilities, especially a solar one, gives rise to a challenging decision-making problem, which involves a series of usually conflicting criteria and the decision process is realized by a group of experts (group decision-making). Solar energy only needs sunlight as a source, but its cost is high and it is directly impacted by the weather [109]–[111]. Wind energy requires open terrain and has a great impact on wildlife, especially birds [112]–[115]. Although tidal energy contains huge energy, deciding how to effectively use and exploit it becomes a huge problem [116]. Apart from the choice of energy type, there are also a lot of issues that need to be considered for a certain type of energy. Take solar energy as an example. It is a widely distributed, low-density, and intermittent energy source in nature, so choosing the location for solar energy development is particularly important. At the same time, the storage, transmission, and subsequent maintenance of electricity should also be considered.

In recent decades, there have been some studies on these issues. Pablo *et al*. applied Analytic Hierarchy Process (AHP) and Analytic Network Process (ANP) to help a Spanish solar power investment company invest a solar-thermal power plant project or not [117]. In [118], Meryem *et* *al*. combined the Geographic Information System (GIS) and the Multi-Criteria Decision-Making (MCDM) method to assess a suitable location for a solar energy project in the southern region of Morocco. The research concluded that Ouarzazate shows a high adaptability to the installation of photovoltaic power plants. Tolga *et al*. integrated TOPSIS (a multicriteria decision making (MCDM) technique) and fuzzy AHP to solve the energy planning decision-making problem. The conclusion was that wind is an optimal energy compared to other energy sources, see [119]. In 2015, Sunil *et al*. proposed a fuzzy Decision-Making Trial and Evaluation Laboratory based methodology to evaluate the solar power development key enables. Through the analysis of 16 factors in India, it was found that the key was state level and power sector reforms [120]. In terms of the optimal solar site selection in Isfahan-Iran, Mahmood *et al*. combined fuzzy logic, weighted linear combination (WLC) and Multiple Criteria Decision Making (MCDM) Process. It is finally determined that some areas in Isfahan, Borkhar, etc. have the potential and suitability for the construction of solar power plants see [121]. Similarly, Seda *et al*. used the combination of both pyranometer and Photovoltaic Geographical Information System (PVGIS) and AHP to analyze the ideal location for installing PV power plant. The experiment showed that the best location is Kulluk [122]. Lindberg *et al*. used multi-criteria analysis with a Boolean approach and power flow simulations to do the geographical assessment and the impact analysis on the grid, respectively, of three different sizes of PV parks to find the optimal one for utility-scale solar guides [123]. Davoudabadi *et al*. presented an approach based on data envelopment analysis (DEA) and fuzzy simulation of interval-valued intuitionistic fuzzy sets (IVIFSs) to evaluate a renewable energy project. Among multiple candidate energy options, wind energy and fossil energy are determined to be the better options. With further analysis, the advantages of wind energy have gradually expanded and become the most suitable choice [124]. Hirushie *et al*. proposed a framework that use system dynamics for rating renewable energy project deployment scenarios and the fuzzy logic-based optimization algorithm is introduced to optimize system capacities and energy mix [125].

In the above problems, the decision-making process is essentially a process of comparison and selection among a limited number of alternatives. Through a comprehensive consideration of various factors and at the same time assigning appropriate weights to each factor, an optimal solution is finally constructed.

5.1 Problem formulation – an overall of design process

In this study, we consider a decision-making scenario in which we have *p* criteria, *c* decision-makers (experts) and *N* alternatives {*alter*1, *alter*2,…, *alterN*}. The problem is structured as illustrated in Figure 5.1; the same figure also shows the main phases of processing with emphasis focused on the visualization of criteria and individual decision-makers by forming a criteria-participants matrix. Furthermore, we highlight the aspect of information granularity which is inherently associated with the diversity of the group nature of the decision process and its quantification.

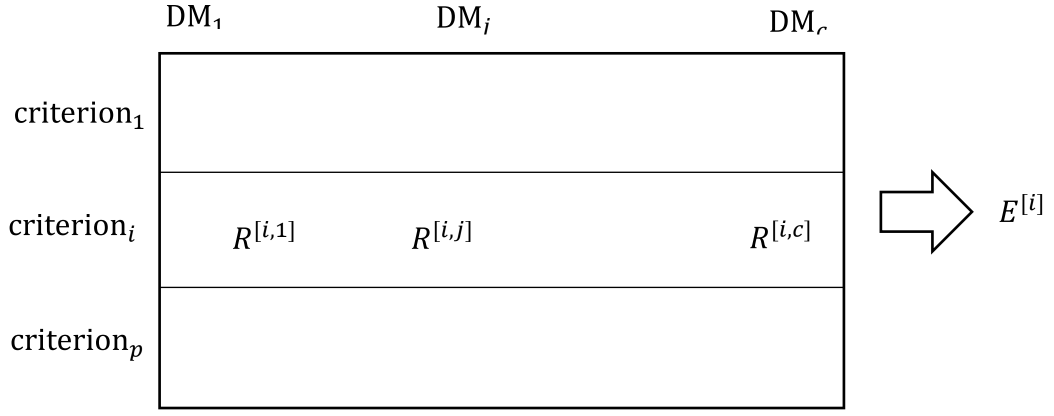


Figure 5.1 Criteria-decision-makers (DM$_j$) array.

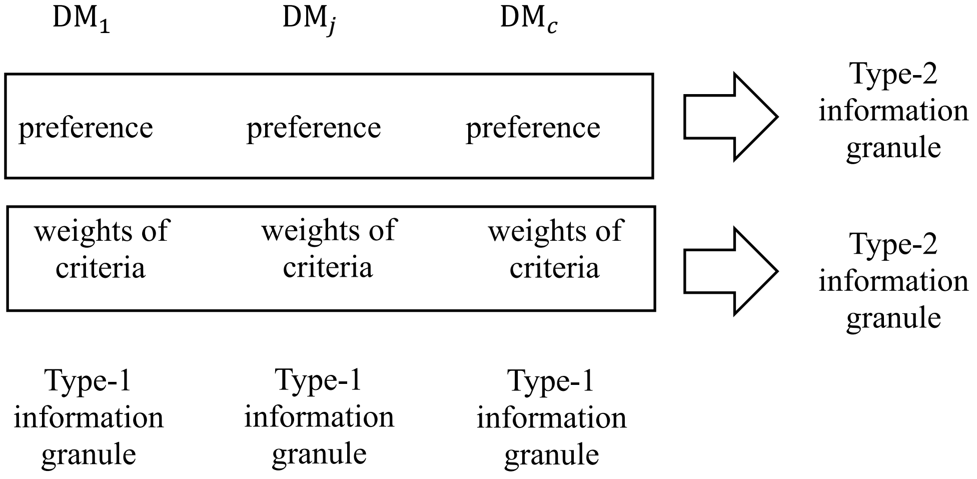


Figure 5.2 Elevation of type of information granule reflective with the movement from individual DMs to a group decision-making.

Considering the *i*-th criterion and the *j*-th decision maker, the decision process is realized with respect to *n* alternatives and their weight. Both of them involve a process of pairwise comparison (described later in detail). In a nutshell, an AHP method [126] considered here is used and *N* alternatives are compared pairwise resulting in an *N* x *N* dimensional matrix of pairwise comparisons $R^{[i,j]} = [r^{[i,j]}]$. The same process of pairwise comparison is completed for the weights of the criteria resulting in a *p* x *p* matrix in the form $W^{[i,j]} =[w^{[i,j]}]$. Once the matrices have been constructed experimentally, the *N*-dimensional vector of preferences of the alternatives associated with $R^{[i,j]}$ is formed as $\boldsymbol{e}^{[i,j]}$. The entries of the vector are regarded as degrees of preference assuming values in the interval [0,1], thus $\boldsymbol{e}^{[i,j]}$ is a discrete fuzzy set defined over a set of alternatives. The method of pairwise comparison also quantifies a level of consistency of the pairwise comparison which is described by the corresponding inconsistency index $\lambda^{[i,j]}$. The same process is carried out for the weight matrix $W^{[i,j]}$ of dimensionality *p* by *p*, resulting in weight vector $\boldsymbol{w}^{[i,j]}$ of dimensionality *p* and the corresponding inconsistency index $\theta^{[i,j]}$. By analyzing a certain row, see

Figure 5.1, it is noticeable that the alternatives evaluated in light of the same criterion by different decision-makers produce a family of fuzzy sets with different membership functions. To aggregate these results and develop a global view conveyed by the group, the individually obtained fuzzy sets have to be aggregated to arrive at the global view. With the intent of describing the diversity of view, the result is formed as an information granule of type-2, namely $E^{[i]}$ (note that $\boldsymbol{e}^{[i,j]}$s are type-1 information granules). This aggregation is obtained with the use of the principle of justifiable granularity [57]; the principle realizes the process of elevating the type of aggregated information granules. In brief, the transformation is expressed as $E^{[i]} = G(E^{[i,1],} E^{[i,2]},.., E^{[i,c]})$, where G stands for the process of granulation guided by the principle of justifiable granularity. As a result, $E^{[i]}$ is a type-2 information granule defined over the space of n alternatives, $E^{[i]} =[\ E^{[i]}_1, E^{[i]}_2,\ldots, E^{[i]}_n]$. $E^{[i]}_j$ denotes a degree of preference (information granule defined over [0,1]) of the j-th alternative expressed with respect to the i-th criterion. In the construction of $E^{[i]}$, the principle involves also the values of the inconsistency index associated with the corresponding fuzzy sets. Likewise we proceed with the fuzzy sets of the weights of the criteria obtained by processing $W^{[i,j]}$, which results in the fuzzy sets of weights of the criteria $\boldsymbol{w}^{[i,j]}$. As in the case of preference levels of alternatives, the aggregation of the results delivered by the individual decision-makers is carried out with the aid of the principle of justifiable granularity resulting in the weight expressed as an information granule (e.g., a fuzzy set) of type-2. Thus, $W^{[i]}$ is an information granule defined over the space of decision-makers. Finally, $E^{[i]}$ and $W^{[i]}$ are aggregated by taking a weighted average

$$E = \sum_{i=1\oplus}^{p} (W^{[i]} \otimes E^{[i]}) \qquad (60)$$

In the above expression, the symbols shown in circles underline that the operations are completed on information granules rather than numeric entities. In sum, E is an information granule of type-

2 defined over the space of *n* alternatives. In other words, the *i*-th coordinate of *E*, E_i captures an overall level of preference of the *i*-th alternative, where the level is expressed as an information granule of type-1 defined over the unit interval. Finally, these information granules of preference E_1, E_2, ..., E_n are ranked with the use of ranking methods such that a linear order is established, where $E_i < E_j$ denotes that E_j is preferred over E_i.

5.2 Literature review

In this section, we briefly recall the key ideas supporting the development.

5.2.1 The analytic hierarchy process (AHP).

The basic idea of the Analytic Hierarchy Process [127], [128] refers to a decision-making method that helps determine preference degrees among a set of alternatives by carrying out a series of pairwise comparisons of preferences of two alternatives at a time. The experimentally determined entries of the pairwise comparison matrix satisfy the two essential properties, namely reflexivity and reciprocity:

-*reflexivity*: the elements on the main diagonal are equal to 1 (they represent situations where a preference of alternative is expressed vis-à-vis itself).

-*reciprocity*: the elements located symmetrically with respect to the main diagonal are inverse of each other.

The entries of the matrix are quantified by using a certain scale; usually, those values are selected from the set {1, 2,..., 9}. If alternative_1 is strongly preferred over alternative_2, then the corresponding entry of the matrix is selected from the upper end of the scale, e.g., 9 or 8. If some moderate preference is considered, the entry of the matrix assumes values in the range 6-7, etc. At the same time the preference of alternative_2 over alternative_1 is taken as a reciprocal value of the estimate already completed, e.g., 1/9.

Having the matrix completed, one determines the maximum eigenvalue (ξ) of the matrix and the corresponding eigenvector. The eigenvector (after normalization) is the estimate of the fuzzy set describing degrees of preference assigned to the individual alternatives. The method comes with a flagging mechanism: the maximum eigenvalue quantifies the consistency in terms of the inconsistency index coming in the following form

$$\lambda = (\xi - N)/(N - 1) \tag{61}$$

where *N* is the number of alternatives. If the inconsistency index λ is greater than some threshold value (say, 0.1), the result lacks consistency and the acquisition of the pairwise matrix has to be repeated. In our research, the value of λ is also used to weight the corresponding eigenvector.

The method can be augmented by a robustness analysis, which helps express a level of robustness of the results when some changes of the pairwise comparison matrix are encountered. In doing this, we analyze on the possible perturbations (due to the subjective way of determining entries from the scale). To quantify the robustness, the following process is considered:

Given some reciprocal matrix *R*, we perturb their entries in an additive way by adding random integers coming from the uniform distribution [-ε, ε] to the integer entries of *R*. The modified entries of *R* are clipped to the range of the assumed scale, using which *R* has been estimated. Note that once some entries have been modified, the entries positioned symmetrically with respect to the diagonal also need to be changed. The modified *R*, denoted as R_ε, produces some fuzzy sets of preferences $\boldsymbol{e}_\varepsilon$. It is compared with the original fuzzy set $\boldsymbol{e}$ produced by *R*. Similarly, one compares the original ranking provided by $\boldsymbol{e}$ to the one produced by $\boldsymbol{e}_\varepsilon$. Given ε, the experiment is repeated a number of times, e.g., 1,000 and the resulting Euclidean distance between $\boldsymbol{e}$ and $\boldsymbol{e}_\varepsilon$

is computed as well as the Hamming distance between the rankings. These two distances were obtained for selected values of ε quantifying the robustness of the AHP method.

As an example, we consider 5 by 5 matrix *R* with some entries. We run the calculations 1,000 times with several values of ε, namely 1, 2, 3, and 4. The histograms of inconsistency values and distances between $\boldsymbol{e}$ and $\boldsymbol{e}_{\varepsilon}$ for different values of ε are shown in Figure 5.3. Especially when ε is 1 or 2, most values of the inconsistent index are less than or around 0.1 and the corresponding distances are 0.

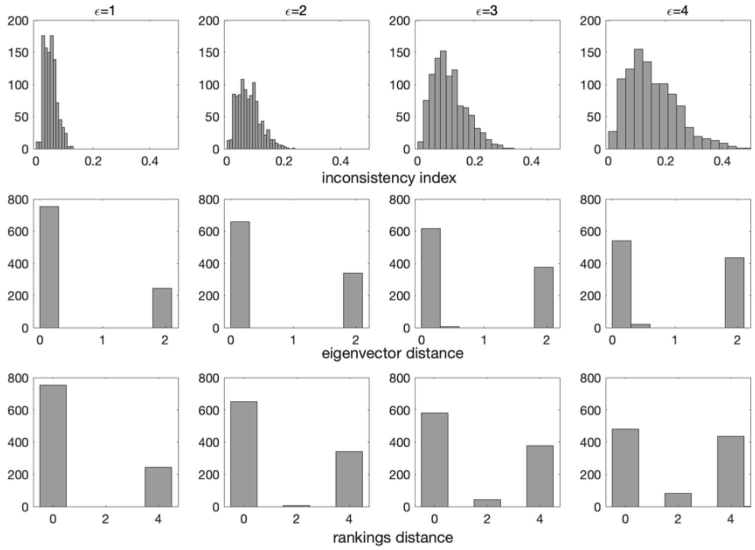


Figure 5.3 Difference of results (consistency, distance between fuzzy sets and rankings) between original and modified *R* as a function of ε.

5.2.2 Arithmetic of information granules

Given two information granules *A* and *B* defined in [0,1], we determine their product and sum. We consider both intervals, namely *A* = [*a, b*] and *B*= [*c, d*] and fuzzy set with triangular membership functions. The information granules are defined in the interval [0,1]; *m* and *n* are the modal values of *A* and *B* while *a* and *c* are the lower bounds and *b* and *d* are the upper bounds.

Let us proceed with the addition of *A* and *B*, i.e., $A \oplus B$:

-intervals. We obtain the following result

$A \oplus B = [a+c, b+d]$

-triangular fuzzy sets

To calculate the resulting fuzzy set *C*, we use the extension principle [36]. Given are the triangular fuzzy sets *A* and *B* as follows

$$A(x) = \begin{cases} (x-a)/(m-a)\,, \text{if } x \in [a,m] \\ (b-x)/(b-m)\,, \text{if } x \in [m,b] \\ 0, \qquad\qquad \text{otherwise} \end{cases}; \quad B(y) = \begin{cases} (y-c)/(n-c)\,, \text{if } y \in [c,n] \\ (d-y)/(d-n)\,, \text{if } y \in [n,d] \\ 0, \qquad\qquad \text{otherwise} \end{cases} \qquad (62)$$

The sum of them is $C(z) = A(x) \oplus B(y)$, where modal value ($C(z) = 1$) is attained at $z = m + n$.

If $x < m$ and $y < n$, one has $z = x + y$ and we involve the increasing parts of fuzzy sets *A* and *B*. $z = x + y = (m-a)\alpha + a + (n-c)\alpha + c = (a+c) + (m+n-(a+c))\alpha$. Thus the membership function is

$$C(z) = \alpha = (z - (a+c))/((m+n) - (a+c)) \qquad (63)$$

If $x > m$ and $y > n$, $z = x + y = b - (b-m)\alpha + d - (d-n)\alpha = (b+d) - (b+d-(m+n))\alpha$, and the membership function of the decreasing part of the resulting fuzzy set is

$$C(z) = \alpha = ((b+d) - z)/((b+d) - (m+n)) \qquad (64)$$

Multiplication of *A* and *B*, i.e., $A \otimes B$

-intervals. The resulting interval is

$A \otimes B$= [min(*ac*, *ad*, *bc*, *bd*), max(*ac*, *ad*, *bc*, *bd*)]

-triangular fuzzy sets

Similarly, we have the triangular fuzzy sets *A* and *B* as mentioned above, and the product of them is $D(z) = A(x) \otimes B(y)$, where modal value is obtained by $z = mn$.

If $x < m$ and $y < n$, one has $z = xy = [(m-a)\alpha + a][(n-c)\alpha + c] = ac + [a(n-c) + c(m-a)]\alpha + (m-a)(n-c)\alpha^2 = f(\alpha)$; thus the membership function of the increasing part of *D* is

$$\begin{aligned} D(z) &= \frac{-[a(n-c)+c(m-a)] + \sqrt{[a(n-c)+c(m-a)]^2 - 4(m-a)(n-c)(ac-z)}}{2(m-a)(n-c)} \\ &= \frac{-[a(n-c)+c(m-a)] + \sqrt{[a(n-c)-c(m-a)]^2 + 4(m-a)(n-c)z}}{2(m-a)(n-c)} \end{aligned} \qquad (65)$$

If $x > m$ and $y > n$, $z = xy = [b - (b-m)\alpha][d - (d-n)\alpha] = bd - [b(d-n) + d(b-m)]\alpha + (b-m)(d-n)\alpha^2 = g(\alpha)$, and the membership function of the decreasing part is

$$\begin{aligned} D(z) &= \frac{[b(d-n)+d(b-m)] - \sqrt{[b(d-n)+d(b-m)]^2 - 4(b-m)(d-n)(bd-z)}}{2(b-m)(d-n)} \\ &= \frac{[b(d-n)+d(b-m)] - \sqrt{[b(d-n)-d(b-m)]^2 + 4(b-m)(d-n)z}}{2(b-m)(d-n)} \end{aligned} \qquad (66)$$

As seen from (61) and (62), the membership function of *D* is not a linear function. An illustrative example for the sum and product is shown in Figure 5.4; *C* is the addition of fuzzy sets *A* and *B*, and *D* is the product of these two fuzzy sets. It is evident that the sum result of triangular fuzzy sets is still triangular, however the product is not.

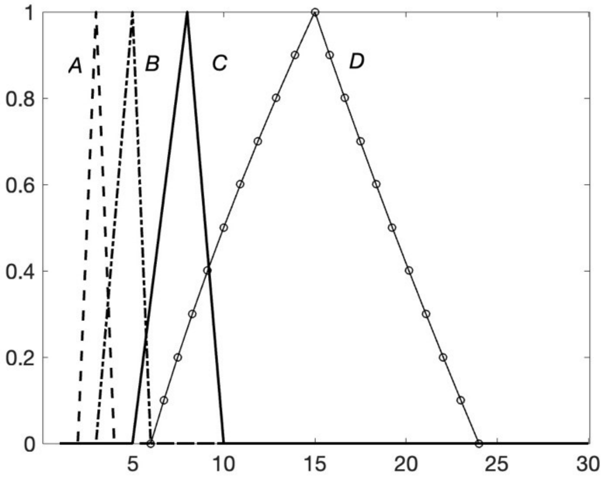


Figure 5.4 Addition and multiplication of triangular fuzzy numbers.

5.2.3 Ranking information granules

Ranking alternatives described by information granules is far more challenging than ranking numbers (which reduces to a simple comparison of numeric values and ordering them in increasing order). Let us consider information granules A_1, A_2, ..., A_N (intervals or fuzzy sets) defined in the interval [0,1]. In what follows, we briefly describe some selected methods by highlighting the underlying rationale behind the ranking technique.

Method #1: The method was proposed in [129]. Assume that we have fuzzy sets A_1, A_2, ..., A_N needed to be ranked. $A_{max}(x)$ is the maximizing set based on the largest value in support of all subsets as below.

$$A_{max}(x) = \begin{cases} x/5, x \in [0,5] \\ 0, \text{ otherwise} \end{cases} \tag{67}$$

Then we rank the fuzzy sets by the intersection point between them and $A_{max}(x)$. The larger the intersection point, the higher the preference of the corresponding fuzzy set is.

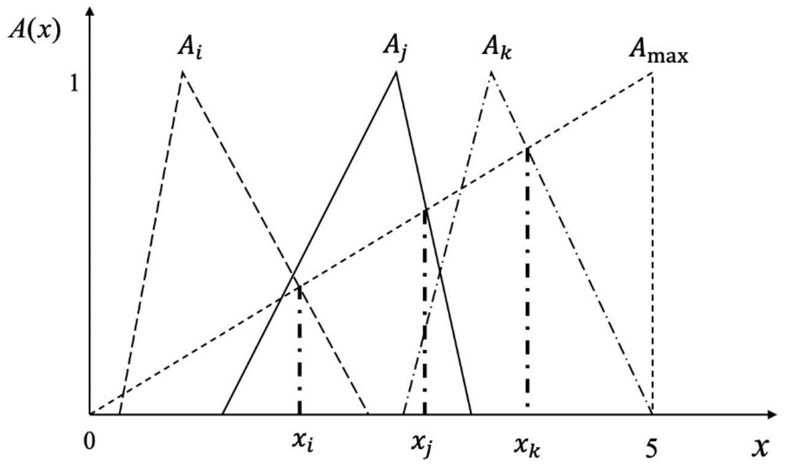


Figure 5.5 Example of ranking.

Method #2: This is a center of gravity method [130]. The fuzzy set A_i is represented by the center of gravity of the membership function, denoting by $CoG(A_i)$, computed as

$$CoG(A_i) = \int_R xA_i(x)dx \Big/ \int_R A_i(x)dx \tag{68}$$

The larger the value of $CoG(A_i)$, the higher preference of A_i.

Method #3: The relative anteriority index in [131] is expressed in the form

$$I(A_i, A_j) = \begin{cases} K(A_i)/(K(A_i) + K(A_j)), if\ K(A_i) + K(A_j) > 0 \\ 1, \qquad\qquad\qquad if\ K(A_i) + K(A_j) = 0. \end{cases} \tag{69}$$

where $K(A_i)$ is computed out by comparing the Hamming distances D_H between fuzzy sets and the maximum (union) of them, i.e.,

$$K(A_i) = D_H(A_i, \max(A_1, A_2, .., A_n)) = \int |A_i(x) - \max(A_1, A_2, .., A_n)\,(x)|dx \tag{70}$$

Given the anteriority index, we compare the two fuzzy numbers A_i, $A_j, i \neq j$, producing the corresponding preference relationship as follows:

$$\begin{cases} A_i \geq A_j, if\ 0 \leq I(A_i, A_j) \leq 0.5\ or\ 0.5 \leq I(A_j, A_i) \leq 1. \\ A_i \leq A_j,\ \ 0.5 \leq I(A_i, A_j) \leq 1\ or\ 0 \leq I(A_j, A_i) \leq 0.5. \end{cases} \tag{71}$$

Method #4: In this approach [132], the ranking is based on the magnitude of fuzzy sets defined as

$$Mag(A_i) = \left(\int_0^1 ({A_i}^L(\alpha) + {A_i}^R(\alpha) + {A_i}^L(1) + {A_i}^R(1))f(\alpha)d\alpha\right)/2 \tag{72}$$

where f is a nonnegative weight function and increasing on [0,1] and α is the α-cut value with $\alpha \in$ [0,1]. ${A_i}^L(\alpha)$and ${A_i}^R(\alpha)$ are the bounded left-continuous non-decreasing function and right-continuous non-increasing function, respectively, over [0,1].

The larger the value of Mag, the higher the preference of the fuzzy number.

Method #5: Median of fuzzy numbers [133] is computed in the following form:

$$\int_{-\infty}^{Med} A_i(x)dx = \int_{Med}^{\infty} A_i(x)dx \tag{73}$$

The larger Med is, the larger the preference of the fuzzy number is.

5.2.4 Information granularity

Information granularity is one of the methods to construct an information granule on the basis of some experimental evidence. In the setting of this study (as we are provided with fuzzy sets), the principle aggregates the fuzzy sets and forms a single information granule of a higher type. Recall that fuzzy sets are information granules of type-1, viz. their degrees of membership are numeric. The result of aggregation of several information granules is an information granule of type-2, for example an interval-valued fuzzy set (where the degrees of membership are intervals) or more generally a fuzzy set whose membership grades are fuzzy sets themselves. An illustration of the resulting information granules is presented in Figure 5.6.

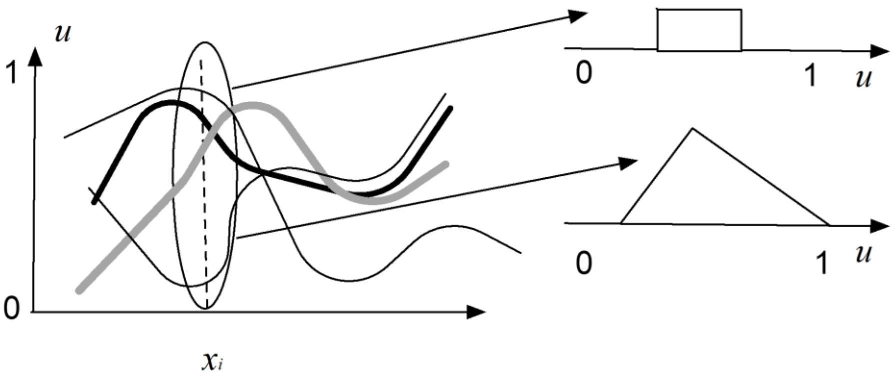


Figure 5.6 Example of resulting information granular (interval-valued or triangular fuzzy set).

To proceed with the detailed algorithm, let us assume that for some alternatives, the degrees of membership are $U = \{u_1, u_2, \dots, u_c\}$ (recall that we have *c* decision-makers). They give rise to an information granule defined in the interval [0,1] being a result of maximization of the product of coverage and specificity. If we consider an interval information granule $A = [a, b]$, the coverage (*cov*) and specificity (*sp*) are defined as follows

$$\begin{aligned} cov(A) &= card(u_k | u_k \in [a,b])/c \\ sp(A) &= 1 - (|b-a|/|\max(u_k) - \min(u_k)|) \end{aligned} \tag{74}$$

where $k = 1,2, \dots, c$.

The convenient construction of the information granule is realized as a two-step procedure:

(i) specification of a numeric representative of *U*, e.g., the average or modal value, denote the result as u^0.

(ii) maximizing the product of coverage and specificity by choosing the lower and upper bound of the interval *a* and *b*.

For a triangular fuzzy set A, the calculations of coverage and specificity are modified because of the use of the membership degrees rather than 0-1 values only. The detailed calculations are completed as follows

$$cov(A) = \sum_{\substack{k=1; \\ u_k: a<u_k<b}}^{c} A(u_k) \tag{75}$$

$$sp(A) = \int_0^1 1 - (|b_\alpha - a_\alpha| / |\max(u_k) - \min(u_k)|) d\alpha \tag{76}$$

5.3 An overall method

In this section, we elaborate on the overall decision-making process. It consists of five main phases.

Step 1: Assuming a certain scale, e.g., 1-9, the reciprocal matrices are elicited producing $R^{[i,j]}$, where $i = 1,2, \ldots, p$; $j = 1,2, \ldots, c$.

Step 2: Consider the *i*-th criterion. For each $R^{[i,j]}$, we solve the eigenvalue problem,

$$R^{[i,j]} \boldsymbol{e}^{[i,j]} = \xi^{[i,j]} \boldsymbol{e}^{[i,j]} \tag{77}$$

and determine $\boldsymbol{e}^{[i,j]}$ which is the eigenvector corresponding to the largest eigenvalue $\xi^{[i,j]}$. The consistency is quantified by computing $\lambda^{[i,j]}$ using (57). The result is used to compute the weight (*weight*) of the corresponding $\boldsymbol{e}^{[i,j]}$

$$weight(\boldsymbol{e}^{[i,j]}) = 1 - ((\lambda^{[i,j]} - \lambda_{\min}) / (\lambda_{\max} - \lambda_{\min})), j = 1,2, \ldots, c \tag{78}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimal and maximal value of $\lambda^{[i,j]}$, respectively, $j = 1,2, \ldots, c$, for the *i*-th criterion.

Step 3: Again considering the i-th criterion, the principle of justifiable granularity is used to determine the interval-valued fuzzy set $E^{[i]}$.

Analyze the eigenvector $\boldsymbol{e}^{[i,j]}, j = 1,2,\dots,c$, with its weight. That is, for each alternative a_s, the corresponding values in $\boldsymbol{e}^{[i,j]}$ compose one-dimensional weighted data. Based on the data, we introduce the principle of justifiable granularity to find the optimal lower bound (a) and upper bound (b) with (70). Since what we input to the information granularity is weighted data, form (70) is replaced by

$$cov = \sum_{\substack{j=1; \\ \boldsymbol{e}^{[i,j]}:a<\boldsymbol{e}^{[i,j]}<b}}^{c} weight(\boldsymbol{e}^{[i,j]})/c$$

$$sp = 1 - \left(|b-a|/\left|\max(\boldsymbol{e}^{[i,j]}) - \min(\boldsymbol{e}^{[i,j]})\right|\right) \tag{79}$$

In the same way, using the principle of justifiable granularity, one can construct $\tilde{E}^{[i]}$ in the form of a triangular fuzzy set.

Step 4: The AHP method is used with regard to pairwise matrices of weights $W^{[i,j]}, j = 1,2,\dots,c$.

$$W^{[i,j]}\boldsymbol{w}^{[i,j]} = \eta^{[i,j]}\boldsymbol{w}^{[i,j]} \tag{80}$$

where $\eta^{[i,j]}$ is the eigenvalue and $\boldsymbol{w}^{[i,j]}$ is the eigenvector. Then for matrix $W^{[i,j]}$, they generate importance vectors of criteria $\boldsymbol{w}^{[i,j]}$ and these weight vectors are aggregated with the principle of justifiable granularity producing interval fuzzy set $W^{[i]}$ or triangular fuzzy set $\widetilde{W}^{[i]}$ for the i-th criterion (the same as before, determine each lower and upper bound by maximizing the product of coverage and specificity).

Step 5: Finally, the vector of preferences of alternatives is constructed by combining interval-valued fuzzy sets $E^{[i]}$ with weights $W^{[i]}$ as described by (56). In addition, for triangular fuzzy sets, the weighted sum is computed by the transformation format of (56) as follows. Then we obtain the result $\tilde{E}$ as below.

$$\tilde{E} = \sum_{i=1\oplus}^{p} (\widetilde{W}^{[i]} \otimes \tilde{E}^{[i]}) \tag{81}$$

In addition, to evaluate the reliability of the results, a process is described in Section 5.2.1.

5.4 Experimental study

In this section, we use the developed methodology to the decision problem of setting up a solar PV power plant by considering five criteria, namely Potential energy production, Environment factors, Safety, Distance from existing transmission line, and Topographical properties and discuss three alternative locations. Following the presented approach, the locations and the weight of the criteria are determined by decision-makers. The elicited matrices of pairwise comparison from experts are shown in Appendix. Twelve decision-makers are involved in the evaluation process.

For $R^{[i,j]}$s, the corresponding eigenvectors $\boldsymbol{e}^{[i,j]}$ along with the values of the inconsistency index λ are reported in Table 5.1.

Table 5.1 The eigenvectors and values of inconsistency index of pairwise comparison matrices.

	Criterion #1	Criterion #2	Criterion #3	Criterion #4	Criterion #5
DM 1	λ: 0.091; $\boldsymbol{e}^{[1,1]}$: [0.07 0.98 0.18]	λ: 0.099; $\boldsymbol{e}^{[2,1]}$: [0.25 0.97 0.08]	λ: 0.001; $\boldsymbol{e}^{[3,1]}$: [0.72 0.68 0.11]	λ: 0.062; $\boldsymbol{e}^{[4,1]}$: [0.33 0.09 0.94]	λ: 0.072; $\boldsymbol{e}^{[5,1]}$: [0.07 0.94 0.34]
DM 2	λ: 0.003; $\boldsymbol{e}^{[1,2]}$: [0.44 0.86 0.27]	λ: 0.016; $\boldsymbol{e}^{[2,2]}$: [0.31 0.92 0.22]	λ: 0.017; $\boldsymbol{e}^{[3,2]}$: [0.25 0.93 0.26]	λ: 0.004; $\boldsymbol{e}^{[4,2]}$: [0.34 0.93 0.11]	λ: 0.025; $\boldsymbol{e}^{[5,2]}$: [0.45 0.89 0.08]
DM 3	λ: 0.029; $\boldsymbol{e}^{[1,3]}$: [0.33 0.53 0.78]	λ: 0.011; $\boldsymbol{e}^{[2,3]}$: [0.91 0.34 0.23]	λ: 0.023; $\boldsymbol{e}^{[3,3]}$: [0.89 0.41 0.20]	λ: 0.007; $\boldsymbol{e}^{[4,3]}$: [0.94 0.26 0.24]	λ: 0.002; $\boldsymbol{e}^{[5,3]}$: [0.89 0.36 0.27]
DM 4	λ: 0.005; $\boldsymbol{e}^{[1,4]}$: [0.44 0.60 0.67]	λ: 0.016; $\boldsymbol{e}^{[2,4]}$: [0.54 0.72 0.43]	λ: 0.071; $\boldsymbol{e}^{[3,4]}$: [0.54 0.66 0.52]	λ: 0.006; $\boldsymbol{e}^{[4,4]}$: [0.66 0.64 0.40]	λ: 0.001; $\boldsymbol{e}^{[5,4]}$: [0.60 0.69 0.41]
DM 5	λ: 0.047; $\boldsymbol{e}^{[1,5]}$: [0.56 0.82 0.15]	λ: 0.016; $\boldsymbol{e}^{[2,5]}$: [0.37 0.92 0.11]	λ: 0.062; $\boldsymbol{e}^{[3,5]}$: [0.09 0.94 0.33]	λ: 0.005; $\boldsymbol{e}^{[4,5]}$: [0.14 0.95 0.26]	λ: 0.091; $\boldsymbol{e}^{[5,5]}$: [0.29 0.95 0.09]
DM 6	λ: 0.005; $\boldsymbol{e}^{[1,6]}$: [0.90 0.41 0.12]	λ: 0.005; $\boldsymbol{e}^{[2,6]}$: [0.97 0.22 0.12]	λ: 0.015; $\boldsymbol{e}^{[3,6]}$: [0.97 0.23 0.09]	λ: 0.040; $\boldsymbol{e}^{[4,6]}$: [0.98 0.08 0.19]	λ: 0.001; $\boldsymbol{e}^{[5,6]}$: [0.90 0.43 0.13]
DM 7	λ: 0.002; $\boldsymbol{e}^{[1,7]}$: [0.62 0.77 0.15]	λ: 0.076; $\boldsymbol{e}^{[2,7]}$: [0.58 0.79 0.21]	λ: 0.000; $\boldsymbol{e}^{[3,7]}$: [0.34 0.92 0.19]	λ: 0.000; $\boldsymbol{e}^{[4,7]}$: [0.60 0.78 0.15]	λ: 0.000; $\boldsymbol{e}^{[5,7]}$: [0.58 0.80 0.16]
DM 8	θ: 0.009; $\boldsymbol{e}^{[1,8]}$: [0.12 0.54 0.83]	λ: 0.000; $\boldsymbol{e}^{[2,8]}$: [0.36 0.35 0.86]	λ: 0.091; $\boldsymbol{e}^{[3,8]}$: [0.36 0.93 0.07]	λ: 0.009; $\boldsymbol{e}^{[4,8]}$: [0.08 0.65 0.76]	λ: 0.037; $\boldsymbol{e}^{[5,8]}$: [0.19 0.49 0.85]
DM 9	λ: 0.093; $\boldsymbol{e}^{[1,9]}$: [0.30 0.58 0.76]	λ: 0.012; $\boldsymbol{e}^{[2,9]}$: [0.19 0.17 0.97]	λ: 0.043; $\boldsymbol{e}^{[3,9]}$: [0.12 0.82 0.55]	λ: 0.002; $\boldsymbol{e}^{[4,9]}$: [0.93 0.17 0.33]	λ: 0.027; $\boldsymbol{e}^{[5,9]}$: [0.30 0.95 0.13]
DM 10	λ: 0.005; $\boldsymbol{e}^{[1,10]}$: [0.16 0.51 0.85]	λ: 0.052; $\boldsymbol{e}^{[2,10]}$: [0.86 0.40 0.31]	λ: 0.009; $\boldsymbol{e}^{[3,10]}$: [0.20 0.35 0.92]	λ: 0.005; $\boldsymbol{e}^{[4,10]}$: [0.95 0.14 0.26]	λ: 0.003; $\boldsymbol{e}^{[5,10]}$: [0.93 0.33 0.14]
DM 11	λ: 0.052; $\boldsymbol{e}^{[1,11]}$: [0.19 0.18 0.97]	λ: 0.040; $\boldsymbol{e}^{[2,11]}$: [0.40 0.91 0.08]	λ: 0.000; $\boldsymbol{e}^{[3,11]}$: [0.51 0.82 0.26]	λ: 0.003; $\boldsymbol{e}^{[4,11]}$: [0.56 0.81 0.17]	λ: 0.000; $\boldsymbol{e}^{[5,11]}$: [0.11 0.56 0.82]
DM 12	λ: 0.006; $\boldsymbol{e}^{[1,12]}$: [0.12 0.55 0.82]	λ: 0.009; $\boldsymbol{e}^{[2,12]}$: [0.38 0.33 0.87]	λ: 0.052; $\boldsymbol{e}^{[3,12]}$: [0.42 0.91 0.08]	λ: 0.005; $\boldsymbol{e}^{[4,12]}$: [0.14 0.88 0.46]	λ: 0.009; $\boldsymbol{e}^{[5,12]}$: [0.19 0.49 0.85]

The vectors $\boldsymbol{e}^{[i,j]}$ are displayed in the form of the bar chart; refer to Figure 5.7.

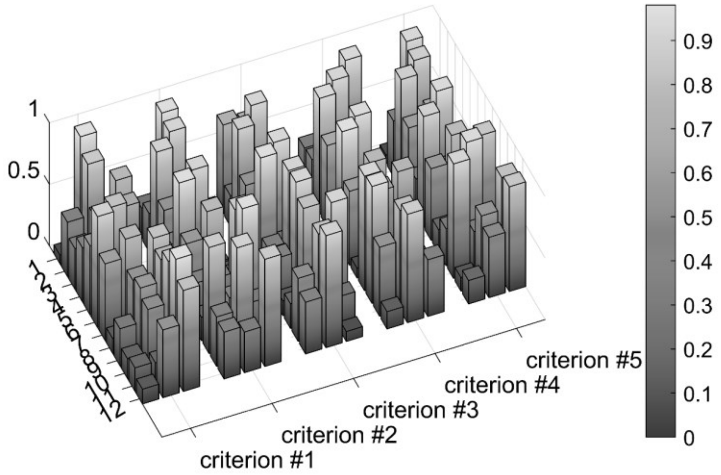


Figure 5.7 Visualization of $\boldsymbol{e}^{[i,j]}$.

The principle of justifiable granularity generates some summarization expressed as interval-valued fuzzy sets $E^{[1]}, E^{[2]}, \dots, E^{[5]}$ or triangular fuzzy sets $\tilde{E}^{[1]}, \tilde{E}^{[2]}, \dots, \tilde{E}^{[5]}$. For the three alternatives under discussion, one obtains the results shown below

$$E^{[1]} = [(0.12{,}0.44)\ (0.41{,}0.60)\ (0.55{,}0.85)],$$

$$E^{[2]} = [(0.31{,}0.54)\ (0.17{,}0.52)\ (0.08{,}0.44)],$$

$$E^{[3]} = [(0.20{,}0.54)\ (0.66{,}0.94)\ (0.08{,}0.30)],$$

$$E^{[4]} = [(0.55{,}0.98)\ (0.60{,}0.95)\ (0.11{,}0.40)],$$

$$E^{[5]} = [(0.53{,}0.93)\ (0.33{,}0.59)\ (0.08{,}0.41)]$$

Likewise, the triangular fuzzy numbers are determined yielding the following membership functions:

$$\tilde{E}^{[1]} = [T(0.11{,}0.12{,}0.72), T(0.51{,}0.51{,}0.98), T(0.57{,}0.85{,}0.85)],$$

$$\tilde{E}^{[2]} = [T(0.31{,}0.31{,}0.66), T(0.17{,}0.35{,}0.44), T(0.07{,}0.12{,}0.53)],$$

$$\tilde{E}^{[3]} = [T(0.09{,}0.15{,}0.64), T(0.59{,}0.93{,}0.94), T(0.07{,}0.10{,}0.61)],$$

$$\tilde{E}^{[4]} = [T(0.30{,}0.95{,}0.98), T(0.41{,}0.93{,}0.93), T(0.15{,}0.15{,}0.61)],$$

$$\tilde{E}^{[5]} = [T(0.07,0.11,0.74), T(0.36,0.36,0.76), T(0.11,0.13,0.48)]$$

Proceeding with the evaluation of weights, we arrive at matrices shown in Appendix. Subsequently, the corresponding results are displayed in Table 5.2; refer also to the bar plot in Figure 5.8.

Table 5.2 Maximal eigenvectors and inconsistency index of pairwise comparison matrix.

	Criterion		Criterion
DM 1	θ: 0.036 $\boldsymbol{w}^{[1]}$: [0.11 0.06 0.51 0.23 0.82]	DM 7	θ: 0.038 $\boldsymbol{w}^{[7]}$: [0.09 0.06 0.87 0.20 0.43]
DM 2	θ: 0.086 $\boldsymbol{w}^{[2]}$: [0.15 0.07 0.32 0.34 0.87]	DM 8	θ: 0.095 $\boldsymbol{w}^{[8]}$: [0.12 0.05 0.81 0.53 0.20]
DM 3	θ: 0.065 $\boldsymbol{w}^{[3]}$: [0.21 0.63 0.25 0.06 0.71]	DM 9	θ: 0.067 $\boldsymbol{w}^{[9]}$: [0.19 0.25 0.46 0.83 0.05]
DM 4	θ: 0.063 $\boldsymbol{w}^{[4]}$: [0.12 0.84 0.50 0.16 0.06]	DM 10	θ: 0.066 $\boldsymbol{w}^{[10]}$: [0.23 0.15 0.10 0.95 0.12]
DM 5	θ: 0.069 $\boldsymbol{w}^{[5]}$: [0.06 0.44 0.33 0.11 0.86]	DM 11	θ: 0.095 $\boldsymbol{w}^{[11]}$: [0.20 0.64 0.35 0.10 0.64]
DM 6	θ: 0.018 $\boldsymbol{w}^{[6]}$: [0.09 0.08 0.50 0.30 0.81]	DM 12	θ: 0.072 $\boldsymbol{w}^{[12]}$: [0.33 0.21 0.23 0.16 0.88]

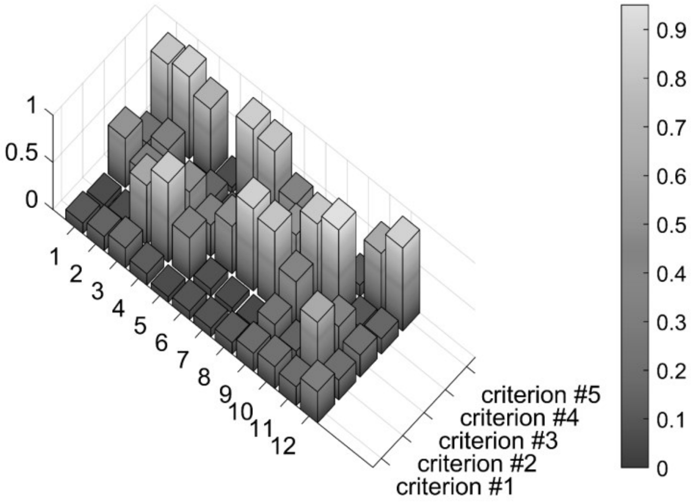


Figure 5.8 Pairwise comparison matrices.

Next, by applying the principle of justifiable granularity, we obtain the interval-valued fuzzy set W, say

$$W = \left[W^{[1]}, W^{[2]}, W^{[3]}, W^{[4]}, W^{[5]}\right]$$

$$= [(0.09,0.14)\ (0.06,0.25)\ (0.46,0.51)\ (0.06,0.31)\ (0.58, 0.88)]$$

and the triangular fuzzy set $\widetilde{W}$ with the entries

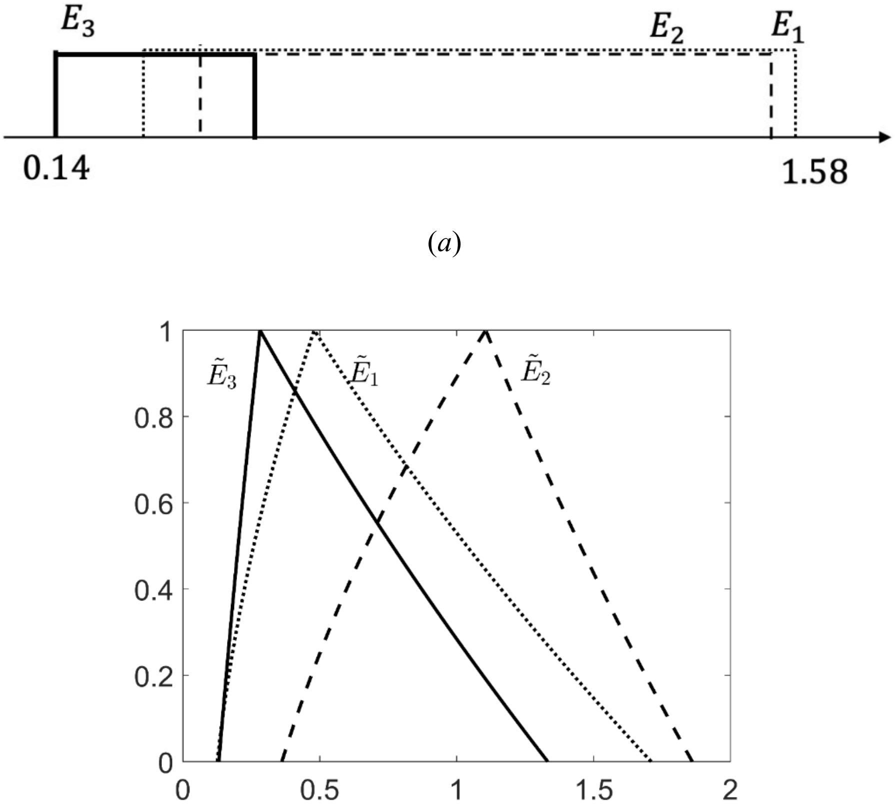
$$\widetilde{W} = \left[\widetilde{W}^{[1]}, \widetilde{W}^{[2]}, \widetilde{W}^{[3]}, \widetilde{W}^{[4]}, \widetilde{W}^{[5]}\right]$$

$$= [T(0.09,0.09,0.25), T(0.05,0.06,0.34), T(0.10,0.50,0.51), T(0.06,0.30,0.35), T(0.64,0.81,0.88)]$$

Finally, based on the above result, we calculate the combination of interval-valued fuzzy sets $E^{[i]}$ with weights $W^{[i]}$ following (56). The vector of preferences of alternatives is resulted as follows, as shown in Figure 5.9(a).

$$E = [E_1, E_2, E_3] = [(0.41,1.58)\ (0.59,1.51)\ (0.14,0.88)]$$

As for triangular fuzzy sets, the weighted sum $\tilde{E}$ is computed following (77), as visualized in Figure 5.9(*b*).



(*a*)

(*b*)

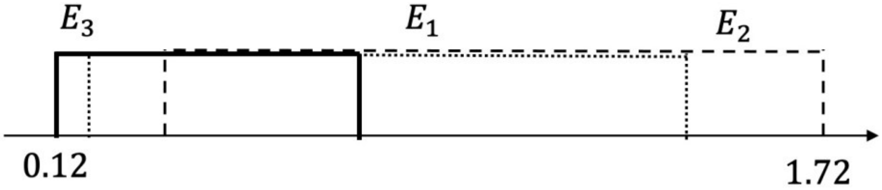
Figure 5.9 Resulting fuzzy sets (*a*) interval-valued (*b*) weighted sum of triangular fuzzy sets for 3 alternatives.

Then, for the interval-valued set E and fuzzy set $\tilde{E}$ in our experiment, according to the ranking methods, the ranking result is shown in Table 5.3. It shows that mostly the alternatives are ranked as (2)>(1)>(3).

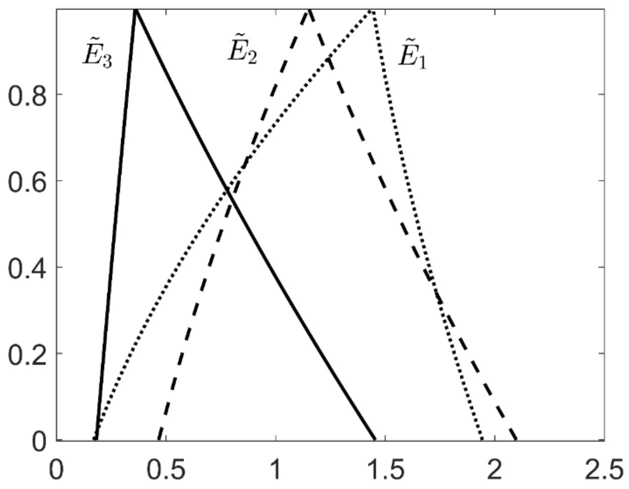
Table 5.3 Ranking results obtained using different ranking methods.

	Ranking result for interval-valued fuzzy sets	Ranking result for fuzzy sets
Method #1	$E_1 > E_2 > E_3$	$\tilde{E}_2 > \tilde{E}_1 > \tilde{E}_3$
Method #2	$E_2 > E_1 > E_3$	$\tilde{E}_2 > \tilde{E}_1 > \tilde{E}_3$
Method #3	$E_1 > E_2 > E_3$	$\tilde{E}_2 > \tilde{E}_1 > \tilde{E}_3$
Method #4	$E_2 > E_1 > E_3$	$\tilde{E}_2 > \tilde{E}_1 > \tilde{E}_3$
Method #5	$E_2 > E_1 > E_3$	$\tilde{E}_2 > \tilde{E}_1 > \tilde{E}_3$

Then considering the sensitivity analysis, we add the noise as mentioned in Section 5.2.1, where the noise coming from the random values in the level of [-4, 4] is taken as an example. Saying [4, -3.7, 2.1, 1.3, -3.1, -2.2, 0.5, 2] is added for corresponding integer values [2, 3, 4, 5, 6, 7, 8, 9] in pairwise matrices. Then we do the same experiments for the generated new matrix, the final interval sets and fuzzy sets are shown in Figure 5.10, and the ranking results are in Table 5.4. Compared to the ranking results in Table 5.3 with the given ranking methods, mostly the Hamming distance between the rankings are 0 except that distance is 2 for Method #1 and #3. It proves the robustness of the results to some extent. Thus, we can conclude that the second location is the best site to install the PV power plant.



(*a*)



(*b*)

Figure 5.10 Ranking of fuzzy sets: (*a*) interval-valued (*b*) weighted sum of triangular fuzzy sets.

The ranking results are reported in Table 5.4.

Table 5.4 Ranking results with different methods.

	Ranking result for interval-valued fuzzy sets	Ranking result for fuzzy sets
Method #1	$E_2 > E_1 > E_3$	$\tilde{E}_1 > \tilde{E}_2 > \tilde{E}_3$
Method #2	$E_2 > E_1 > E_3$	$\tilde{E}_2 > \tilde{E}_1 > \tilde{E}_3$
Method #3	$E_2 > E_1 > E_3$	$\tilde{E}_1 > \tilde{E}_2 > \tilde{E}_3$
Method #4	$E_2 > E_1 > E_3$	$\tilde{E}_2 > \tilde{E}_1 > \tilde{E}_3$
Method #5	$E_2 > E_1 > E_3$	$\tilde{E}_2 > \tilde{E}_1 > \tilde{E}_3$

5.5 Conclusions

In this study, we have established a model for the planning and site selection of renewable energy. Through the combination of AHP, ranking fuzzy sets and the concept of information granularity, from a number of alternatives, the site selection programs are evaluated and sorted from multiple perspectives, and the optimal site is finally selected. The fuzzy analytic hierarchy process adopted in this study maintains the advantages of the traditional analytic hierarchy process, while improving the interpretability.

There are still some further studies worth pursuing in the next step. The analytic hierarchy process is dependent on the suggestions of experts in related fields. In the next step, we can consider evaluating each participant through certain indicators and analyzing each decision matrix, giving appropriate weights to further optimize our decision model.

Chapter 6. Conclusions & future studies

In this thesis, we have proposed some enhancement ideas and algorithms to improve the rule-based models. In this chapter, we briefly summarize the main achievements and identify some interesting future works.

6.1 Conclusions

Although rule-based models, especially fuzzy rule-based models, are commonly used in the research, there is also the evident limitation and shortage existing due to the numeric results and high-dimensional data. Thus, in our proposed studies, we stressed the combination of rule-based models, information granules and the concept of distributed idea. All the experiments demonstrated several essential points:

- The design of distributed fuzzy rule-based models makes an improvement for dealing with large data especially high-dimensional data. The distributed models are a collection of one-dimensional fuzzy local models instead of a monolithic model. Evidently, it achieves lower *RMSE* and lower computing overhead.
- The prediction interval is well known in the statistics area, and its combination with TS model needs more attention. In this study, we developed granular rule-based models with prediction intervals and showed that such augmented models are essential to the quantification of the relevance of the obtained numeric results.
- The analytic hierarchy process is an important tool for a group decision making model. It is used to evaluate the alternatives and the corresponding weights and then the optimal alternative is selected by the weighted preferences (set). In this study, we proposed a concept of information granule injected into the fuzzy analytic hierarchy process, aggregating the eigenvectors and transforming the preferences of alternatives (set) into

interval-valued sets, which realizes the improvement of reliability of decision-making model.

6.2 Future studies

Some further studies worth pursuing can be carried out along several directions. First, some other methods can be tried to form granular rule-based models. For example, the information granularity is used for the local functions to generate the interval-valued functions. Second, several augmentations of clustering techniques can be considered in the context of building rule-based models. One can also concentrate on investigating linkages between standard performance measures of numeric models (such as e.g., *RMSE*) and the quality of granular (interval) prediction results. Third, the role of prediction intervals and their quality can be cast in the framework of coping with ensembles of models. One can easily envision that the use of bagging and boosting of rule-based models (again the topic not studied here in great detail) will exhibit a visible impact on the quality of prediction intervals; a quantification of this effect is worth studying. In the case of a family of models, a way of aggregation of prediction intervals leading to type-2 information granule (referred to as granular prediction intervals) could be investigated. Fourth, the combination of distributed rule-based models and information granules is also an interesting topic. In what follows, we introduce some future studies in detail.

6.2.1 From numeric to granular rule-based model

We aim to designing the granular rule-based models. The difference between this kind of granular model and the granular results in Chapter 4 is that the granular rules are designed on the basis of the granular parameters of the conclusion part instead of using prediction intervals. What is more, the granular models are designed by using an optimization algorithm, e.g., Particle Swarm Optimization (PSO). Firstly, fuzzy rule-based models are used for mapping experiment data to model numeric output, and then granular rule-based models are generated by introducing the information granularity to the parameters that are determined in the fuzzy models. Similarly, the predicted interval will be produced by the granular models.

6.2.2 Analysis and design of distributed granular rule-based models

The distributed architecture of the model provides a sound starting position for further investigations. Let us do the extension for the study of distributed fuzzy rule-based models, e.g., distributed granular rule-based models. Firstly, the data are divided into different feature spaces and then output membership grades. Next, we combine all the membership grades into one vector as the input mapping to an output interval by a granular linkage matrix. For each feature space, the membership grades are generated by triangular membership functions with the cluster centers being determined by uniformly distributed in distributed models in Chapter 3. However, we have introduced an optimization algorithm, PSO algorithm, to optimize the prototypes instead of assigned centers by uniformly distributed. The above process is the design of distributed granular rule-based models; thus, the performance index *RMSE*, coverage and specificity can be used to estimate the quality of the model.

6.2.3 Augmentation of rule-based models—distributed models with a biclustering algorithm

Here we continue the development of distributed models, which are combined with a biclustering algorithm, see Section 2.1. With the biclustering algorithm, the data are separated and then input to individual rule-based models. Then we design each model and generate the corresponding output space; next, they are aggregated with an optimal weight to obtain the model output.

Bibliography

[1] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, Nov. 1964.

[2] 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, Nov. 2017.

[3] J. Kerr-Wilson and W. Pedrycz, "Some new qualitative insights into quality of fuzzy rule-based models," *Fuzzy Sets and Systems*, vol. 307, pp. 29–49, Jan. 2017.

[4] S. Beyhan, "Affine TS fuzzy model-based estimation and control of hindmarsh–rose neuronal model," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2342–2350, Feb. 2017.

[5] C. M. Salgado, J. L. Viegas, C. S. Azevedo, M. C. Ferreira, S. M. Vieira, and J. M. C. Sousa, "Takagi–Sugeno fuzzy modeling using mixed fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1417–1429, Dec. 2016.

[6] Z. Deng, Y. Jiang, K.-S. Choi, F.-L. Chung, and S. Wang, "Knowledge-leverage-based TSK fuzzy system modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 8, pp. 1200–1212, Aug. 2013.

[7] P. Baranyi, "The generalized TP model transformation for T–S fuzzy model manipulation and generalized stability verification," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 934–948, Aug. 2013.

[8] B. Rezaee and M. H. F. Zarandi, "Data-driven fuzzy modeling for Takagi–Sugeno–Kang fuzzy system," *Information Sciences*, vol. 180, no. 2, pp. 241–255, Mar. 2008.

[9] W. Pedrycz and M. Reformat, "Evolutionary fuzzy modeling," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 5, pp. 652–665, Oct. 2003.

[10] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.

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

[12] Q. Zhang and M. Mahfouf, “A hierarchical mamdani-type fuzzy modelling approach with new training data selection and multi-objective optimisation mechanisms: A special application for the prediction of mechanical properties of alloy steels,” *Applied Soft Computing*, vol. 11, no. 2, pp. 2419–2443, Mar. 2011.

[13] 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, Jan. 1975.

[14] W. Pedrycz and M. Reformat, “Rule-based modeling of nonlinear relationships,” *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 2, pp. 256–269, May 1997.

[15] P. Hońko, “Properties of a granular computing framework for mining relational data,” *International Journal of Intelligent Systems*, vol. 32, no. 3, pp. 227–248, Mar. 2017.

[16] V. J. Kok and C. S. Chan, “GrCS: Granular computing-based crowd segmentation,” *IEEE Transactions on Cybernetics*, vol. 47, no. 5, pp. 1157–1168, May 2017.

[17] L. A. Zadeh, “Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic,” *Fuzzy Sets and Systems*, vol. 90, no. 2, pp. 111–127, Sep. 1997.

[18] J. T. Yao, A. v. Vasilakos, and W. Pedrycz, “Granular computing: Perspectives and challenges,” *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1977–1989, Dec. 2013.

[19] Y. Yao, “Perspectives of granular computing,” in *2005 IEEE International Conference on Granular Computing*, 2005, pp. 85-90 Vol. 1.

[20] S. W. Knox, *Machine learning: a concise introduction*, vol. 285. Hoboken, NJ, USA: Wiley & Sons, 2018.

[21] D. Pollard, “A central limit theorem for $ k $-means clustering,” *The Annals of Probability*, vol. 10, no. 4, pp. 919–926, Nov. 1982.

[22] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14, pp. 281–297, Jun. 1967.

[23] K. P. Sinaga and M.-S. Yang, “Unsupervised K-means clustering algorithm,” *IEEE Access*, vol. 8, pp. 80716–80727, 2020.

[24] K. K. Paliwal and V. Ramasubramanian, “Comments on ‘modified K-means algorithm for vector quantizer design,’” *IEEE Transactions on Image Processing*, vol. 9, no. 11, pp. 1964–1967, Nov. 2000.

[25] G. F. Tzortzis and A. C. Likas, “The global kernel k-means algorithm for clustering in feature space,” *IEEE Transactions on Neural Networks*, vol. 20, no. 7, pp. 1181–1194, Jul. 2009.

[26] J. Qin, W. Fu, H. Gao, and W. X. Zheng, “Distributed k-means algorithm and fuzzy c-means algorithm for sensor networks based on multiagent consensus theory,” *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 772–783, Mar. 2017.

[27] J. K. Parker and L. O. Hall, “Accelerating fuzzy c-means using an estimated subsample size,” *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 5, pp. 1229–1244, Oct. 2013.

[28] Y. Zhang, X. Bai, R. Fan, and Z. Wang, “Deviation-sparse fuzzy c-means with neighbor information constraint,” *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 1, pp. 185–199, Nov. 2018.

[29] W. Pedrycz and H. Izakian, “Cluster-centric fuzzy modeling,” *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 6, pp. 1585–1597, Jan. 2014.

[30] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, “Fuzzy c-means algorithms for very large data,” *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 6, pp. 1130–1146, May 2012.

[31] J. C. Dunn, “A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters,” *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, Jan. 1973.

[32] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Boston, MA: Springer US, 1981.

[33] M.-S. Yang, “A survey of fuzzy clustering,” *Mathematical and Computer Modelling*, vol. 18, no. 11, pp. 1–16, Dec. 1993.

[34] A. Baraldi and P. Blonda, “A survey of fuzzy clustering algorithms for pattern recognition. I,” *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 6, pp. 778–785, 1999.

[35] D. Graves and W. Pedrycz, “Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study,” *Fuzzy Sets and Systems*, vol. 161, no. 4, pp. 522–543, Feb. 2010.

[36] D. Gustafson and W. Kessel, “Fuzzy clustering with a fuzzy covariance matrix,” in *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, Jan. 1978, pp. 761–766.

[37] X. Wang, Y. Wang, and L. Wang, “Improving fuzzy c-means clustering based on feature-weight learning,” *Pattern Recognition Letters*, vol. 25, no. 10, pp. 1123–1132, Jul. 2004.

[38] D. C. Park and I. Dagher, “Gradient based fuzzy c-means (GBFCM) algorithm,” in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN’94)*, 1994, pp. 1626–1631.

[39] P. Fazendeiro and J. V. de Oliveira, “A fuzzy clustering algorithm with a variable focal point,” in *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, Jun. 2008, pp. 1049–1056.

[40] J. Zhang and Z. Ma, “Hybrid fuzzy clustering method based on FCM and enhanced logarithmical PSO (ELPSO),” *Computational Intelligence and Neuroscience*, vol. 2020, pp. 1–12, Mar. 2020.

[41] J. A. Hartigan, “Direct clustering of a data matrix,” *J Am Stat Assoc*, vol. 67, no. 337, p. 123, Mar. 1972.

[42] G. Getz, E. Levine, and E. Domany, “Coupled two-way clustering analysis of gene microarray data,” *Proceedings of the National Academy of Sciences*, vol. 97, no. 22, pp. 12079–12084, Oct. 2000.

[43] Chun Tang, Li Zhang, Aidong Zhang, and M. Ramanathan, “Interrelated two-way clustering: an unsupervised approach for gene expression data analysis,” in *Proceedings 2nd Annual IEEE International Symposium on Bioinformatics and Bioengineering (BIBE 2001)*, 2001, pp. 41–48.

[44] Y. Cheng and M. C. George, “Biclustering of expression data,” *Ismb*, vol. 8, pp. 93–103, 2000.

[45] A. Tanay, R. Sharan, and R. Shamir, “Discovering statistically significant biclusters in gene expression data,” *Bioinformatics*, vol. 18, no. Suppl 1, pp. S136–S144, Jul. 2002.

[46] Q. Sheng, Y. Moreau, and B. de Moor, “Biclustering microarray data by Gibbs sampling,” *Bioinformatics*, vol. 19, no. Suppl 2, pp. ii196–ii205, Sep. 2003.

[47] D. E. Duffy and adolfo J. Quiroz, “A permutation-based algorithm for block clustering,” *Journal of Classification*, vol. 8, no. 1, pp. 65–91, Jan. 1991.

[48] R. Tibshirani, T. Hastie, M. Eisen, D. Ross, D. Botstein, and P. Brown, “Clustering methods for the analysis of DNA microarray data,” Stanford,CA, Oct. 1999.

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

[50] J. Kerr-Wilson and W. Pedrycz, “Generating a hierarchical fuzzy rule-based model,” *Fuzzy Sets and Systems*, vol. 381, pp. 124–139, Feb. 2020.

[51] Y. Jin, “Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement,” *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 2, pp. 212–221, Apr. 2000.

[52] J. Li and Q. Liu, “Forecasting of short-term photovoltaic power generation using combined interval type-2 Takagi-Sugeno-Kang fuzzy systems,” *International Journal of Electrical Power & Energy Systems*, vol. 140, p. 108002, Sep. 2022.

[53] E. Soares, P. P. Angelov, B. Costa, M. P. G. Castro, S. Nageshrao, and D. Filev, “Explaining deep learning models through rule-based approximation and visualization,” *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 8, pp. 2399–2407, Aug. 2021.

[54] X. Zhang, L. Zhao, H. Li, and S. Ma, “A novel three-dimensional fuzzy modeling method for nonlinear distributed parameter systems,” *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 489–501, Mar. 2019.

[55] W. Pedrycz, *Granular Computing*. CRC Press, 2018.

[56] L. A. Zadeh, “Fuzzy sets and information granularity,” *Advances in fuzzy set theory and applications*, vol. 11, pp. 3–18, 1979.

[57] W. Pedrycz and W. Homenda, “Building the fundamentals of granular computing: A principle of justifiable granularity,” *Applied Soft Computing*, vol. 13, no. 10, pp. 4209–4218, Oct. 2013.

[58] J. J. Wu, “Semiparametric forecast intervals,” *Journal of Forecasting*, vol. 31, no. 3, pp. 189–228, Apr. 2012.

[59] Y. S. Lee and S. Scholtes, “Empirical prediction intervals revisited,” *International Journal of Forecasting*, vol. 30, no. 2, pp. 217–234, Apr. 2014.

[60] K. Li, R. Wang, H. Lei, T. Zhang, Y. Liu, and X. Zheng, “Interval prediction of solar power using an improved bootstrap method,” *Solar Energy*, vol. 159, pp. 97–112, Jan. 2018.

[61] R. N. Forthofer, E. S. Lee, and M. Hernandez, “Linear Regression,” in *Biostatistics*, Elsevier, 2007, pp. 349–386.

[62] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, 1995, pp. 1942–1948.

[63] W. Jiao, G. Liu, and D. Liu, “Elite particle swarm optimization with mutation,” in *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing*, Oct. 2008, pp. 800–803.

[64] Y. Chen, W. Peng, and M. Jian, “Particle swarm optimization with recombination and dynamic linkage discovery,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 6, pp. 1460–1470, Dec. 2007.

[65] W. Dong and M. Zhou, “A supervised learning and control method to improve particle swarm optimization algorithms,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1135–1148, Sep. 2016.

[66] K. Chen, F. Zhou, and A. Liu, “Chaotic dynamic weight particle swarm optimization for numerical function optimization,” *Knowledge-Based Systems*, vol. 139, pp. 23–40, Jan. 2018.

[67] M. Hasanipanah, R. Naderi, J. Kashir, S. A. Noorani, and A. Zeynali Aaq Qaleh, “Prediction of blast-produced ground vibration using particle swarm optimization,” *Engineering with Computers*, vol. 33, no. 2, pp. 173–179, Jun. 2016.

[68] W. Du, W. Ying, G. Yan, Y. Zhu, and X. Cao, “Heterogeneous strategy particle swarm optimization,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 4, pp. 467–471, Jul. 2016.

[69] D. Wang, D. Tan, and L. Liu, “Particle swarm optimization algorithm: an overview,” *Soft Computing*, vol. 22, no. 2, pp. 387–408, Jan. 2017.

[70] M. R. AlRashidi and M. E. El-Hawary, “A survey of particle swarm optimization applications in electric power systems,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 913–918, Nov. 2008.

[71] N. Delgarm, B. Sajadi, F. Kowsary, and S. Delgarm, “Multi-objective optimization of the building energy performance: A simulation-based approach by means of particle swarm optimization (PSO),” *Applied Energy*, vol. 170, pp. 293–303, May 2016.

[72] M. A. Montes de Oca, T. Stutzle, M. Birattari, and M. Dorigo, “Frankenstein’s PSO: A composite particle swarm optimization algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1120–1132, Oct. 2009.

[73] Z. Zhan, J. Zhang, Y. Li, and Y. Shi, “Orthogonal learning particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 6, pp. 832–847, Sep. 2010.

[74] J. Robinson and Y. Rahmat-Samii, “Particle swarm optimization in electromagnetics,” *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 2, pp. 397–407, Feb. 2004.

[75] Zhi-Hui Zhan, Jun Zhang, Yun Li, and H. S.-H. Chung, “Adaptive particle swarm optimization,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1362–1381, Apr. 2009.

[76] J. B. Park, K. S. Lee, J. R. Shin, and K. Y. Lee, “A particle swarm optimization for economic dispatch with nonsmooth cost functions,” *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 34–42, Jan. 2005.

[77] H. Jiang, Z. He, G. Ye, and H. Zhang, “Network intrusion detection based on PSO-Xgboost model,” *IEEE Access*, vol. 8, pp. 58392–58401, 2020.

[78] J. Cai, H. Wei, H. Yang, and X. Zhao, “A novel clustering algorithm based on DPC and PSO,” *IEEE Access*, vol. 8, pp. 88200–88214, 2020.

[79] T. Cuong-Le, T. Nghia-Nguyen, S. Khatir, P. Trong-Nguyen, S. Mirjalili, and K. D. Nguyen, “An efficient approach for damage identification based on improved machine learning using PSO-SVM,” *Engineering with Computers*, Feb. 2021.

[80] R. E. Bellman, *Adaptive Control Processes*. Princeton University Press, 1961.

[81] M. Lee, K. Kwak, and W. Pedrycz, "An expansion of local granular models in the design of incremental model," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Jul. 2016, pp. 1664–1670.

[82] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures," *Information Sciences*, vol. 181, no. 20, pp. 4340–4360, Oct. 2011.

[83] C. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, pp. 79–82, 2005.

[84] T. O. Kvalseth, "Cautionary note about R 2," *The American Statistician*, vol. 39, no. 4, p. 279, Nov. 1985.

[85] F. Fatemipour and M. R. Akbarzadeh-T, "Dynamic fuzzy rule-based source selection in distributed decision fusion systems," *Fuzzy Information and Engineering*, vol. 10, no. 1, pp. 107–127, Jan. 2018.

[86] A. Segatori, F. Marcelloni, and W. Pedrycz, "On distributed fuzzy decision trees for big data," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 1, pp. 174–192, Feb. 2018.

[87] S. R. Samantaray, K. El-Arroudi, G. Joos, and I. Kamwa, "A fuzzy rule-based approach for islanding detection in distributed generation," *IEEE Transactions on Power Delivery*, vol. 25, no. 3, pp. 1427–1433, Jul. 2010.

[88] M. Marin-Perianu and P. Havinga, "D-FLER – A Distributed Fuzzy Logic Engine for Rule-Based Wireless Sensor Networks," in *Ubiquitous Computing Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 86–101.

[89] J. Cózar, L. delaOssa, and J. A. Gámez, "Learning compact zero-order TSK fuzzy rule-based systems for high-dimensional problems using an Apriori + local search approach," *Information Sciences*, vol. 433–434, pp. 1–16, Apr. 2018.

[90] A. G. Yuksek, H. Arslan, and O. Kaynar, "Comparison of the effects dimensionalty methods in the training of neuro-fuzzy (ANFIS) classifications," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Sep. 2017, pp. 1–9.

[91] R. C. Holte, "Very simple classification rules perform well on most commonly used datasets," *Machine Learning*, vol. 11, no. 1, pp. 63–90, 1993.

[92] J. F. Kolen and T. Hutcheson, "Reducing the time complexity of the fuzzy c-means algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 263–267, Apr. 2002.

[93] F. Fatemipour and M. R. Akbarzadeh-T, "Dynamic Fuzzy Rule-based Source Selection in Distributed Decision Fusion Systems," *Fuzzy Information and Engineering*, vol. 10, no. 1, pp. 107–127, Jan. 2018.

[94] G. A. Seber and Alan J. Lee, *Linear regression analysis*. John Wiley & Sons, 1994.

[95] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control, 5th Edition*. Englewood Cliffs, 1994.

[96] R. D. de Veaux, J. Schumi, J. Schweinsberg, and L. H. Ungar, "Prediction intervals for neural networks via nonlinear regression," *Technometrics*, vol. 40, no. 4, p. 273, Nov. 1998.

[97] D. L. Shrestha and D. P. Solomatine, "Machine learning approaches for estimation of prediction interval for the model output," *Neural Networks*, vol. 19, no. 2, pp. 225–235, Mar. 2006.

[98] A. Khosravi, S. Nahavandi, and D. Creighton, “A prediction interval-based approach to determine optimal structures of neural network metamodels,” *Expert Systems with Applications*, vol. 37, no. 3, pp. 2377–2387, Mar. 2010.

[99] J. T. G. Hwang and A. A. Ding, “Prediction intervals for artificial neural networks,” *J Am Stat Assoc*, vol. 92, no. 438, pp. 748–757, Jun. 1997.

[100] R. Ak, Y. Li, V. Vitelli, and E. Zio, “Adequacy assessment of a wind-integrated system using neural network-based interval predictions of wind power generation and load,” *International Journal of Electrical Power & Energy Systems*, vol. 95, pp. 213–226, Feb. 2018.

[101] H. Chiang, M. Chen, and Z. Wu, “Applying fuzzy petri nets for evaluating the impact of bedtime behaviors on sleep quality,” *Granular Computing*, vol. 3, no. 4, pp. 321–332, Dec. 2018.

[102] M. Tang, X. Chen, W. Hu, and W. Yu, “Generation of a probabilistic fuzzy rule base by learning from examples,” *Information Sciences*, vol. 217, pp. 21–30, Dec. 2012.

[103] N. G. Cogan, M. Y. Hussaini, and S. Chellam, “Uncertainty propagation in a model of dead-end bacterial microfiltration using fuzzy interval analysis,” *Journal of Membrane Science*, vol. 546, pp. 215–224, Jan. 2018.

[104] K. Xu, W. Pedrycz, Z. Li, and W. Nie, “High-accuracy signal subspace separation algorithm based on gaussian kernel soft partition,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 491–499, Jan. 2019.

[105] W. Pedrycz, *Granular Computing*. CRC Press, 2018.

[106] W. Pedrycz, W. Lu, X. Liu, W. Wang, and L. Wang, “Human-centric analysis and interpretation of time series: a perspective of granular computing,” *Soft Computing*, vol. 18, no. 12, pp. 2397–2411, Dec. 2014.

[107] X. Hu, W. Pedrycz, and X. Wang, “Granular Fuzzy Rule-Based Models: A Study in a Comprehensive Evaluation and Construction of Fuzzy Models,” *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1342–1355, Oct. 2017.

[108] G. Hesamian and M. Shams, “Parametric testing statistical hypotheses for fuzzy random variables,” *Soft Computing*, vol. 20, no. 4, pp. 1537–1548, Apr. 2016.

[109] F. Rodríguez, A. Fleetwood, A. Galarza, and L. Fontán, “Predicting solar energy generation through artificial neural networks using weather forecasts for microgrid control,” *Renewable Energy*, vol. 126, pp. 855–864, Oct. 2018.

[110] T. Simioni and R. Schaeffer, “Georeferenced operating-efficiency solar potential maps with local weather conditions – An application to Brazil,” *Solar Energy*, vol. 184, pp. 345–355, May 2019.

[111] D. Liu, Y. Xu, Q. Wei, and X. Liu, “Residential energy scheduling for variable weather solar energy based on adaptive dynamic programming,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 36–46, Jan. 2018.

[112] D. Fernández-Bellon, M. W. Wilson, S. Irwin, and J. O’Halloran, “Effects of development of wind energy and associated changes in land use on bird densities in upland areas,” *Conservation Biology*, vol. 33, no. 2, pp. 413–422, Apr. 2019.

[113] J. Coppes *et al.*, “The impact of wind energy facilities on grouse: a systematic review,” *Journal of Ornithology*, vol. 161, no. 1, pp. 1–15, Jan. 2020.

[114] J. Parisé and T. R. Walker, “Industrial wind turbine post-construction bird and bat monitoring: A policy framework for Canada,” *Journal of Environmental Management*, vol. 201, pp. 252–259, Oct. 2017.

[115] R. May, C. R. Jackson, H. Middel, B. G. Stokke, and F. Verones, “Life-cycle impacts of wind energy development on bird diversity in Norway,” *Environmental Impact Assessment Review*, vol. 90, p. 106635, Sep. 2021.

[116] F. Pollmann, J. Nycander, C. Eden, and D. Olbers, “Resolving the horizontal direction of internal tide generation,” *Journal of Fluid Mechanics*, vol. 864, pp. 381–407, Apr. 2019.

[117] P. Aragonés-Beltrán, F. Chaparro-González, J.-P. Pastor-Ferrando, and A. Pla-Rubio, “An AHP (analytic hierarchy process)/ANP (analytic network process)-based multi-criteria decision approach for the selection of solar-thermal power plant investment projects,” *Energy*, vol. 66, pp. 222–238, Mar. 2014.

[118] M. Tahri, M. Hakdaoui, and M. Maanan, “The evaluation of solar farm locations applying Geographic Information System and Multi-Criteria Decision-Making methods: Case study in southern Morocco,” *Renewable and Sustainable Energy Reviews*, vol. 51, pp. 1354–1362, Nov. 2015.

[119] T. Kaya and C. Kahraman, “Multicriteria decision making in energy planning using a modified fuzzy TOPSIS methodology,” *Expert Systems with Applications*, vol. 38, no. 6, pp. 6577–6585, Jun. 2011.

[120] S. Luthra, K. Govindan, R. K. Kharb, and S. K. Mangla, “Evaluating the enablers in solar power developments in the current scenario using fuzzy DEMATEL: An Indian perspective,” *Renewable and Sustainable Energy Reviews*, vol. 63, pp. 379–397, Sep. 2016.

[121] M. Zoghi, A. Houshang Ehsani, M. Sadat, M. javad Amiri, and S. Karimi, “Optimization solar site selection by fuzzy logic model and weighted linear combination method in arid and semi-arid region: A case study Isfahan-IRAN,” *Renewable and Sustainable Energy Reviews*, vol. 68, pp. 986–996, Feb. 2017.

[122] S. Ozdemir and G. Sahin, “Multi-criteria decision-making in the location selection for a solar PV power plant using AHP,” *Measurement*, vol. 129, pp. 218–226, Dec. 2018.

[123] O. Lindberg, A. Birging, J. Widén, and D. Lingfors, “PV park site selection for utility-scale solar guides combining GIS and power flow analysis: A case study on a Swedish municipality,” *Applied Energy*, vol. 282, p. 116086, Jan. 2021.

[124] R. Davoudabadi, S. M. Mousavi, and V. Mohagheghi, “A new decision model based on DEA and simulation to evaluate renewable energy projects under interval-valued intuitionistic fuzzy uncertainty,” *Renewable Energy*, vol. 164, pp. 1588–1601, Feb. 2021.

[125] H. Karunathilake, K. Hewage, T. Prabatha, R. Ruparathna, and R. Sadiq, “Project deployment strategies for community renewable energy: A dynamic multi-period planning approach,” *Renewable Energy*, vol. 152, pp. 237–258, Jun. 2020.

[126] T. L. Saaty, “A scaling method for priorities in hierarchical structures,” *Journal of Mathematical Psychology*, vol. 15, no. 3, pp. 234–281, Jun. 1977.

[127] Y. Wind and Thomas L. Saaty, “Marketing applications of the analytic hierarchy process,” *Management Science*, vol. 26, no. 7, pp. 641–658, 1980.

[128] T. L. Saaty, “Axiomatic foundation of the analytic hierarchy process,” *Management Science*, vol. 32, no. 7, pp. 841–855, 1986.

[129] “Decisionmaking in the Presence of Fuzzy Variables,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 10, pp. 698–703, Oct. 1976.

[130] R. R. Yager, “A procedure for ordering fuzzy subsets of the unit interval,” *Information Sciences*, vol. 24, no. 2, pp. 143–161, Jul. 1981.

[131] C. de Runz, E. Desjardin, F. Piantoni, and M. Herbin, “Anteriority index for managing fuzzy dates in archæological GIS,” *Soft Computing*, vol. 14, no. 4, pp. 339–344, Feb. 2010.

[132] R. Ezzati, T. Allahviranloo, S. Khezerloo, and M. Khezerloo, “An approach for ranking of fuzzy numbers,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 690–695, Jan. 2012.

[133] Z. Wang and L. Zhang-Westmant, “New ranking method for fuzzy numbers by their expansion center,” *Journal of Artificial Intelligence and Soft Computing Research*, vol. 4, no. 3, pp. 181–187, Jul. 2014.

Appendix

The pairwise comparison matrix of three alternatives coming from 12 decision makers under different criteria.

The 12 comparison matrices of alternatives following criterion #1: Potential energy production

$$R^{[1,1]}=\begin{bmatrix}1 & \frac{1}{9} & \frac{1}{4}\\ 9 & 1 & 8\\ 4 & \frac{1}{8} & 1\end{bmatrix}, R^{[1,2]}=\begin{bmatrix}1 & \frac{5}{9} & \frac{3}{2}\\ \frac{9}{5} & 1 & \frac{7}{2}\\ \frac{2}{3} & \frac{2}{7} & 1\end{bmatrix}, R^{[1,3]}=\begin{bmatrix}1 & \frac{4}{5} & \frac{1}{3}\\ \frac{5}{4} & 1 & \frac{6}{7}\\ 3 & \frac{7}{6} & 1\end{bmatrix}, R^{[1,4]}=\begin{bmatrix}1 & \frac{4}{5} & \frac{1}{3}\\ \frac{5}{4} & 1 & 1\\ \frac{5}{3} & 1 & 1\end{bmatrix}$$

$$R^{[1,5]}=\begin{bmatrix}1 & \frac{1}{2} & 5\\ 2 & 1 & 4\\ \frac{1}{5} & \frac{1}{4} & 1\end{bmatrix}, R^{[1,6]}=\begin{bmatrix}1 & 2 & 8\\ \frac{1}{2} & 1 & 3\\ \frac{1}{8} & \frac{1}{3} & 1\end{bmatrix}, R^{[1,7]}=\begin{bmatrix}1 & \frac{3}{4} & \frac{9}{2}\\ \frac{4}{3} & 1 & 5\\ \frac{2}{9} & \frac{1}{5} & 1\end{bmatrix}, R^{[1,8]}=\begin{bmatrix}1 & \frac{1}{4} & \frac{1}{8}\\ 4 & 1 & \frac{3}{4}\\ 8 & \frac{4}{3} & 1\end{bmatrix}$$

$$R^{[1,9]}=\begin{bmatrix}1 & \frac{1}{3} & \frac{3}{5}\\ 3 & 1 & \frac{1}{2}\\ \frac{5}{3} & 2 & 1\end{bmatrix}, R^{[1,10]}=\begin{bmatrix}1 & \frac{1}{3} & \frac{1}{6}\\ 3 & 1 & \frac{2}{3}\\ 6 & \frac{3}{2} & 1\end{bmatrix}, R^{[1,11]}=\begin{bmatrix}1 & \frac{3}{2} & \frac{1}{7}\\ \frac{2}{3} & 1 & \frac{1}{4}\\ 7 & 4 & 1\end{bmatrix}, R^{[1,12]}=\begin{bmatrix}1 & \frac{1}{5} & \frac{1}{6}\\ 5 & 1 & \frac{3}{5}\\ 6 & \frac{5}{3} & 1\end{bmatrix}$$

The comparison matrices following criterion #2: Environment factors

$$R^{[2,1]}=\begin{bmatrix}1 & \frac{1}{6} & 5\\ 6 & 1 & 8\\ \frac{1}{5} & \frac{1}{8} & 1\end{bmatrix}, R^{[2,2]}=\begin{bmatrix}1 & \frac{2}{5} & \frac{7}{6}\\ \frac{5}{2} & 1 & 5\\ \frac{6}{7} & \frac{1}{5} & 1\end{bmatrix}, R^{[2,3]}=\begin{bmatrix}1 & \frac{7}{3} & \frac{9}{2}\\ \frac{3}{7} & 1 & \frac{5}{4}\\ \frac{2}{9} & \frac{4}{5} & 1\end{bmatrix}, R^{[2,4]}=\begin{bmatrix}1 & \frac{5}{8} & \frac{3}{2}\\ \frac{8}{5} & 1 & \frac{7}{5}\\ \frac{2}{3} & \frac{5}{7} & 1\end{bmatrix}$$

$$R^{[2,5]}=\begin{bmatrix}1 & \frac{1}{3} & 4\\ 3 & 1 & 7\\ \frac{1}{4} & \frac{1}{7} & 1\end{bmatrix}, R^{[2,6]}=\begin{bmatrix}1 & 4 & 9\\ \frac{1}{4} & 1 & \frac{5}{3}\\ \frac{1}{9} & \frac{3}{5} & 1\end{bmatrix}, R^{[2,7]}=\begin{bmatrix}1 & \frac{1}{2} & 4\\ 2 & 1 & \frac{5}{2}\\ \frac{1}{4} & \frac{2}{5} & 1\end{bmatrix}, R^{[2,8]}=\begin{bmatrix}1 & 1 & \frac{3}{7}\\ 1 & 1 & \frac{2}{5}\\ \frac{7}{3} & \frac{5}{2} & 1\end{bmatrix}$$

$$R^{[2,9]} = \begin{bmatrix} 1 & \frac{4}{3} & \frac{1}{6} \\ \frac{3}{4} & 1 & \frac{1}{5} \\ 6 & 5 & 1 \end{bmatrix}, R^{[2,10]} = \begin{bmatrix} 1 & 3 & 2 \\ \frac{1}{3} & 1 & \frac{7}{4} \\ \frac{1}{2} & \frac{4}{7} & 1 \end{bmatrix}, R^{[2,11]} = \begin{bmatrix} 1 & \frac{1}{3} & 7 \\ 3 & 1 & 9 \\ \frac{1}{7} & \frac{1}{9} & 1 \end{bmatrix}, R^{[2,12]} = \begin{bmatrix} 1 & 1 & \frac{1}{2} \\ 1 & 1 & \frac{1}{3} \\ 2 & 3 & 1 \end{bmatrix}$$

The comparison matrices based on criterion #3: Safety

$$R^{[3,1]} = \begin{bmatrix} 1 & 1 & 7 \\ 1 & 1 & 6 \\ \frac{1}{7} & \frac{1}{6} & 1 \end{bmatrix}, R^{[3,2]} = \begin{bmatrix} 1 & \frac{2}{9} & \frac{7}{6} \\ \frac{9}{2} & 1 & 3 \\ \frac{6}{7} & \frac{1}{3} & 1 \end{bmatrix}, R^{[3,3]} = \begin{bmatrix} 1 & \frac{8}{3} & \frac{7}{2} \\ \frac{3}{8} & 1 & \frac{5}{2} \\ \frac{2}{7} & \frac{2}{5} & 1 \end{bmatrix}, R^{[3,4]} = \begin{bmatrix} 1 & \frac{5}{9} & \frac{3}{2} \\ \frac{9}{5} & 1 & \frac{7}{8} \\ \frac{2}{3} & \frac{8}{7} & 1 \end{bmatrix}$$

$$R^{[3,5]} = \begin{bmatrix} 1 & \frac{1}{7} & \frac{1}{5} \\ 7 & 1 & 4 \\ 5 & \frac{1}{4} & 1 \end{bmatrix}, R^{[3,6]} = \begin{bmatrix} 1 & 5 & 9 \\ \frac{1}{5} & 1 & 3 \\ \frac{1}{9} & \frac{1}{3} & 1 \end{bmatrix}, R^{[3,7]} = \begin{bmatrix} 1 & \frac{3}{8} & \frac{7}{4} \\ \frac{8}{3} & 1 & 5 \\ \frac{4}{7} & \frac{1}{5} & 1 \end{bmatrix}, R^{[3,8]} = \begin{bmatrix} 1 & \frac{1}{4} & 8 \\ 4 & 1 & 9 \\ \frac{1}{8} & \frac{1}{9} & 1 \end{bmatrix}$$

$$R^{[3,9]} = \begin{bmatrix} 1 & \frac{1}{5} & \frac{1}{6} \\ 5 & 1 & 2 \\ 6 & \frac{1}{2} & 1 \end{bmatrix}, R^{[3,10]} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} \\ 2 & 1 & \frac{1}{3} \\ 4 & 3 & 1 \end{bmatrix}, R^{[3,11]} = \begin{bmatrix} 1 & \frac{3}{5} & 2 \\ \frac{5}{3} & 1 & 3 \\ \frac{1}{2} & \frac{1}{3} & 1 \end{bmatrix}, R^{[3,12]} = \begin{bmatrix} 1 & \frac{1}{3} & 7 \\ 3 & 1 & 8 \\ \frac{1}{7} & \frac{1}{8} & 1 \end{bmatrix}$$

The comparison matrices coming from criterion #4: Distance from existing transmission line

$$R^{[4,1]} = \begin{bmatrix} 1 & 5 & \frac{1}{4} \\ \frac{1}{5} & 1 & \frac{1}{7} \\ 4 & 7 & 1 \end{bmatrix}, R^{[4,2]} = \begin{bmatrix} 1 & \frac{1}{3} & \frac{7}{2} \\ 3 & 1 & 8 \\ \frac{2}{7} & \frac{1}{8} & 1 \end{bmatrix}, R^{[4,3]} = \begin{bmatrix} 1 & 4 & \frac{7}{2} \\ \frac{1}{4} & 1 & \frac{5}{4} \\ \frac{2}{7} & \frac{4}{5} & 1 \end{bmatrix}, R^{[4,4]} = \begin{bmatrix} 1 & \frac{7}{6} & \frac{3}{2} \\ \frac{6}{7} & 1 & \frac{9}{5} \\ \frac{2}{3} & \frac{5}{9} & 1 \end{bmatrix}$$

$$R^{[4,5]} = \begin{bmatrix} 1 & \frac{1}{6} & \frac{1}{2} \\ 6 & 1 & 4 \\ 2 & \frac{1}{4} & 1 \end{bmatrix}, R^{[4,6]} = \begin{bmatrix} 1 & 9 & 7 \\ \frac{1}{9} & 1 & \frac{1}{3} \\ \frac{1}{7} & 3 & 1 \end{bmatrix}, R^{[4,7]} = \begin{bmatrix} 1 & \frac{3}{4} & 4 \\ \frac{4}{3} & 1 & 5 \\ \frac{1}{4} & \frac{1}{5} & 1 \end{bmatrix}, R^{[4,8]} = \begin{bmatrix} 1 & \frac{1}{9} & \frac{1}{8} \\ 9 & 1 & \frac{3}{4} \\ 8 & \frac{4}{3} & 1 \end{bmatrix}$$

$$R^{[4,9]} = \begin{bmatrix} 1 & 5 & 3 \\ \frac{1}{5} & 1 & \frac{1}{2} \\ \frac{1}{3} & 2 & 1 \end{bmatrix}, R^{[4,10]} = \begin{bmatrix} 1 & 6 & 4 \\ \frac{1}{6} & 1 & \frac{1}{2} \\ \frac{1}{4} & 2 & 1 \end{bmatrix}, R^{[4,11]} = \begin{bmatrix} 1 & \frac{3}{4} & 3 \\ \frac{4}{3} & 1 & 5 \\ \frac{1}{3} & \frac{1}{5} & 1 \end{bmatrix}, R^{[4,12]} = \begin{bmatrix} 1 & \frac{1}{7} & \frac{1}{3} \\ 7 & 1 & \frac{7}{4} \\ 3 & \frac{4}{7} & 1 \end{bmatrix}$$

The comparison matrices following criterion #5: Topographical properties

$$R^{[5,1]} = \begin{bmatrix} 1 & \frac{1}{9} & \frac{1}{7} \\ 9 & 1 & 4 \\ 7 & \frac{1}{4} & 1 \end{bmatrix}, R^{[5,2]} = \begin{bmatrix} 1 & \frac{2}{5} & 7 \\ \frac{5}{2} & 1 & 9 \\ \frac{1}{7} & \frac{1}{9} & 1 \end{bmatrix}, R^{[5,3]} = \begin{bmatrix} 1 & \frac{7}{3} & \frac{7}{2} \\ \frac{3}{7} & 1 & \frac{5}{4} \\ \frac{2}{7} & \frac{4}{5} & 1 \end{bmatrix}, R^{[5,4]} = \begin{bmatrix} 1 & \frac{5}{6} & \frac{3}{2} \\ \frac{6}{5} & 1 & \frac{8}{5} \\ \frac{2}{3} & \frac{5}{8} & 1 \end{bmatrix}$$

$$R^{[5,5]} = \begin{bmatrix} 1 & \frac{1}{5} & 5 \\ 5 & 1 & 7 \\ \frac{1}{5} & \frac{1}{7} & 1 \end{bmatrix}, R^{[5,6]} = \begin{bmatrix} 1 & 2 & 7 \\ \frac{1}{2} & 1 & 3 \\ \frac{1}{7} & \frac{1}{3} & 1 \end{bmatrix}, R^{[5,7]} = \begin{bmatrix} 1 & \frac{3}{4} & \frac{7}{2} \\ \frac{4}{3} & 1 & 5 \\ \frac{2}{7} & \frac{1}{5} & 1 \end{bmatrix}, R^{[5,8]} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{6} \\ 2 & 1 & \frac{3}{4} \\ 6 & \frac{4}{3} & 1 \end{bmatrix}$$

$$R^{[5,9]} = \begin{bmatrix} 1 & \frac{1}{4} & 3 \\ 4 & 1 & 6 \\ \frac{1}{3} & \frac{1}{6} & 1 \end{bmatrix}, R^{[5,10]} = \begin{bmatrix} 1 & 3 & 6 \\ \frac{1}{3} & 1 & \frac{5}{2} \\ \frac{1}{6} & \frac{2}{5} & 1 \end{bmatrix}, R^{[5,11]} = \begin{bmatrix} 1 & \frac{1}{5} & \frac{1}{7} \\ 5 & 1 & \frac{2}{3} \\ 7 & \frac{3}{2} & 1 \end{bmatrix}, R^{[5,12]} = \begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{4} \\ 3 & 1 & \frac{1}{2} \\ 4 & 2 & 1 \end{bmatrix}$$

The pairwise comparison matrix of criteria coming from 12 decision makers.

$$W^{[1]}=\begin{bmatrix}1 & 2 & \frac{1}{5} & \frac{1}{3} & \frac{1}{6}\\ \frac{1}{2} & 1 & \frac{1}{7} & \frac{1}{5} & \frac{1}{9}\\ 5 & 7 & 1 & 3 & \frac{1}{2}\\ 3 & 5 & \frac{1}{3} & 1 & \frac{1}{5}\\ 6 & 9 & 2 & 5 & 1\end{bmatrix}, W^{[2]}=\begin{bmatrix}1 & 3 & \frac{1}{2} & \frac{1}{3} & \frac{1}{5}\\ \frac{1}{3} & 1 & \frac{1}{7} & \frac{1}{3} & \frac{1}{9}\\ 2 & 7 & 1 & \frac{1}{2} & \frac{1}{2}\\ 3 & 3 & 2 & 1 & \frac{1}{5}\\ 5 & 9 & 2 & 5 & 1\end{bmatrix}, W^{[3]}=\begin{bmatrix}1 & \frac{1}{3} & \frac{4}{5} & 4 & \frac{1}{3}\\ 3 & 1 & 5 & 7 & \frac{1}{2}\\ \frac{5}{4} & \frac{1}{5} & 1 & 6 & \frac{2}{5}\\ \frac{1}{4} & \frac{1}{7} & \frac{1}{6} & 1 & \frac{1}{8}\\ 3 & 2 & \frac{5}{2} & 8 & 1\end{bmatrix}$$

$$W^{[4]}=\begin{bmatrix}1 & \frac{1}{8} & \frac{1}{5} & \frac{4}{5} & 3\\ 8 & 1 & 2 & 7 & 9\\ 5 & \frac{1}{2} & 1 & 5 & 6\\ \frac{5}{4} & \frac{1}{7} & \frac{1}{5} & 1 & 5\\ \frac{1}{3} & \frac{1}{9} & \frac{1}{6} & \frac{1}{5} & 1\end{bmatrix}, W^{[5]}=\begin{bmatrix}1 & \frac{1}{6} & \frac{1}{5} & \frac{1}{3} & \frac{1}{8}\\ 6 & 1 & 3 & 5 & \frac{1}{3}\\ 5 & \frac{1}{3} & 1 & 3 & \frac{1}{5}\\ 3 & \frac{1}{5} & \frac{1}{3} & 1 & \frac{1}{7}\\ 8 & 3 & 5 & 7 & 1\end{bmatrix}, W^{[6]}=\begin{bmatrix}1 & 1 & \frac{1}{5} & \frac{1}{3} & \frac{1}{9}\\ 1 & 1 & \frac{1}{7} & \frac{1}{5} & \frac{1}{7}\\ 5 & 7 & 1 & 2 & \frac{1}{2}\\ 3 & 5 & \frac{1}{2} & 1 & \frac{1}{3}\\ 9 & 7 & 2 & 3 & 1\end{bmatrix}$$

$$W^{[7]}=\begin{bmatrix}1 & 2 & \frac{1}{8} & \frac{1}{3} & \frac{1}{5}\\ \frac{1}{2} & 1 & \frac{1}{9} & \frac{1}{4} & \frac{1}{7}\\ 8 & 9 & 1 & 5 & 3\\ 3 & 4 & \frac{1}{5} & 1 & \frac{1}{3}\\ 5 & 7 & \frac{1}{3} & 3 & 1\end{bmatrix}, W^{[8]}=\begin{bmatrix}1 & 4 & \frac{1}{8} & \frac{1}{6} & \frac{1}{2}\\ \frac{1}{4} & 1 & \frac{1}{8} & \frac{1}{7} & \frac{1}{7}\\ 8 & 8 & 1 & 3 & 5\\ 6 & 7 & \frac{1}{3} & 1 & 5\\ 2 & 7 & \frac{1}{5} & \frac{1}{5} & 1\end{bmatrix}, W^{[9]}=\begin{bmatrix}1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{6} & 7\\ 2 & 1 & \frac{2}{5} & \frac{1}{4} & 6\\ 3 & \frac{5}{2} & 1 & \frac{1}{2} & 7\\ 6 & 4 & 2 & 1 & 9\\ \frac{1}{7} & \frac{1}{6} & \frac{1}{7} & \frac{1}{9} & 1\end{bmatrix}$$

$$W^{[10]}=\begin{bmatrix}1 & 3 & 2 & \frac{1}{5} & \frac{7}{6}\\ \frac{1}{3} & 1 & 2 & \frac{1}{9} & 2\\ \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{7} & 1\\ 5 & 9 & 7 & 1 & 7\\ \frac{6}{7} & \frac{1}{2} & 1 & \frac{1}{7} & 1\end{bmatrix}, W^{[11]}=\begin{bmatrix}1 & \frac{1}{2} & \frac{1}{5} & 3 & \frac{1}{4}\\ 2 & 1 & 3 & 4 & \frac{3}{2}\\ 5 & \frac{1}{3} & 1 & \frac{5}{2} & \frac{1}{2}\\ \frac{1}{3} & \frac{1}{4} & \frac{2}{5} & 1 & \frac{1}{9}\\ 4 & \frac{2}{3} & 2 & 9 & 1\end{bmatrix}, W^{[12]}=\begin{bmatrix}1 & 2 & 1 & 3 & \frac{1}{3}\\ \frac{1}{2} & 1 & 1 & 2 & \frac{1}{5}\\ 1 & 1 & 1 & \frac{2}{3} & \frac{2}{5}\\ \frac{1}{3} & \frac{1}{2} & \frac{3}{2} & 1 & \frac{1}{7}\\ 3 & 5 & \frac{5}{2} & 7 & 1\end{bmatrix}$$