University of Alberta

Fuzzy Modeling through Granular Computing

by

Sharifah Sakinah Syed Ahmad

A thesis submitted to the Faculty of Graduate Studies and Research
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

Fall 2012

Edmonton, Alberta

Dedication

Sjarifa Mariam Sayid Abdoellah, Syed Hisham Syed Zain
Syed Ahmad Hassan, and Syed Muhammad Sajad

Abstract

In this thesis, we introduce a concept of feature reduction, in which the reduction is guided by a criterion of structure retention. In other words, the features forming the reduced space are selected in such a way that the original structure present in the highly dimensional space is retained in the reduced space to the highest possible extent.

Then, we provide a new method for complexity reduction in fuzzy modeling through the feature and data reduction approach. Data and feature reduction activities are advantageous to fuzzy models in terms of both the effectiveness of their construction and the interpretation of the resulting models. The formation of a subset of meaningful features and a subset of essential instances is discussed in the context of fuzzy rule-based models. The reduction problem is combinatorial in its nature and, as such, calls for the use of advanced optimization techniques. Here, we use the technique of Particle Swarm Optimization as an optimization vehicle for forming a subset of features and data to design a fuzzy model.

Next, we develop a comprehensive design process of granular fuzzy rule-based systems. These constructs arise as a result of a structural compression of fuzzy rule-based systems in which a subset of originally existing rules is retained. Because of the reduced subset of the originally existing rules, the remaining rules are made more abstract (general) by expressing their conditions in the form of granular fuzzy sets, hence the name of granular fuzzy rule-based systems emerging during the compression of the rule bases. The design of these systems dwells upon an important mechanism of allocation of information granularity using which the granular fuzzy rules are formed.

Finally, we introduce a new framework of *Takagi-Sugeno-Kang* fuzzy systems via the concept of information granulation. In spite of the standard TSK model being used, the representation of the antecedent part is numeric (coming from structure identification process via fuzzy clustering). We consider a concept of granular antecedent and consequent parts that generalize the numeric representation of the firing strength for the predicted output, in this way; helps capture more details about the fuzzy system.

Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful

My deepest gratitude is to Allah the Almighty, who always showed me the right path and gave me the courage to tread upon that path.

Special appreciation goes to my supervisor, Prof. Witold Pedrycz for his valuable suggestions and guidance in my research direction, generous sharing of his professional and personal experiences.

I am grateful to my examining committee, Professor Marek Reformat, Professor Petr Musilek, Professor Aminah Robinson Fayek, and Professor Yiyu Yao for taking their precious time to serve on my committee.

I am indebted to my family members especially, my mother, Sjarifa Mariam for supporting and encouraging me to pursue this degree. Last but not least, I would like to thank my husband, Syed Hisham and my sons Syed Ahmad Hassan and Syed Muhammad Sajad for their understanding, constant support and love during this research period.

I acknowledge the Ministry of Higher Education (MOHE) and Universiti Teknikal Melaka Malaysia (UTeM) for providing scholarships to pursue doctoral studies at the University of Alberta, Canada.

Finally, I thank all those who have helped me directly or indirectly in the successful completion of my thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF NOMENCLATURE

Symbol	Description
D	Data Space
$\mathbf{X}= \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$	Dataset
n	Dimensionality of data
M	Number of data
F	Feature Space
c	Number of clusters
$\mathbf{V}=\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c\}$	Set of prototypes
$U= [u_{ik}]$ for i=1,2,..,c , k=1,2,…M	Partition matrix
$\| . \|$	Euclidean norm
N	Number of rules
G	Granular representation

1.Introduction and Motivation

In fuzzy modeling, the two main approaches for generating the rules rely on knowledge acquisition from human experts and knowledge discovery from data (Pedrycz and Gomide 2007). However, the consistent and complete expert knowledge for designing the fuzzy model is not always available or the cost of deriving such expert knowledge may be too high. On the other hand, knowledge discovery from data (data-driven fuzzy models) can enable one to identify the structure and the parameters of fuzzy models directly form numerical data (Zhang and Mahfouf 2011). In recent years, data-driven fuzzy modeling has becoming an urging issue whose relevance is growing together with the technological progress that permit the manipulation of massive amounts of data (Castellano, et al. 2005, Jin 2000).

In this thesis, the fuzzy model computation framework is based on the concept of fuzzy if-then rules (fuzzy rule-based system). In the literature, various fuzzy rule-based models were proposed (Mamdani and Assilian 1975, Tsukamoto 1979, Takagi and Sugeno 1985). The differences among them involve a format of the conclusion part. The two most well-known approaches encountered are the *Mamdani* fuzzy model (Mamdani and Assilian 1975) and the *Takagi-Sugeno* fuzzy model (Takagi and Sugeno 1985). Both models are realized as "if-then" rules and share the same antecedent structure. However, the differences between them include the structure of the consequent part and the defuzzification process. In the *Takagi-Sugeno* model, the consequent part includes a function instead of a fuzzy set, as in the *Mamdani* model. In terms of the defuzzification process, only the *Mamdani* model needs the defuzzification process because the output is a fuzzy set. In contrast, the output for the *Takagi-Sugeno* model is numeric.

The construction of a fuzzy rules-based system exhibits two important objectives: (1) to achieve an acceptable approximation for the problem based on the accuracy of the resulting model and (2) to reduce the complexity of the fuzzy rules by reducing the total number of rules. Achieving both objectives at once is difficult because of their conflicting nature (Baranyi and Yam 2000, Alcala, Alcala-Fdez, et al. 2006, Mikut, Jakel and Groll 2005). In order to obtain good approximation we typically need to use more rules. In contrast, to achieve a comprehensible and interpretable model, a smaller number of rules are required. There is no obvious way to balance both objectives. Most of the research uses a method that focuses on only accuracy and neglects the other component. Therefore, complexity reduction is becoming a pertinent research topic for the fuzzy rule-based system.

On the other hand, for high-dimensional problems in a continuous input-output domain requires a significant number of fuzzy rules. In many cases, the ability to develop models efficiently is hampered by the dimensionality of the input space as well as the number of data. If we are concerned with rule-based models, the high dimensionality of the feature space, along with the topology of the rules, gives rise to the curse of dimensionality (Pedrycz and Gomide 2007). The number of rules increases exponentially and is equal to P^n, where n is the number of features (variables), and P stands for the number of fuzzy sets defined for each feature. In addition, creating the training data is one of the important steps in designing the fuzzy models. A large number of data significantly impacts data-driven fuzzy models. It is well known that using more training data will not always improve the performance of the models. A large number of training data has significant implications on model's capabilities because it is quite likely that many noisy data are present in the training data set. This may mislead the fuzzy model or cause it to over fit the data (Zhang and Mahfouf 2011). Thus, the effectiveness of the fuzzy models relies on the quality of the training data.

Motivated by our findings about this topic, the thesis presents an alternative approach to the construction of fuzzy models, with a better tradeoff between accuracy and complexity. Here, we implement the integration of data and feature selection for fuzzy modeling. Intuitively, the data and feature reduction activities are advantageous to fuzzy models in terms of both the effectiveness of their construction and the interpretation of the resulting models. Therefore, the use of such activities deserves particular attention. The formation of a subset of meaningful features and a subset of essential instances is discussed in the context of fuzzy rule-based models. In contrast to the existing studies, which focus mainly on feature selection (or a reduction of the input space), we propose here that a reduction has to involve both the data and the features to become efficient in the design of a fuzzy model. The reduction problem is combinatorial and, as such, calls for the use of advanced optimization techniques. In this study, we use the technique of Particle Swarm Optimization (PSO) as an optimization vehicle for forming a subset of features and data (instances) to design a fuzzy model. In order to deal with a high dimensional search space that involves both features and instances, we implement a cooperative version of the PSO, along with a clustering mechanism for forming a partition of the overall search space.

In Chapter 5, we move to the next stage of improving the efficiency of the fuzzy model by using the concept of Granular Computing (Zadeh 1997). Here, we deal with the reducing the complexity of existing the fuzzy rule-based system. Therefore, we introduce an alternative method for the complexity reduction of the original fuzzy rule-based system by using granular realization in the form of interval-valued fuzzy sets. The motivation behind using this granular

generalization of fuzzy rules is to reduce the number of the original fuzzy rules. The underlying intuitively appealing idea is that to compensate for the reduction in the size of the rule base, we need to make the fuzzy set in the remaining rules more abstract viz. more granular (Bargiela and Pedrycz 2003). The granular fuzzy rule originates from the main concept in information granularity, which emphasizes the generality of the granular representation. The reduced set of rules is composed of granular fuzzy sets; via fuzzy sets whose membership grades are described in terms of information granules, say intervals, fuzzy sets or probability density function.

In Chapter 6, we deal with the *Takagi-Sugeno* (TS) fuzzy model. We introduce and develop a comprehensive framework of the granular Takagi-Sugeno fuzzy model. This research is motivated by our desire to use the concept of Granular Computing in processing the fuzzy rules-based system. Here, we develop a new way of generating the fuzzy rules via the concept of information granularity. The standard TS fuzzy model has several limitations, especially when the dimensionality of the system is large. The method of constructing the fuzzy model by using information granulation offers an immediate advantage as it supports meaningful ways of striking a balance between the complexity and accuracy of the fuzzy model.

1.1 Objectives

The main objectives of this thesis are:

1. To develop a comprehensible framework of the feature reduction guided by the idea of structure preservation
2. To develop a data-driven fuzzy modeling via the integration of feature and instance selection.
3. To introduce a new particle representation of the cooperative Particle Swarm Optimization method for feature and data selection based on the concept of information granulation
4. To introduce a concept and develop a comprehensive design process of granular fuzzy rule-based systems.
5. To introduce a concept and develop a comprehensive framework the Granular Takagi-Sugeno fuzzy model.

1.2 Contributions

In Figure 1-1 we illustrate the overall structure of the thesis. The findings of our research contribute to the field of fuzzy modeling in the following ways:

1. Feature reduction through structure retention is guided by the idea of structure preservation: given the structure of the data in the original space, we reduce the feature space so that the original structure is retained to a significant extent. The structure in the data is determined through fuzzy clustering, and the abilities of the structure obtained in this way are quantified by using the granulation-degranulation criterion.
2. We develop a comprehensive framework to construct a data-driven fuzzy modeling framework for a high dimensional dataset, which is capable of generating a rule-base automatically from numerical data. Here, we integrate the concept of feature selection and data selection together in a unified form to further reduce the fuzzy models. In this regard, the PSO technique is applied in order to search for the best subset of data. In order to increase the effectiveness of the PSO techniques, we introduce a new implementation of Cooperative PSO method based on the information granulation approach. The proposed approach allows the user to choose the predetermined fraction of variables and data that can be used to construct the fuzzy models.
3. We introduce the concept of a Granular Fuzzy Rule-based System, which directly results from the compactification of the rule-based system and provides a more compact, interpretable yet highly representative collection of rules than the previous system provide. These constructs arise as a result of a structural compression of fuzzy rule-based systems in which only a subset of originally existing rules is retained. The development of the granular rule based system comprises two influential and intertwined phases: the selection of a subset of the rules and the formation of the optimal allocation of information granularity.
4. We introduce the concept of a Granular Takagi-Sugeno fuzzy model and develop a comprehensive framework for its construction processes. The construction of this model is based on the granular realization of the information granules (prototypes) with its corresponding granular firing strength (membership grades). This framework improves the generalization of the constructed model, whose outputs are also realized with the use of information granularity. In order to increase the effectiveness of the representation of the information granules, we introduce several protocols for finding the best allocation of the information granularity.

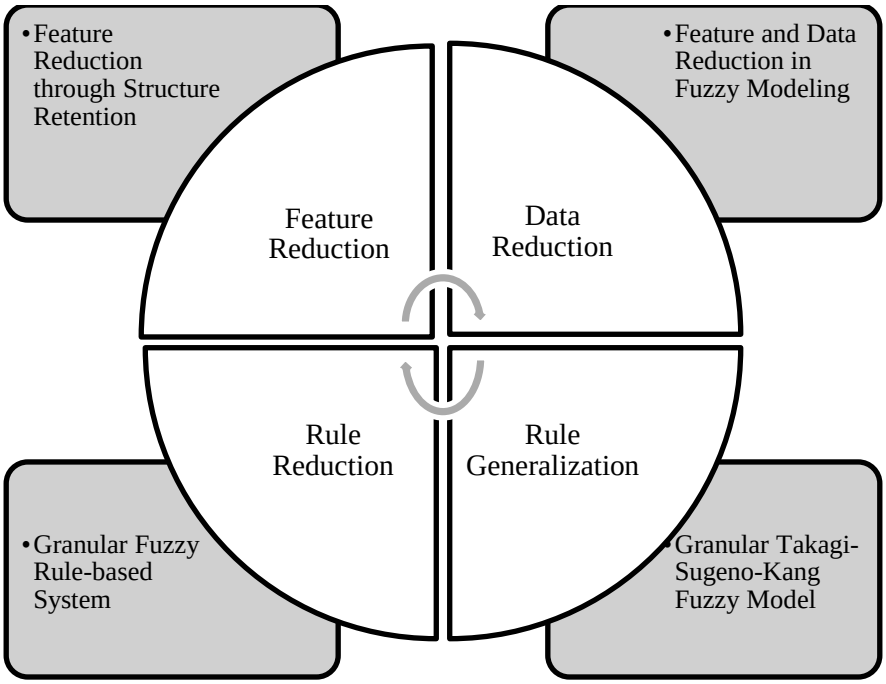


Figure 1-1: The overall structure of the thesis

1.3 Dissertation organization

The dissertation is organized as follows:

In Chapter 2, we briefly review the background knowledge that we implemented in our research. First, we discuss the fundamentals concept of information granulation. Next, we provide detailed introduction to fuzzy clustering. Then we explain the main fuzzy modeling approaches namely fuzzy *Takagi-Sugeno* (TS) fuzzy model and *Mamdani* fuzzy model. Finally, we elaborate in detail the optimization methods implemented in this research.

In Chapter 3, we introduce a concept of feature reduction via information granulation, in which the reduction process is guided by the criterion of structure retention. Fuzzy clustering (and FCM, in particular) is used as an algorithmic vehicle for information granulation.

In Chapter 4, we introduce a data reduction approach for fuzzy modeling. The data and feature reduction is advantageous to fuzzy models in terms of both the effectiveness of their construction and the interpretation of the resulting models. In this chapter, we discuss the formation of a subset of meaningful features and a subset of essential instances in the context of fuzzy rule-based models.

In Chapter 5, we develop a comprehensive design process for granular fuzzy rule-based systems. These constructs arise from the structural compression of fuzzy rule-based systems, in which a subset of the original rules is retained. Because of the reduced subset of the originally existing rules, the remaining rules are made more abstract by expressing their conditions in the form of granular fuzzy sets (hence the name of granular fuzzy rule-based systems)

In Chapter 6, we introduce a concept and develop a comprehensive design process for Granular Takagi-Sugeno systems. The construction of this model is based on the concept of information granularity, in which the firing strength and predicted outputs are non-numeric.

Finally in Chapter 7, we draw conclusions from our work; we review our contributions and consider future work in the area of fuzzy modeling.

2.Background & Literature Review

This chapter briefly describes some basic elements and fundamental concept of Granular Computing. In Section 2.2, we elaborate on fuzzy clustering algorithm. In Section 2.3, we discuss the fundamental aspects of fuzzy modeling. Finally, in Section 2.4, we elaborate in detail the population-based methods used for solving the optimization problems.

2.1 Granular Computing

In this section, we discuss the concept of Granular Computing proposed by Zadeh (1997). The fundamental concept in granular computing is the concept of information granulation. The granulation of a universe involves grouping similar elements into granules to form a coarse-grained view of the universe. Granular Computing fully acknowledges a notion of variable granularity whose range could cover detailed numeric entities and very abstract and general information granules. This advantage is important when dealing with incomplete, uncertain, or vague information regarding the problem at hand. In some problems, although detailed information is available, the concept of information granulation can be used in order to produce an efficient and practical solution (Yao 2005). Moreover, the use of Granular Computing simplifies the original problem. Obviously, the cost of acquiring precise information is high compared to the cost of using coarse-grained information.

The fundamental issues in Granular Computing are the construction of information granules, the representation of each granule, and, finally, the utilization of granules for solving the problem. The information granules are based on the available knowledge.

2.1.1 Fundamental concept of Granular Computing

The best approach for understanding the concept of any method is to study its fundamental operations and their basic elements. The basic elements of Granular Computing are called granules, and the operation on the granules is called granulation.

The definition of a granule in *Merriam-Webster's Dictionary* is "a small particle; especially one of numerous particles forming a larger unit". This definition is similar to that given by the granular computing communities. Granules are composed of finer granules that are drawn together by distinguishability, similarity, and functionality (Zadeh 1996). Granules are the subsets, classes, objects, clusters, and elements of a universe. Granules can be measured in different levels based on their complexity, abstraction and size. The lowest level of granules is

composed of the basic particles of the particular model that is used. For example, if we consider an article as a granule, the lowest level of granules is the words or letters.

2.1.2 Representation of information granulation

Information granules can be represented by using several frameworks such as, fuzzy sets, rough sets, and shadowed sets. Fuzzy sets have been defined as a collection of objects with membership values between 0 and 1. These values express the degree to which each object is compatible with the properties or features distinctive to the collection (Pedrycz and Gomide 1998). A family of fuzzy sets defined in **X** is denoted by **F(X)**.

The rough sets emphasize the roughness of the description of a given concept **X** when being realized in terms of the indiscernibility relation provided in advance (Pawlak 1982). The roughness of the description of **X** is manifested in terms of its lower and upper approximation of a certain rough set. A family of rough sets defined in **X** is denoted by **R(X).**

Shadowed sets offer descriptions of information granules by distinguishing among the elements, that fully belong to the concept, are excluded from it and whose belongingness is completely unknown (Pedrycz 2005). The information granules are formally described as a mapping of X: $\mathbf{X} \rightarrow \{1, 0, [0, 1]\}$, where the elements with the membership quantified as the entire [0, 1] interval are used to describe a shadow of the construct. Given the nature of the mapping here, shadowed sets can be used as a granular description of fuzzy sets where the shadow is used to localize partial membership values, which in a fuzzy set are distributed over the entire universe of discourse. A family of shadowed sets defined in **X** is denoted by **S(X)**.

2.1.3 Description of information granules

The granular representation can easily be illustrated by using new evidence of X in terms of the elements in the information granules. Let us assume given a finite vocabulary of information granules $\mathbf{A} = \{A_1, A_2, ..., A_c\}$, where the information granules A_i can be presented by using any form given before. The relationship between X and A_i can be described in terms of the coincidence (overlap) of these two and the inclusion of X in some information granules. The following are the two concepts that can be used to describe the coincidence and the inclusion between X and A (Bargiela and Pedrycz 2003):

The degree of coincidence (overlap) of X in A_i (possibility):

$$\text{Poss}(X, A_i) = \sup_{x \in X}[X(x) t A_i(x)] \qquad (2\text{-}1)$$

The degree of inclusion of X in A_i (necessity):

$$Nec(X , A_i) = inf_{x \in X}[(1\text{-}X(x)) s A_i(x)] \quad (2\text{-}2)$$

where s and t are some t-norm and t-conorm, respectively. The relationship between the possibility measurement and the necessity measurement is Nec $(X,A_i) \leq$ Poss (X,A_i). These two descriptions can be used to identify the relationship between two information granules. In this thesis, the above formulas are modified, so that the supremum and infimum operations are replaced by the maximum and minimum operations.

2.1.4 The development of information granules through fuzzy clustering: Fuzzy C-Means (FCM)

Information granulation can be obtained in various ways depending on the type of problem and the type of data available. For example, granules can be obtained manually through expert interviews or automatically by clustering techniques. Expert interviews are useful when the designer wants to obtain information granules that reflect subjective perceptions about concepts associated with the problem that is being solved. In contrast, clustering techniques are used when the information granules must account for information contained in experimental data. For this thesis, we used fuzzy sets as the formal framework for building the information granules. The construction of information granules can be carried out by means of fuzzy clustering (Yu and Pedrycz 2009, Pedrycz and Vokovich 2001). A fuzzy clustering algorithm can be described as a function that accepts a set of observations and returns as a set of prototypes together with a partition matrix. Here, the number of clusters represents the number of information granules. The details of the construction of the fuzzy clustering algorithm are explained in Section 2.2.

2.2 Fuzzy clustering

Clustering is one of the popular approaches for exploring data and has been addressed in many application domains (Oliveira and Pedrycz 2007, Pedrycz 2005). The process of data clustering can be divided into two categories: hierarchical clustering and partitional clustering. The hierarchical clustering methods produce a graphic representation of data. The construction of hierarchical clustering is done in two ways: bottom-up and top-down. The partitional clustering is concerned with building partitions (clusters) of data sets based on some objective function. The process of grouping the patterns can be represented by using hard clustering or fuzzy clustering. The hard clustering approach allocates each pattern into a single cluster. In contrast, in the fuzzy

clustering approach, each pattern is assigned to several clusters with the corresponding degree of membership.

In this section, we describe in detail the procedure of using the most widely used fuzzy clustering method: the Fuzzy C-Means (FCM) (Bezdek, Ehrlich and Full 1984, Bezdek 1981). The objective of using fuzzy clustering is to reveal the structure of the data set and to present it in a comprehensible and readable format.

2.2.1 Fuzzy C-Mean algorithm

The FCM algorithm starts with the desired number of clusters and initial prediction for each membership grade. Therefore, all data points have a membership grade for each cluster. By iteratively updating the membership grades, as well as the prototypes (cluster centers) of the data point, the algorithm aims to guide the cluster center to the optimal location in the data space. Given a data set $\mathbf{X}=\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ in the n-dimensional space $\Re^n$ with n data vectors (features) $\mathbf{x}_i \in \Re^n, i = 1,2,...,n$. This clustering procedure is based on minimizing the objective function described as follows:

$$J = \sum_{i=1}^{c}\sum_{k=1}^{M} u_{ik}^{m} \left\| \mathbf{x}_k - \mathbf{v}_i \right\|^2 \tag{2-3}$$

where u_{ik} represents the membership grade of data $\mathbf{x}_k$ in i-th cluster, where i=1,2,…c and k=1,2,…,M. $\mathbf{v}_i$ is the i-th prototypes, ||.|| is a distance between the data and prototype, and m is constant. Here, the parameter m, (m>1) controls the fuzziness of the resulting partition.
The partition matrix U satisfies the following conditions:

(a) $0 \le u_{ik} \le 1$ (2-4)

(b) $\sum_{i=1}^{c} u_{ik} = 1$

(c) $0 < \sum_{i=1}^{M} u_{ik} < M$

i=1, 2, …, c and k=1, 2, …, M.

The minimization of the objective function, J is completed with the respect to partition matrix, U and the prototypes **V** of the clusters. The optimization task involves two processes. The first one involves the minimization of J with respect to the constraints given the requirement (2-4(b)) which holds for each data point $\mathbf{x}_k$. Here, the use of Langrage multipliers converts the problem into its constraint-free version. The augmented functional is formulated as the follows (Pedrycz, 2005),

$$V = \sum_{i=1}^{c} u_{ik}^{m} d_{ik}^{2} + \lambda\left(\sum_{i=1}^{c} u_{ik}^{m} - 1\right) \tag{2-5}$$

where λ denotes the Langrage multiplier and $d_{ik}^{2} = \left\|\mathbf{x}_k - \mathbf{v}_i\right\|^2$.

The following are the necessary condition for minimum of V for k=1, 2, …, M,

$$\frac{\partial V}{\partial u_{st}} = 0 \qquad \frac{\partial V}{\partial \lambda} = 0 \tag{2-6}$$

where s=1,2,…, c and t=1,2, …, M.

Now we calculate the derivative of V with respect of the elements of partition matrix. By making it equal to 0, we obtain

$$\frac{\partial V}{\partial u_{st}} = m u_{st}^{m-1} d_{st}^{2} + \lambda = 0 \tag{2-7}$$

and

$$u_{st} = \left(-\frac{\lambda}{m}\right)^{1/(m-1)} \frac{1}{\left(d_{st}\right)^{2/(m-1)}} \tag{2-8}$$

Given the normalization condition $\sum_{j=1}^{c} u_{jt} = 1$ we have

$$\left(-\frac{\lambda}{m}\right)^{1/(m-1)} \sum_{j=1}^{c} \frac{1}{\left(d_{jt}\right)^{2/(m-1)}} = 1 \tag{2-9}$$

Then we compute

$$\left(-\frac{\lambda}{m}\right)^{1/(m-1)} = \sum_{j=1}^{c} \frac{1}{\dfrac{1}{\left(d_{jt}\right)^{2/(m-1)}}} \tag{2-10}$$

Next, we insert the above expression into (2-8) and obtain the successive entries of the partition matrix

$$u_{st}(x) = \frac{1}{\displaystyle\sum_{j=1}^{c} \left(\frac{d_{st}^{2}}{d_{jt}^{2}}\right)^{2/(m-1)}} \tag{2-11}$$

The optimization of the prototypes $\mathbf{v}_i$ is carried out assuming that the weighted Euclidean distance between the data and the prototypes is

$$\left\|\mathbf{x}_k - \mathbf{v}_i\right\|^2 = \sum_{j=1}^{n} \frac{(x_{kj} - v_{ij})^2}{\sigma_j^2} \tag{2-12}$$

where σ_j^2 is the sample variance of the j-th coordinate (variable) of the feature space.

The minimum objective function, J computed to $\mathbf{v}_s$ yields the system of linear equations

$$2\sum_{k=1}^{M} u_{sk}^{m}(\mathbf{x}_k - \mathbf{v}_s) = 0 \tag{2-13}$$

Thus

$$\mathbf{v}_s = \frac{\sum_{k=1}^{M} u_{sk}^{m}\mathbf{x}_{ik}}{\sum_{k=1}^{N} u_{sk}^{m}} \tag{2-14}$$

The objective function is minimized when the data are close to the prototype of their cluster and are assigned high membership grades; then the low membership grades are assigned to data that are far from the center. The membership grades represent the degree that the data belong to a specific cluster.

The FCM clustering starts from some random allocation of data (a certain randomly initialized partition matrix). We carry out an iterative process involving successive computations of prototypes and the partition matrix. The stopping criterion used serves to compare the partition matrices produced in two successive iterations of the FCM, say, U (iter+1) and U (iter), and if the distance between them || U (iter+1) –U (iter) || does exceed a certain threshold (ε), it will stops the computing. The threshold value is usually in the range of 10^{-3}-10^{-5}.The distance function must take into account the biggest change in the partition matrix; that is,

$$\|U(\text{iter}+1) - U(\text{iter})\| = \max_{i,k} | u_{ik}(\text{iter}+1) - u_{ik}(\text{iter}) | \tag{2-15}$$

Summarizing the FCM clustering procedure consists of the following steps:

Input: data points $\mathbf{X}= \{\mathbf{x}_1, \mathbf{x}_2,\ldots,\mathbf{x}_n\}$

number of centers (clusters) to c , $2 \le c \le M$

fuzzification coefficient, m

threshold, ε

Initialize partition matrix U(0) using random values between 0 and 1

t ←0

repeat

for i=1:c do

Compute the prototypes

$$\mathbf{v}_i(t) = \frac{\sum_{k=1}^{M} u_{ik}^{m}(t)\mathbf{x}_{ik}}{\sum_{k=1}^{N} u_{ik}^{m}(t)}$$

for i=1:c do

for k=1:M do

Update partition matrix

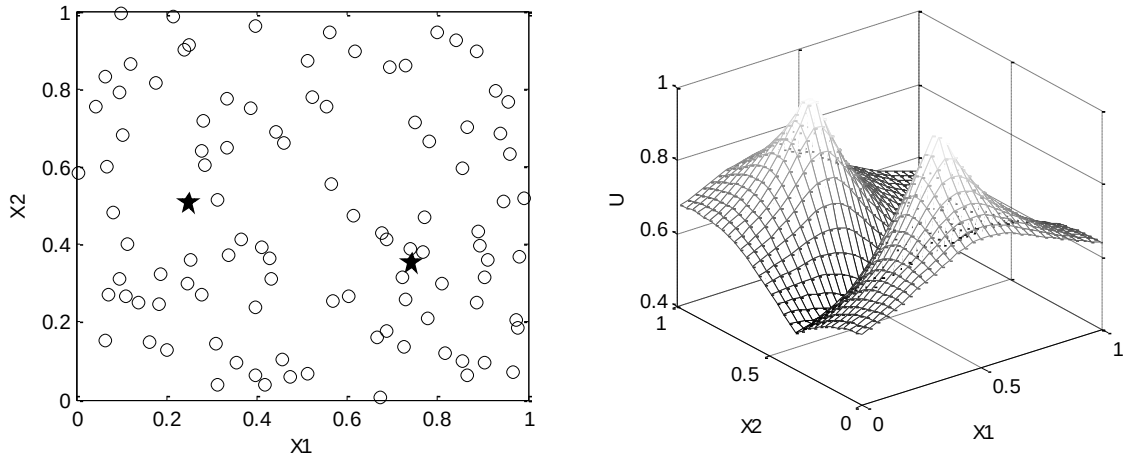
$$u_{ik}(t+1) = \frac{1}{\sum_{j=1}^{c}\left(\frac{\| \mathbf{x}_k - \mathbf{v}_i(t) \|}{\| \mathbf{x}_k - \mathbf{v}_j(t) \|}\right)^{2/(m-1)}}$$

t←t+1

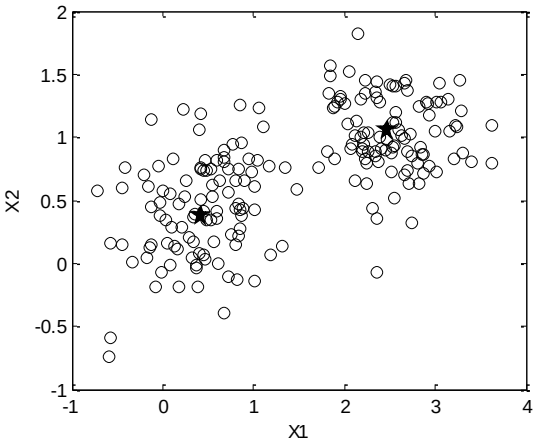
until $\|U(t+1) - U(t)\| \leq \varepsilon$

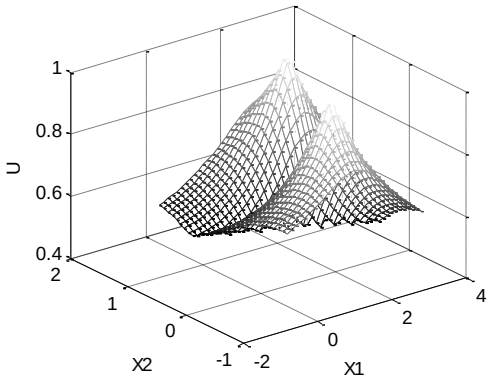
return U,V

The elements of the fuzzy partition matrix describe the membership of the feature x_k belonging to the cluster i. Based on these membership grades, we can easily separate the patterns that are typical of the cluster (as they have membership grade close to 1) from the borderline data (Zadeh 1997, Bezdek 1981). The following examples are used to illustrate how Fuzzy C-Means clustering works with a two-dimensional dataset. Figure 2-1 show the optimal position of the centers (prototypes) based on the data points given, with the corresponding membership grades of the fuzzy clusters obtained by using FCM clustering.

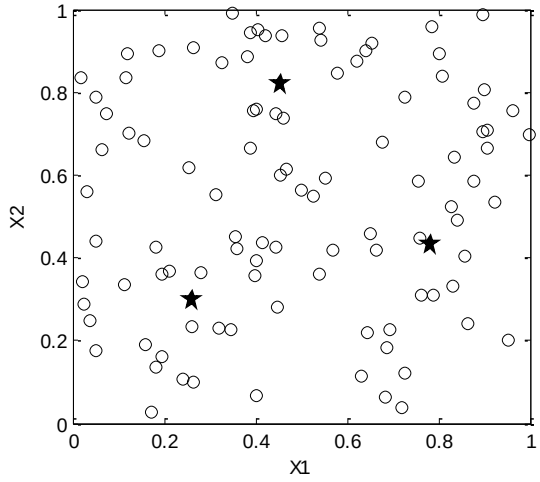


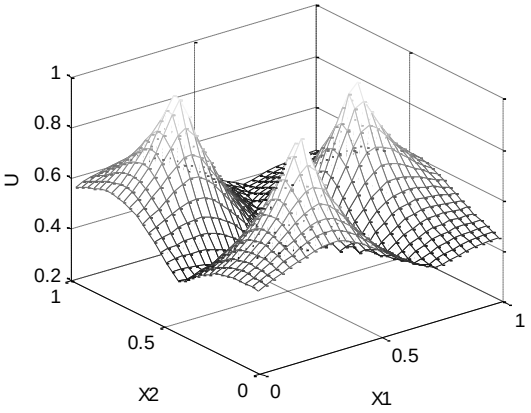
(a)



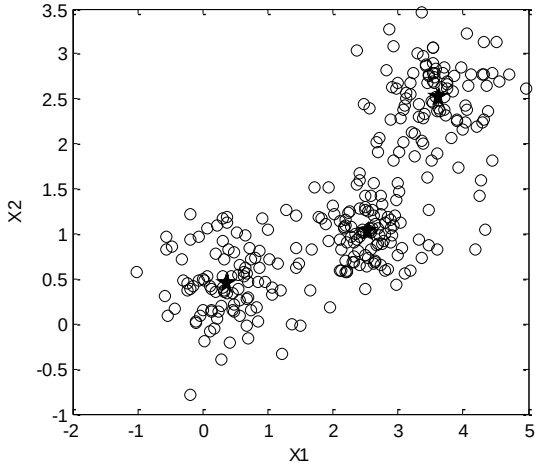


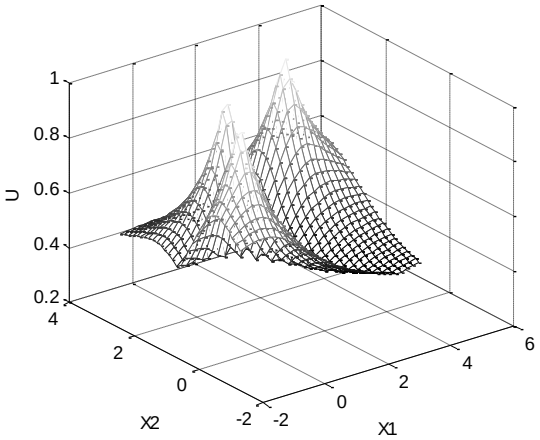
(b)





(c)





(d)

Figure 2-1: Clustering data in the product space with the use of FCM; (a) & (b) number of clusters=2, and (c) & (d) number of clusters=3

2.3 Fuzzy modeling

Fuzzy model is suitable for solving a non-linear problem, especially when the underlying physical relationships are not easy to understand. The framework for a fuzzy model is based on the concepts of fuzzy sets, fuzzy rules-based system, and fuzzy reasoning. The designed fuzzy models are capable of conducting perceptual uncertainties, such as the ambiguity and vagueness involved in a real-world problem (Pedrycz 1984, Zadeh 1973).

The earliest application of fuzzy modeling was by Zadeh (1973) who constructed fuzzy models directly from an expert's knowledge of the system. Here, an expert provides a description of a system by using linguistic terms, which are then represented within the approximate reasoning framework. However, this method is limited by the nature of the knowledge extracted from the expert. When the information which is normally provided by an expert is unavailable, yet the input-output data are present, the structure of the model can be generated by using various methods such as clustering techniques (Tsekouras and Bafas 2003, Nayak and Sudheer 2008). The clustering algorithm partitions the input-output space in order to find the membership function of the model (Kim, et al. 1997, Tsekouras, et al. 2005, Gomez-Skarmeta and Delgado 1999).

The general configuration of a fuzzy model (Pedrycz and Gomide 2007), illustrated in Figure 2-2, is composed of five generic modules: the input interface, rule base, database, inference engine, and output interface.

The input interface obtains the input values and maps them into suitable fuzzy sets represented by the membership functions (MFs). This process is also called the fuzzification. The value of the membership function is defined by using a number between 0 and 1, where 0 implies the total absence of membership, 1 implies the complete membership, and any value in between implies the partial membership in a fuzzy set. The Gaussian membership function and some others commonly used membership functions are shown in Figure 2-3.

The rule base is the cornerstone of the fuzzy model, which is composed of a set of fuzzy if-then rules describing the input-output relationship being recognized at the level of the information granules. All the values of the parameters of the rule-based model are stored in the database. These parameters include the definition of the universes discourse of the input and output variables and the details of the membership function. The inference engine is the computational method which calculates the degree to which each rule fires for a given fuzzified input pattern. Finally, the output inference translates the results from the inference engine into the suitable format required by the application domain. This process is called defuzzification.

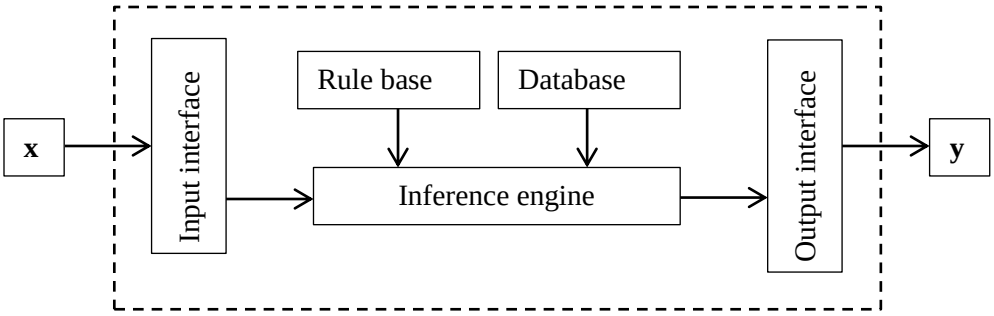
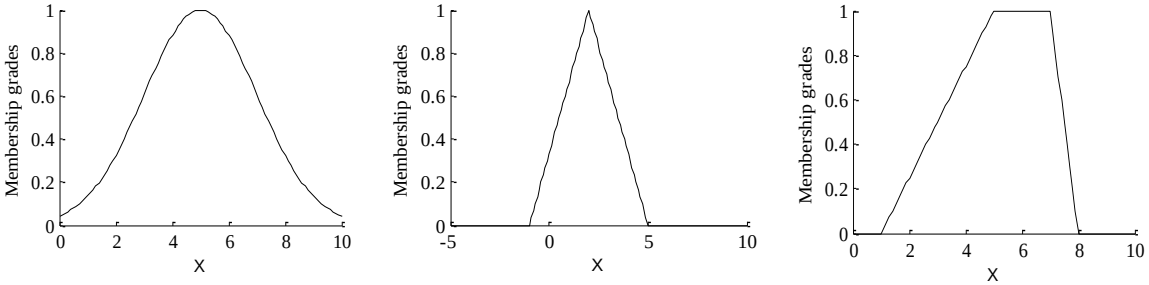


Figure 2-2: The general configuration of a fuzzy model



(a) (b) (c)

Figure 2-3: Examples of membership functions; (a) Gaussian, (b) Triangular, and (d) Trapezoidal

In a fuzzy rules-based system, the relationships between variables are presented by the following form:

IF an antecedent (proposition) **THEN** a consequent (proposition) (2-16)

The three different types of rule-based fuzzy models are categorized according to their consequent proposition representation:

1. Linguistic Fuzzy Models where both the antecedent and consequent parts are fuzzy sets (Pedrycz and Gomide 2007). The following expression is the general form of linguistic fuzzy model:

   R_i: if X is A_i and Y is B_i then Z is C_i (2-17)

   where R_i denotes the i-th rule, i=1,2,…,N, and N is the total number of rules. X, Y, and Z are linguistic variables with base variables x, y and z. A_i, B_i, and C_i are fuzzy sets on **X**, **Y**, and **Z** described by a certain membership function.
2. Fuzzy relational models are the generalization of the fuzzy linguistic model. The process is based on the fuzzy relation and relational equation (Yi and Chung 1993, Pedrycz 1984).

3. Functional fuzzy models or *Takagi-Sugeno* fuzzy models have a rule base composed by fuzzy rules whose consequent propositions are functions of the antecedent proposition, rather than the fuzzy proposition (Sugeno and Kang 1988). The rule have the form:

$$R_i: \quad \text{if } X \text{ is } A_i \text{ and } Y \text{ is } B_i \text{ then } z = f_i(x, y) \tag{2-18}$$

where X and Y are linguistic variables with values A_i and B_i are fuzzy sets on **X** and **Y** with base variables x and y. Function $f_i(x, y)$ is any function of the antecedent variables that appropriately describes the model output in the region specified by the fuzzy Cartesian product of the antecedent fuzzy sets.

2.3.1 Fuzzy Linguistic Models

The generic version of processing realized in rule based systems is express in equation (2-17). The aggregation of the rules is realized as a union of the Cartesian products of the fuzzy sets standing in the antecedents and consequents parts of the individual rule. There are several options to aggregate the rules, but often rule aggregation is done by using minimum or product t-norms. Here, if-then rules defined as Cartesian products using the minimum or product t-norms, and the maximum s-norm to perform aggregation rules. In what follow we illustrate one of the most important linguistic model that is called min-max models (Pedrycz and Gomide 2007).

The main steps of min-max are as follows:

1. Antecedent matching: For each rule, compute the degree of matching by using possibility measure.

   $m_i = \max [\min(A(x), A_i(x))]$

   $n_i = \max [\min(B(x), B_i(x))]$
2. Antecedent aggregation: For each rule, compute the rule activation degree by conjunctively or disjunctively operating on the corresponding degrees of matching:

   $\lambda_i = \min(m_i, n_i)$
3. Rule result derivation: For each rule, compute the corresponding inferred value based on its antecedent aggregation and the rules semantics chosen.

   $C_i' = \min(\lambda_i, C_i)$
4. Rule aggregation: Compute the inferred value from the complete set of rules by aggregating the result of the inferred values derived from individual rules:

   $C(y) = \max (C_i, C_i')$

In case a numeric outcome of inference is required a certain decoding is completed. In spite of the evident simplicity of the overall construct outlined above, it exhibits a number of interesting properties and supports efficient nonlinear input-output mapping. In terms of the input-output

mapping realized here, one can treat the rule-based system as a certain associative memory realizing a retrieval of items given a finite collection of associations (Kosko 1992).

2.3.2 Takagi-Sugeno Fuzzy Model

The *Takagi–Sugeno* (TS) fuzzy model provides a systematic approach for generating fuzzy rules from a given input-output data set. This model is also called the Fuzzy Functional Model, where the rule base is composed by using fuzzy rules, whose consequent parts are a function of the antecedent variables (Pedrycz and Gomide 2007, Sugeno and Kang 1988). The consequent rules are represented by either the crisp number or linear functions of the input given. The antecedents' part is represented by a number of fuzzy regions based on the partition of the input space. The format of the TS model provides an effective way to represent the non-linear system by combining a rule-based description with local functional description, for example, in the form of linearization (Takagi and Sugeno 1985). The main process of TSK model can be divided into a two-step procedure of system identification: structure identification and parameter estimation. Here, the structure of the model is identified by using the given input-output data (Sugeno and Kang 1988).

Consider a function y=f(**x**) being mapped by the TS fuzzy model, in which y is the output variable (dependent variable) and **x** is the input variable (independent variable). The data set is in the form of finite input-output pairs, k=1, 2, …, M, where M is the total number of the input-output data available for parameter estimation. By considering N number of rules for generating the TSK fuzzy model, the representation of the TSK fuzzy model is

$$R_i : \text{if } x_1 \text{ is } A_{i,1} \text{ and } \dots x_n \text{ is } A_{i,k} \text{ Then } y_i = \mathbf{a}_i^T \mathbf{x} + a_{i0} \tag{2-19}$$

where R_i is the i-th rule, $A_{i,k}$ is a fuzzy subset, y_i is the predicted output from the i-th rule, the coefficient $\mathbf{a}_i^T$ of the linear equation is called the consequent parameter, and a_{i0} is the scalar offset. Here, each rule represents a locally linearized model with the fuzzy regions defined by the rules' premises.

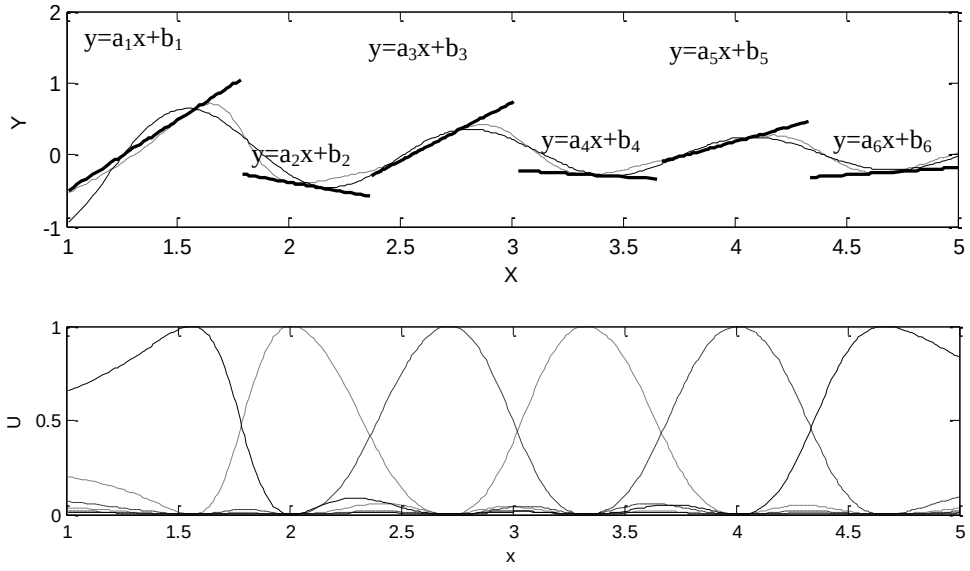
The inference formula of the TS model is represented by the following expression:

$$y = \frac{\sum_{i=1}^{C} \lambda_i(\mathbf{x}_k)\, y_i}{\sum_{i=1}^{C} \lambda_i(\mathbf{x}_k)} = \frac{\sum_{i=1}^{C} \lambda_i(\mathbf{x}_k)(\mathbf{a}_i^T \mathbf{x} + a_{i0})}{\sum_{i=1}^{C} \lambda_i(\mathbf{x}_k)} \tag{2-20}$$

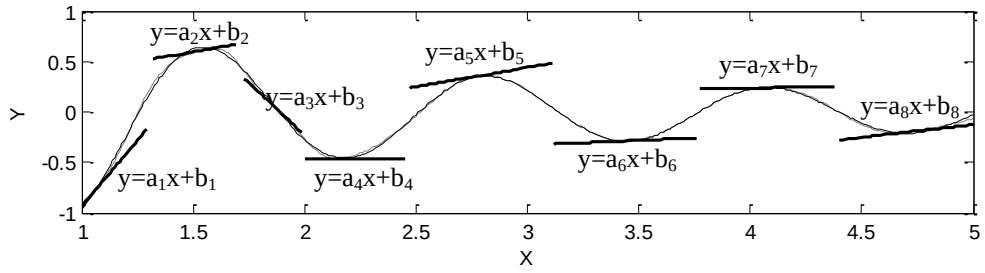
Where $\lambda_i(\mathbf{x}_k)$ is the degree of activation of rule R_i.

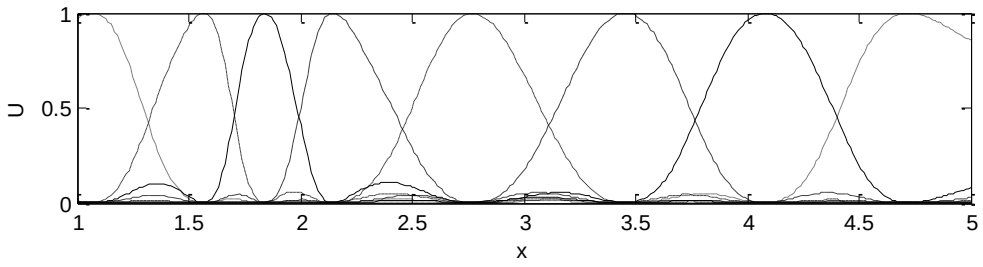
Figure 2-3 illustrates the example of a single–input single-output (SISO) fuzzy TS model. This model can describe the non-linearity where the universes of the input are partitioned by using linguistic labels, and the output is partitioned by using polynomials. Here, **X** is the input variable with six membership functions: A_1, A_2, A_3, A_4, A_5 and A_6, and y is the output variable of the form y= ax + b. In this case, we represent the fuzzy model with six rules as follows:

R_1	:	If x is A_1 then $y=a_1x+b_1$
R_2	:	If x is A_2 then $y=a_2x+b_2$
R_3	:	If x is A_3 then $y=a_3x+b_3$
R_4	:	If x is A_4 then $y=a_4x+b_4$
R_5	:	If x is A_5 then $y=a_5x+b_5$
R_6	:	If x is A_3 then $y=a_6x+b_6$



(a)





(b)

Figure 2-3: TS model as a smooth piece-wise linear approximation of non-linear function, (a) number of rules is 6 and (b) number of rules is 8

The most crucial task in constructing a TSK fuzzy model is to perform structure identification and parameter estimation. The structure identification concerned with determining the number of rules and the parameter estimation is refers to the calculation of the appropriate model parameter values for the fuzzy model (Rezaee and Zarandi 2010, Tsekourus 2005). Figure 2-4 displays the overall processes for a fuzzy model by using input-output data that consists of two parts: structure identification and parameter estimation. Structure identification can be approached as the problem of partitioning the input space **X** into the minimum number of fuzzy subspaces needed to form the fuzzy model. Various approaches have been proposed to construct the structure of the fuzzy model such as the Fuzzy C-Means clustering algorithm (Tsekourus 2005, Kim, et al. 1997, Gomez-Skarmeta and Delgado 1999, Chui 1994, Gillaume 2001). The Fuzzy C-Means clustering produces a fuzzy partition of the input space by using cluster projections (Emami, Turken and Goldenberg 1998). The results of transforming numeric data into fuzzy sets are used directly in constructing a rule-based system. Parameter estimation also can be called the consequent parameter estimation. It consists of determining the optimum parameter $\mathbf{a}_i$ to minimize a performance index. Here, the performance index is the root mean square error of the output error.

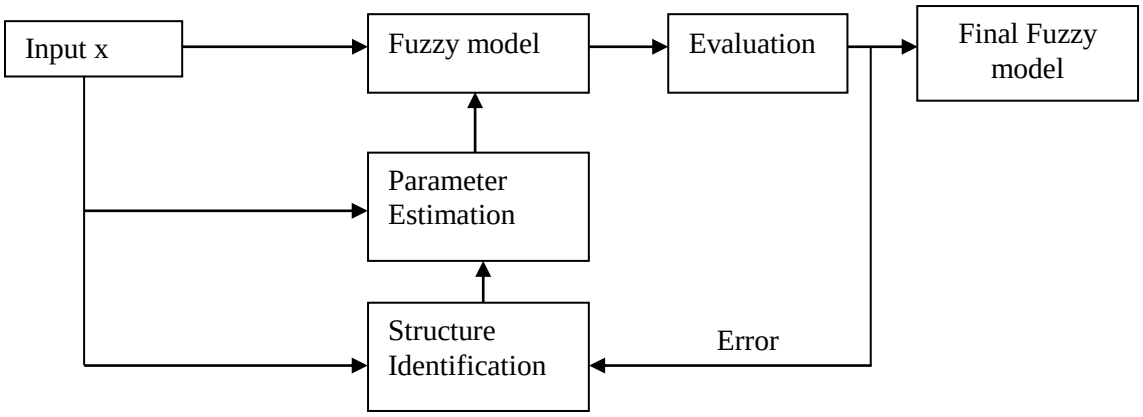


Figure 2-4: TS process

2.3.2.1 Fuzzy model structure identification and parameter estimation

The process of model structure identification is based on the fuzzy clustering method (Chui 1994, Yoshinari, Pedrycz and Hirota 1993, Chen, Xi and Zhang 1998). The concept of fuzzy clustering allows the partitioning of the collected input data from a given dataset. The group of clusters is formed by determining the data points that have system behavior, which are interrelated to each other. Here, the characteristics of each cluster can be used as the properties to identify their members from the members of the other clusters. In Fuzzy C-Means (FCM), each data point belongs to a cluster based on the degree specified by a membership grade. The cluster information can be used as a rule that describes the fuzzy model. The output of the fuzzy model is inferred as

$$\hat{y}_k = \sum_{i=1}^{C} w_{ik} f_i(\mathbf{x}_k, \mathbf{a}_i) \tag{2-21}$$

with the w_{ik} expressed in the form of

$$w_{ik} = \frac{\lambda_i(\mathbf{x}_k)}{\sum_{i=1}^{C} \lambda_i(\mathbf{x}_k)} \tag{2-22}$$

where, $\lambda_i(\mathbf{x})$ is the i-th rule firing strength. Obviously, the firing values are the membership grades generated from the fuzzy clustering approach. The $\lambda_i(\mathbf{x})$ is defined as

$$\lambda_i(x) = A_{i1}(x_1) \wedge \ldots \wedge A_{in}(x_n) \tag{2-23}$$

where $A_i(x)$ is the membership grades.

After the structure identification of the fuzzy model, the next step is to fine-tune its consequent parameter values in order to improve the model performance. The parameter estimation step can be viewed as a linear/ nonlinear optimization problem requiring the minimization of some predefined loss function such as the mean squared error.

The consequent parameter for TS fuzzy models corresponds to the parameters of the local models. These local models are polynomial defined on input **x**; therefore, the output of the TS fuzzy model is always linear in the consequent parameters. Thus, the identification of these parameters can be simply a linear optimization problem (Pedrycz and Gomide 2007). The least square method is used for solving this problem. The unknown consequent parameter, **a**, can be represented as

$$\mathbf{a} = \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \\ \vdots \\ \mathbf{a_N} \end{bmatrix} \tag{2-24}$$

Here, we want to estimate the optimal consequent parameter, **a**, by using a set of M training data points. In this case, the structures of the fuzzy model are fixed; thus, each data point results in a linear equation, as shown in the following equation:

$$\hat{y}_k = \sum_{i=1}^{C} \mathbf{z}_{ik}^T \mathbf{a}_i \tag{2-25}$$

where $\mathbf{z}_{ik} = w_{ik}\mathbf{x}_{ik}$ and $\mathbf{a}_i = [a_{i0}, a_{i1}, \ldots, a_{in}]$ is the consequent parameter. By using M training points, we generate the system of linear equations as the following:

$$Z\mathbf{a} = \hat{y}_k \tag{2-26}$$

where **a** is the unknown parameter vector given in equation (2-26), $\hat{y}$ is the predicted output from the fuzzy model, and Z is matrix given by

$$Z = \begin{bmatrix} \mathbf{z}_{11}^T & \mathbf{z}_{21}^T & \cdots & \mathbf{z}_{N1}^T \\ \mathbf{z}_{12}^T & \mathbf{z}_{22}^T & \cdots & \mathbf{z}_{N2}^T \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{z}_{1M}^T & \mathbf{z}_{2M}^T & \cdots & \mathbf{z}_{NM}^T \end{bmatrix} \tag{2-27}$$

Obviously, we expect that the output of the model should equal the experimental data. Therefore, we require that y = Z**a** where **a** should result as a solution to the system of the M linear equation. The solution of the linear system is not unique because of the larger M compared to the number of the parameter **a**. Therefore, we can solve the value of **a** by using a certain optimization process. Here, we minimize the distance between y and Z**a**. To deal with the estimation of the parameter we look at the determination of the minimize J:

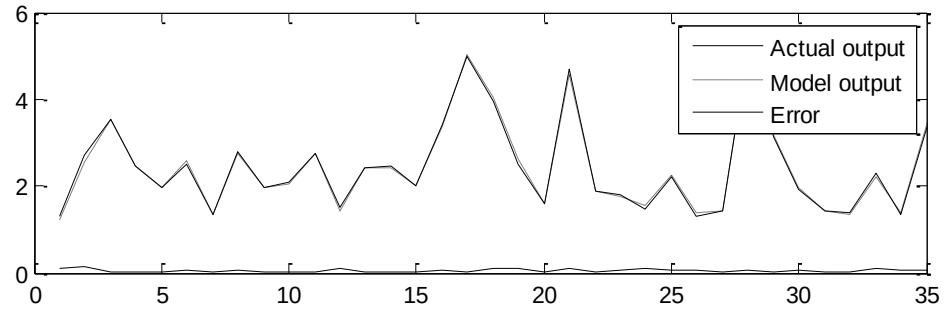
$$\mathrm{Min}_a J = \|y - Z\mathbf{a}\|^2, \tag{2-28}$$

Employing the least square method the vector of **a,** that minimizes J is

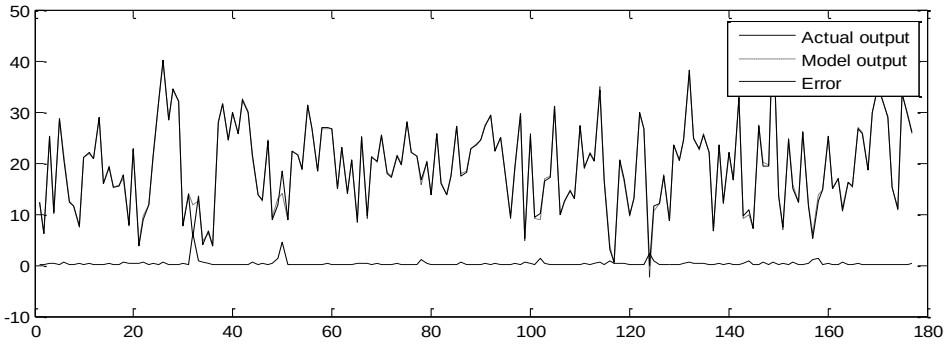
$$\mathbf{a}_{opt} = Z^{\#}y \tag{2-29}$$

where $Z^{\#} = (Z^T Z)^{-1} Z^T$.

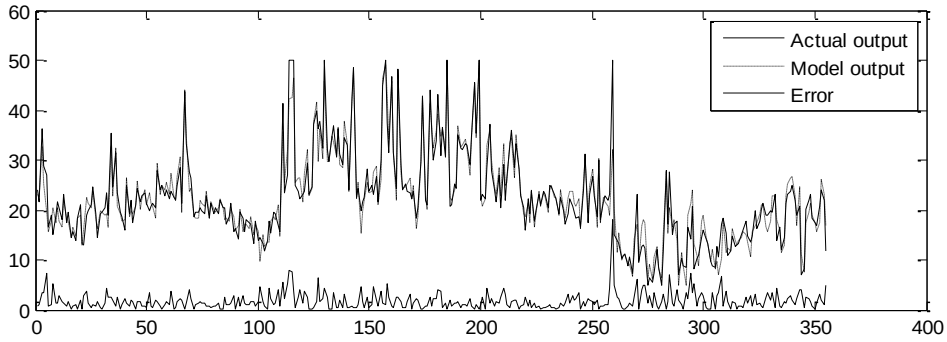
In Figure 2-5 we show several TS fuzzy models' results by using several data sets with different dimensionality.



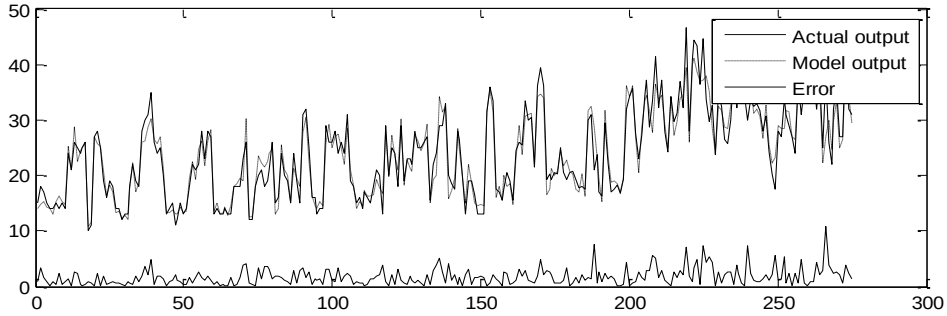
(a)



(b)



(c)



(d)

Figure 2-5: Comparison of output data with model's output; (a) Static function (dimensionality=2), (b) Body fat data (dimensionality=14), (b) Housing data (dimensionality=13), and (d) Auto-MPG data (dimensionality=7)

2.4 Optimization method

Population-based algorithms provide efficient solutions since any constructive method can be used to generate the initial population, and any local search technique can be used to improve each solution in the population (Hertz and Kobler 2000). In addition, population-based methods have the advantage of being able to combine good solutions in order to get possibly better ones. The basic idea behind this approach is that good solutions often share parts with optimal solutions. Here, we use two kinds of population-based methods: Genetic Algorithms and Particle Swarm Optimization. These methods are useful in developing an algorithm that can adapt to different types of population-based methods. Moreover, each of the methods has its own pros and cons that we can fully utilize in order to get a good optimization with minimum computation complexity.

2.4.1 Genetic algorithm

Genetic algorithms (GA) are a stochastic optimization technique that explores the search space by using the mechanisms encountered in natural evolution: mutation, crossover (recombination), and selection (Karray and DeSilva 2004). In nature, the fitness of individuals depends on their phenotype, which is directly influenced by their chromosome (genotype). Individuals with greater fitness have a greater chance of survival and also a wider range of mating partners to choose from within the population. The 'new' individuals are generated by genetic operators, namely crossover and mutation (Eiben and Smith 2003). The implementation of GA requires the determination of six fundamental issues: chromosome representation, selection function, the function of genetic operators, initialization, termination and evaluation function.

The basic structure of GA includes three main operations:

1. The process of evaluating of individual fitness
2. The construction of an intermediate population through the selection mechanism
3. The recombination process via the two operators (crossover and mutation)

The chromosome representation of the population is the main concern in GA. The popular representation uses the fixed-length and binary coded strings. However, for more complex problems, the use of a real coded string is favorable because the representation is more natural to the specific application domains (Herrera 1998).

The mechanisms for selection can be divided into selection probability calculation and sampling algorithm. The most well-known selection methods in selection probability are proportional selection, tournament selection, and ranking selection. After the selection process has been carried out, the construction of the intermediate population is complete, and we move to

the final stage, or the recombination process, which involves two operators called crossover and mutation. The crossover operator plays an important role in GA, where this operator is involved the sharing of information between chromosomes; it combines the features of two parent chromosomes to form two children, with the possibility that the good parent may generate better chromosome. Next, the role of the mutation operators is to prevent the premature convergence of GA to suboptimal solutions. The mutation operator restores lost or unexplored genetic material into the population. These two operators are not usually applied to all pairs/single chromosomes in the immediate population. A random choice is made according to the probability defined by the mutation rate and crossover rate. Finally, another selection technique called the elitist strategy is usually implemented after the process of crossover and mutation. By using the elitist strategy, we can confirm that the best performing chromosome always survives intact from one generation to the next. Figure 2-6 shows the pseudo-code for the GA.

```
begin (1)
        t=0
        Initialize population: P(t)
        Evaluate: P(t)
        While stop criteria not met do
        begin (2)
                t=t+1
                Select: P(t) form P(t-1)
                Recombine: P(t)
        Evaluate: P(t)
        end
end
```

Figure 2-6: Pseudo code for GA

2.4.2 Particle swarm optimization

Particle swarm optimization (PSO), a population-based stochastic optimization technique, was developed by Kennedy and Eberhart (1995). PSO simulates the social behavior of organisms, such as bird flocking and fish schooling, to describe an automatically evolving system. In PSO, each single candidate solution is a particle in the search space. Each particle uses its individual memory and knowledge gained by the swarm as a whole to find the best solution. All of the particles have fitness values, which are evaluated according to their fitness function (which is later optimized), and have velocities which direct the movement of the particles. During movement, each particle adjusts its position according to the experience of a neighboring particle, and makes use of the best position achieved by itself and its neighbor. The particles move through the problem space by following a current of optimum particles (Paterlini and Krink 2006).

The initial swarm is generally created so that the population is distributed randomly over the search space. Each particle memorizes two positions in order to find the best position in the search space. One is its own best position, called the personal best, and the other is the global best, that is the best among all particles, denoted by P_{LB} and P_{GB}, respectively (Hu, Shi and Eberhart 2004). The neighbor particle is defined by the topology of the particles, which represents the networks structure of the population. Memories are utilized in adjusting the velocity to find better solutions.

In one of the standard versions of the PSO algorithm, the velocity and position are updated at each time step, according to the following two equations:

$$v_{id}(iter+1) = w.v_{id}(iter) + c1.r1\,(P_{LBi} - x_{id}(iter)) + c2.r2\,(P_{GB} - x_{id}(iter)) \quad (2\text{-}30)$$

$$x_{id}(iter+1) = x_{id}(iter) + v_{id}(iter+1) \quad (2\text{-}31)$$

where v_{id} indicates d-th dimension of the velocity of the i-th particle, and x_{id} indicates the d-th dimension of the i-th position. The two factors r1 and r2 are two random numbers uniformly distributed in the range [0, 1], whereas c_1 and c_2 are acceleration factors; usually $c_1<c_2\leq 2$. The inertia weight w was linearly from 1 to 0 over the course of optimization. The use of w prevents an explosion of the velocity, and provides balances between exploration and exploitation. The values P_{LBi} is defined as follows (assuming a maximization problem):

$$P_{LBi} = \begin{cases} P_{LBi} & \text{if} \quad f(P_{LBi}) \geq f(x_i) \\ x_i & \text{if} \quad f(P_{LBi}) < f(x_i) \end{cases} \quad (2\text{-}32)$$

Finally the P_{GB} is updated as follows:

$$P_{GB} = \arg\max_{P_{LBi}} f(P_{LBi}) \quad (2\text{-}33)$$

where f (.) is the objective function that evaluates the fitness value for a given position.

Figure 2-7 shows the pseudo-code for the standard PSO algorithm.

```
Initialize n-dimensional PSO: P
While stop criteria not met do
        for each particle i ∈ [1, ... ,s] do
                if f (P (x_i)) > f ( P_LBi)
                then  P_LBi = P(x_i)

                if f (P_LBi) > f ( P_GB)
                then  P_GB = P_LBi
        end for
                for each P do
                Perform PSO updates on P using Eqn. (2-30) & Eqn. (2-31)
                end for
        end for
end while
```

Figure 2-7: Pseudo code for Cooperative PSO

Several parameters need to be initialized in the first step of the algorithm:

1. The coordinate x_{id} is initialize to the value drawn from the uniform random distribution on the interval $[-x_{max}, x_{max}]$, for all i=1,2,….s and d=1,2,…n. This process distributes the initial positions of the particles throughout the search space.
2. The value of v_{id} is initialized to the value drawn from the uniform random distribution on the interval $[-v_{max},v_{max}]$ for all i=1,2,….s and d=1,2,…n. Usually, the velocities of the particles are initialized to 0**,** since the starting positions are already randomized.
3. Set $P_{LBi} = x_i$, for all i=1, 2, …, s.

Figure 2-7 portrays the use of the stopping criterion to stop the algorithm from searching for the best particle representing the optimization problem. Here, the stopping criterion depends on the type of problem being solved; for example, this criterion could use either a fixed number for the function evaluation (a fixed number of iterations) or a specified error bound.

2.4.2.1 Binary Particle Swarm Optimization

The modified version of PSO deals with the binary representation of the particles (Kennedy and Eberhart 1997). The position of each particle is given in a binary string form that represents the candidate solution for the optimization problem. The binary version permits x_i and P_{LB} to be either 0 or 1 (Khanesar and Teshnehlab 2007). Although there is no restriction on the value of velocity v_i for each particle the velocity is at the threshold of the range [0, 1] and is treated as a probability function. This process can be done by using the following sigmoid function,

$$S(v_{id}) = \frac{1}{1+e^{-v_{id}(iter+1)}} \tag{2-34}$$

The update equation for each particle is as follows:

$$x_{id}(iter+1) = \begin{cases} 0 & \text{if} \quad r(iter) \geq S(V_{id}(iter+1)) \\ 1 & \text{if} \quad r(iter) < S(V_{id}(iter+1)) \end{cases} \tag{2-35}$$

where r is random number uniformly distributed in the range [0, 1]. Here, if random number is greater than $S(V_{id})$, then its position value is represented as 0. On the other hand, if $S(V_{id})$ is smaller than a random number, the position value is represented as 1.

2.4.2.2 Cooperative Particle Swarm Optimization

The Cooperative Particle Swarm Optimization (CPSO) is another version of modified PSO and is suitable for dealing with a high dimensional search space. The divide and conquer concept that is usually used for solving complex problems has been implemented in constructing the CPSO. Here, the original problem is divided into several sub-problems, and each one is solved by using a different sub-swarm. The sub-swarms share the solutions they find and cooperate with others sub-swarms in order to reach a global solution for the problem at hand.

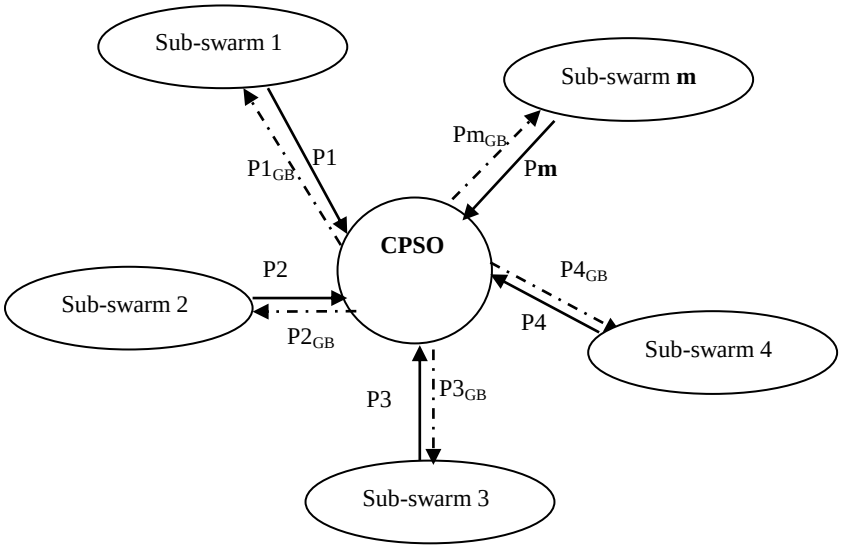


Figure 2-8: The schematic diagram of information sharing in CPSO

The CPSO's mechanism of information sharing is shown in Figure 2-8. The cooperative search between one sub-swarm and another is achieved by sharing the information of the global best position (P_{GB}) across all sub-swarms. Here, the algorithm has the advantage of taking two steps forward because the candidate solution comes from the best position for all sub-swarms except for the current sub-swarm being evaluated. Therefore, the algorithm will not spend too much time optimizing the candidate solutions that have little effect on the overall solution. The rate at which each swarm converges to the solution is significantly higher than the rate of convergence of the standard PSO. Figure 2-9 shows the general scheme of the cooperative PSO.

Figure 2-10 presents the Cooperative PSO pseudo code implementing the optimization process (van den Bergh and Engelbrecht 2004). Firstly, the particles are divided into m subspaces, called sub-swarms. $P_j(x_i)$ refers to the position of particle i of sub-swarms j. The global best for each sub-swarm is defined as $P_j(_{GB})$, and the local best is defined as $P_j(_{LBi})$. The cooperation between the sub-swarms is employed in the function **C**(j,k), which returns the m-dimensional vector formed by concatenating all the global best vectors across all sub-swarms, except for the current position j. Here, the j-th component is called k and represent the position of any particle from sub-swarm P_j.

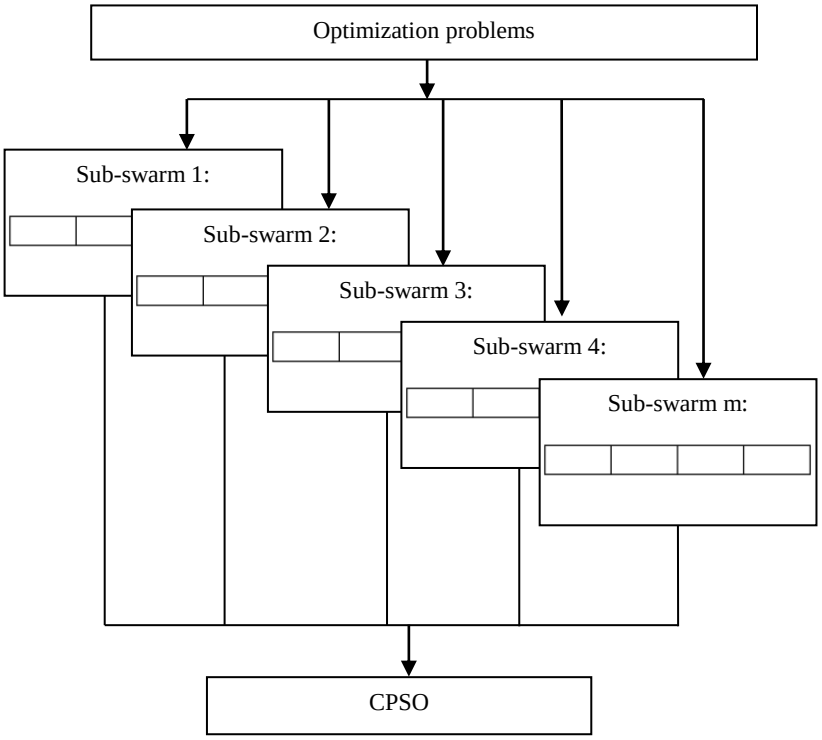


Figure 2-9: The particle scheme of Cooperative PSO

```
Initialize m one-dimensional PSO: P_j, j∈ [1, ... ,m]
Create
C(j,k)=[ P_1(BG),P_2(BG),..., P_j-1(BG), k , P_j+1(BG), ..., P_m(BG) ]
While stop criteria not met do
        for each sub-swarm j ∈ [1, ..., m] do
                for each particle i ∈ [1, ... ,s] do
                        if f (C(j, P_j (x_i) )) > f (C(j, P_j (LBi)))
                                then  P_j (LBi)= P_j(x_i)
                        if f (C(j, P_j (LBi) )) > f (C(j, P_j (GB)))
                                then  P_j (GB)= P_j(LBi)
                end for
                for each P_j do
                        v_i,j (t+1) = w.v_i,j (t)+c_1.r_1,i (t)[ P_LBi,j (t) - x_i,j (t)]
                                +c_2.r_2,i (t) [ P_GBi,j (t)  - x_i,j (t)]
                        x_i,j (t+1)= x_i,j (t)+v_i,j (t+1)
                end for
        end for
end while
```

Figure 2-10: Pseudo code for Cooperative PSO

2.5 Conclusions

This chapter has provided an overview to Granular Computing and fuzzy clustering. This chapter has also elaborated in detail the fundamental aspects of fuzzy modeling and the population-based methods being used for solving the optimization problems.

3.Feature Reduction through Structure Retention

In this chapter, we explain in details about the method of using feature reduction, in which the reduction process is guided by a criterion of structure retention. The features in the reduced space are selected in such a way that the original structure present in the original highly dimensional space is retained to the highest possible extent. The chapter is arranged into 5 sections. We start with the brief introduction about the feature selection. Then we present concept of data granulation and the idea of structure retention in the reduced feature space (Section 3.2). The granulation-degranulation principle is used along with its performance index and its direct usage to the optimization of a subset of features in the reduced feature space (Section 3.3). Experimental studies are reported in Section 3.4 while some conclusions are covered in Section 3.5.

3.1 Feature selection

Recently, high dimensional data set are becoming the norm as the process of data collection becomes automated. As a result, the knowledge discovery from these datasets faces important challenges. This motivates the idea of feature reduction in fuzzy clustering.

The problem of feature reduction (Lui and Horoshi 1998, Jain, Murty and Flynn 1999, Pavlenko 2003, Dy and Brodley 2004) has occupied an important position in pattern recognition as being of paramount relevance to the design of a variety of classifiers impacting their effectiveness and accuracy. There have been a significant number of studies in this area, which, in general, can be classified as wrappers or filters. The design of feature selection techniques falling under the category of filters stresses the generality of the selection process offered (which is independent from a particular form of the classifier). Quite commonly, the filters exploit some information-theoretic criteria relying on the probabilistic characteristics of data. The wrapper approaches are focused on a certain, predefined type of the classifier. A set of the resulting set of features could be viewed optimal in the context of the given classifier (Lee, et al. 2001). The same set of features could result in a quite poor performance when some other classifier is considered. There have been some hybrid approaches, see e.g., (Uncu and Turksen 2007). The ideas of fuzzy clustering have been also a subject of studies in this area (Marcelloni 2003, Nuovo and Catania 2008, Pedrycz and Bargiela 2010).

The proposed approach is guided by the idea of structure preservation: given a structure of data in the original space, we reduce a feature space in such a way so that the original structure is retained to a significant extent. The structure in data is determined through fuzzy clustering and the abilities of the structure obtained in this way are quantified through the granulation-degranulation criterion. However, as structure of data is fundamental to a way in which classifiers are built, the study here exhibits some linkages with the category of feature wrappers. Likewise in other studies on feature selection problems, which are of combinatorial character and exploit evolutionary and population-based techniques of optimization, see (Nouvo and M. Palesi 2007, Hall, Ozyurt and J.C.Bezdek 1999, Wang, et al. 2007), in this study we use methods of Genetic Algorithms and Particle Swarm Optimization.

We use the standard notation used in pattern recognition. A set of n-dimensional patterns $\mathbf{x}_1$, $\mathbf{x}_2$, …,$\mathbf{x}_n$ is located in $\mathbf{R}^n$. The distance function between two elements in $\mathbf{R}^n$ is the Euclidean one with eventual weighting by the standard deviation, which express in equation (2-12).

3.2 Data granulation and structure retention

Data are granulated giving rise to a certain, quite limited number of information granules. In a nutshell, information granules form a collection of entities brought together, which exhibit a certain, well-defined semantics (meaning) being reflective of the nature of the problem. Fuzzy clustering, in general offers an algorithmic framework to design information granules. In this case, each information granule is a cluster. The collection of information granules (along with their form and distribution) is reflective of the underlying structure in the data. It becomes apparent that the clusters quantify the structure of data. If we reduce the feature space by retaining only a subset of the features, we would like to complete this reduction in such a way so that the structure (being conveyed by the clusters) becomes retained in the resulting reduced space to the highest possible extent. This intuitively appealing formulation has to be made fully operational that is expressed in terms of some tangible optimization criterion. It is expressed in the form of the reconstruction criterion, which comes as a result of the granulation–degranulation principle

The representation of any numeric data in the form of information granules by using the clustering process can be referred to as granulation (encoding). Then the conversion of the information granules into the numeric data is referred to as degranulation (decoding). Both these processes are also known as a reconstruction problem. Figure 3-1 illustrates the granulation and degranulation of the numeric data by using information granules.

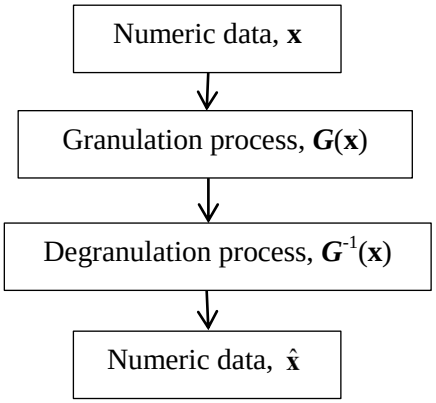


Figure 3-1: The granulation-degranulation of numeric data through information granulation

The information granules (clusters) forming a collection of G are described by their prototypes v_1, v_2, …, v_c. The granulation of **x** returns its representation in terms of the collection of the available information granules expressed in terms of their prototypes. For instance, **x** is expressed in the form of the membership grades u_i of **x** to the individual granules $\boldsymbol{G}$ (A_i). The following equation describes the information granules as a vector in the c-dimensional hypercube, namely $[0, 1]^c$:

$$\boldsymbol{G}: \ \mathbf{R}^n \rightarrow [0,1]^c \tag{3-1}$$

The degranulation step involves the reconstruction of **x** based on the family of information granules (clusters). This step can be treated as a certain mapping:

$$\boldsymbol{G}^{-1}: [0,1]^c \rightarrow \mathbf{R}^n \tag{3-2}$$

The capabilities of the information granules to reflect the structure of the original data can be conveniently expressed by comparing how much the result of degranulation, say, $\hat{\mathbf{x}}$**,** differs from the original pattern of **x**; that is, $\hat{\mathbf{x}} \neq \mathbf{x}$. More formally, $\hat{\mathbf{x}} = \boldsymbol{G}^{-1}$ ($\boldsymbol{G}$(**x**)) where $\boldsymbol{G}$ and $\boldsymbol{G}^{-1}$ denote the corresponding phases of information granulation and de-granulation. The mechanisms of granulation and de-granulation as well as a detailed elaboration of the underlying principle were discussed in detail in (Pedrycz and Oliveira 2008, Pedrycz 2005). The granulation $\boldsymbol{G}$ of **x** carried out in terms of the information granules (prototypes) results in the c-dimensional vector **u**(**x**) in the $[0,1]^c$ with the use of equation (2-11) in the FCM method. The degranulation $\boldsymbol{G}^{-1}$ leads to the reconstruction expressed in terms of **u** and the prototypes of the clusters,

$$\hat{\mathbf{x}} = \frac{\sum_{i=1}^{c} u_i^m \mathbf{v}_i}{\sum_{i=1}^{c} u_i^m} \tag{3-3}$$

Ideally, we wish to see equality, yet in practice, this is not accomplished. The quality of information granulation is quantified in terms of the following reconstruction criterion:

$$Q = \sum_{k=1}^{M} (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T (\mathbf{x}_k - \hat{\mathbf{x}}_k) \tag{3-4}$$

where the above sum is taken over all data of interest, k=1, 2,…,M. In particular, we can think of this data set as the one originally used in the clustering process. The lower the values of Q, the better the representation of data in terms of the information granules, which capture the structure of the data thus allowing for, lower values of the reconstruction criterion. The essence of the granulation–degranulation is visualized in Figure 3-2. Note the transformations and $\boldsymbol{G}^{-1}$ operate between the spaces of data and information granules.

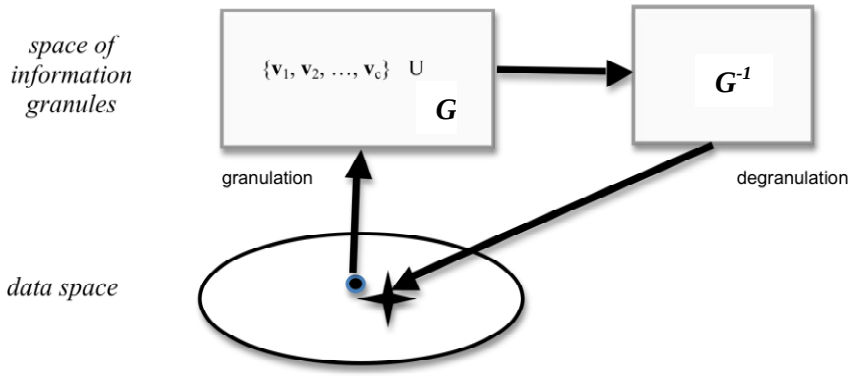


Figure 3-2: The granulation-degranulation mechanism as a realization of mapping between data space and the space of information granules

3.3 Feature selection: combinatorial optimization with the use of Genetic Algorithms

Feature reduction is guided by the reconstruction criterion. We envision a feature space being optimally reduced if the granulation–degranulation mechanism realized for the data in this reduced space leads to the minimum of the reconstruction error. Schematically, the overall process is outlined in Figure 3-2. Originally, information granulation (obtained through clustering) is realized in the original feature space producing a collection of prototypes $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_c$. The data in the reduced feature space is granulated and expressed by information granules in the reduced feature space. The corresponding membership grades of **x** are computed in the same way as expressed by (2-11) but now the corresponding distances involved in these calculations are determined in the reduced space. In a convenient way, we express the reduced feature space by introducing an n-dimensional Boolean vector $\mathbf{b} = [b_i]$, i=1, 2, …, n with the following entries

$$b_j = \begin{cases} 1 & \text{if the j– th feature is included in the reduced feature space} \\ 0 & \text{otherwise} \end{cases} \tag{3-5}$$

For instance, the weighted Euclidean distance between $\mathbf{x}$ and $\mathbf{v}_i$, $\|\mathbf{x}-\mathbf{v}_i\|_\mathbf{b}$ computed in the reduced feature space (which is characterized uniquely by the associated vector $\mathbf{b}$) reads as

$$\| \mathbf{x} - \mathbf{v}_i \|_b^2 = \sum_{j=1}^{n} \frac{(\mathbf{x}_j - v_{ij})^2 \, b_j}{\sigma_j^2} \tag{3-6}$$

with σ_j being the standard deviation of the j-th feature. This gives rise to the following membership grades

$$\tilde{u_i}(\mathbf{x}) = \frac{1}{\sum_{j=1}^{c} \left(\frac{\| \mathbf{x} - \mathbf{v}_i \|_b^2}{\| \mathbf{x} - \mathbf{v}_j \|_b^2} \right)^{2/(m-1)}} \tag{3-7}$$

which, as seen above, are determined based upon the distances computed in the reduced feature space. The notation used to denote membership grades ($\tilde{u}$) stresses a fact that these values are computed based on the reduced feature space.

The degranulation is realized with the aid of the information granules (membership values), which are formed in the reduced feature space, see also Figure 3-3.

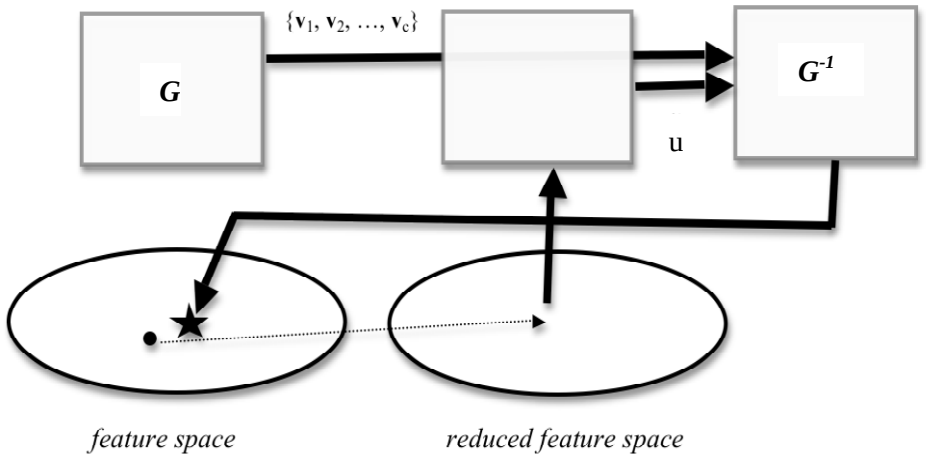


Figure 3-3: The degranulation process realized in the reduced feature space and evaluation of its quality

More specifically, the reconstructed pattern comes in the form

$$\hat{\mathbf{x}} = \frac{\sum_{i=1}^{c} \left(\tilde{u_i}\right)^m \mathbf{v}_i}{\sum_{i=1}^{c} \left(\tilde{u_i}\right)^m} \tag{3-8}$$

It becomes apparent that the problem, as being of combinatorial nature, calls for the use of techniques capable of realizing such structural optimization. A sound choice with this regard would be to engage some techniques of evolutionary optimization or population-based optimization such as e.g., Genetic Algorithm or Particle Swarm Optimization. The fitness function of interest is the reconstruction error. Let us denote by **F**' a subset of all features **F**, **F**' $\subset$ **F**. The optimization task is expressed as

$$\text{Min}_{F'} \, Q, \tag{3-9}$$

for the cardinality of the reduced set of features **F**', card (**F**'), being specified in advance. The minimized performance index V is the reconstruction error determined as shown in (3-4) but now the reconstructed pattern is determined using the partition matrix formed in the reduced feature space, see (3-7).

The genetic algorithm (GA) is well documented in the literature along with the application to feature selection (Jain and Zongker 1997, Enzhe and Sungzoon 2006, Yang and Honavar 1998, Raymer, et al. 2000). GA is an example of biologically inspired technique that is aimed at structure optimization using to construct an optimal collection of at features. Here, the potential solution is represented in a form of a chromosome that identifies the features contributing to the formation of the optimal feature space. The proposed populations for GA are using a real-number representation in the interval [0, 1]. The proposed chromosome is a direct reflection of the feature space for a particular data set. Let us assume that the dimensionality of selected feature space to be used is given in advance and equal to p where $p < n$. The chromosome of the populations corresponds to the subset of the original feature space that is chosen as select features. Here, with the predefined dimensionality of feature space (p), we then use only the first p entries of the chromosome and this produces a sequence of the selected features. The entries in the chromosome are ranked (ascending order) and the first p entries of the chromosome are used to choose among all features. The basic mechanisms of GA used in this study involve the elitism strategy that re-uses the overall best individuals of the whole population, the tournament selection that select the population for the next iteration, and the standard crossover and mutation operators. The number of chromosome involved in the crossover and mutation are defined according to the probability of crossover and probability of mutation, respectively.

In this research, we prefer to use a real coded GA that generates individual genotypes based on the number of selected features is given by the user. Therefore, the genotype of the population is a real number in the interval between 0 and 1. The dimension of the genotype is constant and equal to the number of features. The genes are ranked (in increasing order) and the entries represent the features group number (refer to Figure 3-4). Therefore, the purpose of applying the ranking approach to the genotype is to map the genotype to the phenotype representation. The phenotype represents an index of the features.

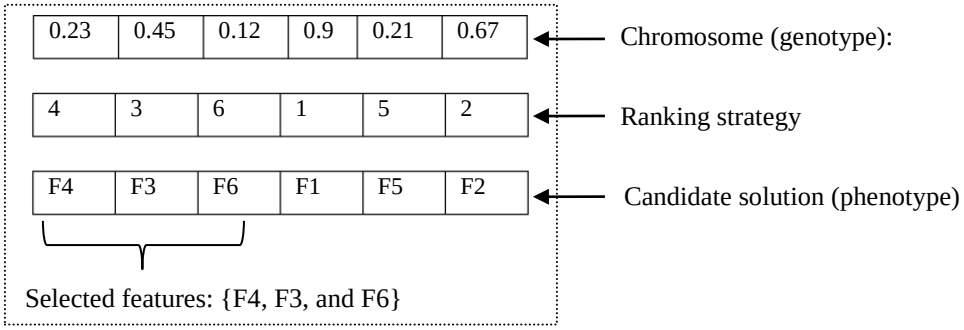


Figure 3-4: The process of forming a subset of feature for a given chromosome

In our GA implementation, a population of individuals containing the candidate solutions (encoded in floating point numbers) is created and the fitness of each individual is evaluated by the fitness function. In the initialization of the population, the GA uses randomly chosen clusters. After initialization, we evaluate the individuals according to the fitness function and determine the elite. To do this, we rank the population to identify and mark the best individuals, which are left unchanged by selection, mutation and crossover operators during the next iteration. The population is iteratively refined by the selection of individuals using tournament selection, the application of mutation and crossover operators, and the re-evaluation of the new population according to the fitness function and the updating of the elite. The details of the genetic algorithm can be found in the literature (see e.g. ref (Eiben and Smith 2003, Whitley 1994, Haupt and Haupt 2004)).

Particle Swarm Optimization (PSO) is also based on the biological inspired technique. Both GA and PSO are similar in the sense that these two methods are population-based search methods and they search for the optimal over a number of iterations. In PSO each particles compete to improve themselves by duplicating traits from their successful peers. Furthermore, each particle has a memory and hence it is capable of remembering the best position in the search space ever visited by it. The implementation of PSO requires the determination of three basic issues: particle representation, the cognitive and social acceleration factors, and the inertia weight factor. The representation of the particles is same as the population's representation in GA and the value for

cognitive and social acceleration factors are both equal to 2. Here we used the time decreasing inertia weight to control the overall velocity of the swarm.

3.4 Experimental studies

In this section, we elaborate on a set of experiments, in which we used several Machine Learning data sets (see http://www.ics.uci.edu/~mlearn/MLRepository.html). The objective of these experiments is to show the abilities of the proposed method and quantify a performance of the reduced subsets of features. A brief summary of the data sets used in the experiments is presented in Table 3-1. The data concern both classification (discrete outputs) and regression problems (continuous outputs). The choice of specific values of the genetic optimization was made on a basis of dimensionality of the feature space of the data. The size of the population was set to 50 (for data with dimensionality of the feature space of 10 or less) and 100 for the data with the feature space of higher dimensionality. These values were determined experimentally where it was found that their further increase did not led to any improvement of the results. The crossover rate and probability of mutation were equal to 0.75 and 0.15, respectively. Again, these particular values of the GA parameters are in line with those encountered in the literature.

Table 3-1: Data description (note that Boston housing and Auto MPG have continuous outputs)

Data set	*Abbreviation*	*Number of data*	*Number of features*	*Number of classes*
Ecoli	Ecoli	336	7	8
Glass Identification	Glass	214	9	6
Pima Indian Diabetes	Pima	768	8	2
Wine Recognition	Wine	178	13	3
Boston Housing	Housing	506	13	-
Auto MPG	Auto	392	7	-
Vowel	Vowel	990	10	11

The FCM was run for each data set using the complete feature space. The parameters of the clustering method were equal as: m=2 for fuzzification coefficient while the termination criterion of the clustering was set to $\varepsilon = 10^{-5}$ (meaning that the method was terminated once the distance between partition matrices in two successive iterations does not exceed ε (Eqn. (2-15)). The weighted Euclidean distance in Eqn. (2-12) was used in all experiments. The method was initialized by starting with random entries of the partition matrix. The reconstruction error is the fitness function used in the genetic optimization. The optimization process was repeated 10 times. A visualization of the performance of the evolutionary optimization is presented in Figure 3-5.

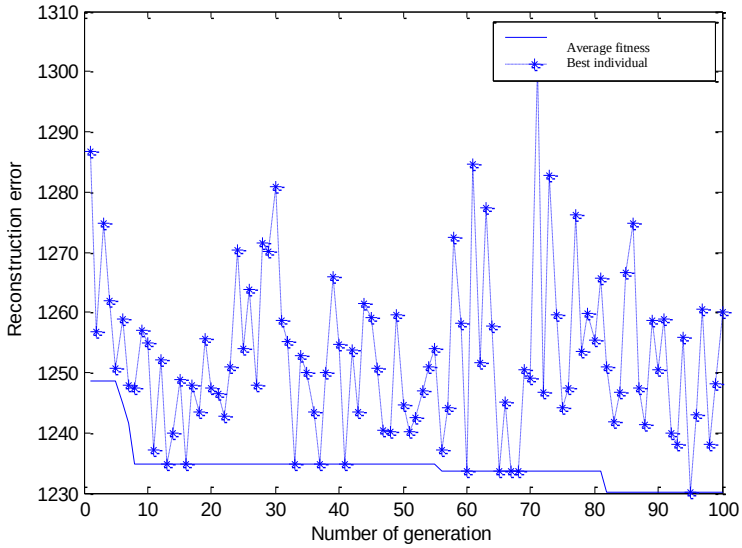


Figure 3-5: Values of the fitness function in successive generations – fitness of the best individual and average fitness.

The average values of the fitness function along with the associated standard deviation are illustrated in Figure 3-6. Here the reconstruction error is reported versus the dimensionality of the input space and this relationship is provided for selected number of clusters. As expected, the reconstruction error is a decreasing function of the number of features. The relationship between the reduced dimensionality of the input space and the resulting values of V show that while a few retained features result in significant values of the reconstruction error, the dependency is quite "flat" with the increase of the number of the features being retained. For instance, in most cases, when moving up beyond 4 or 5 features, the reduction of the reconstruction quality is limited. Considering the reconstruction error for the entire input space to be a reference value, V_{ref}, the ratio of V for dimensionality 4 and above to V_{ref} are close to 1, namely for Pima we obtain 0.994. Similarly for Auto MPG we have 0.979. Table 3-2 shows the complete results for all data sets used in the experiments.

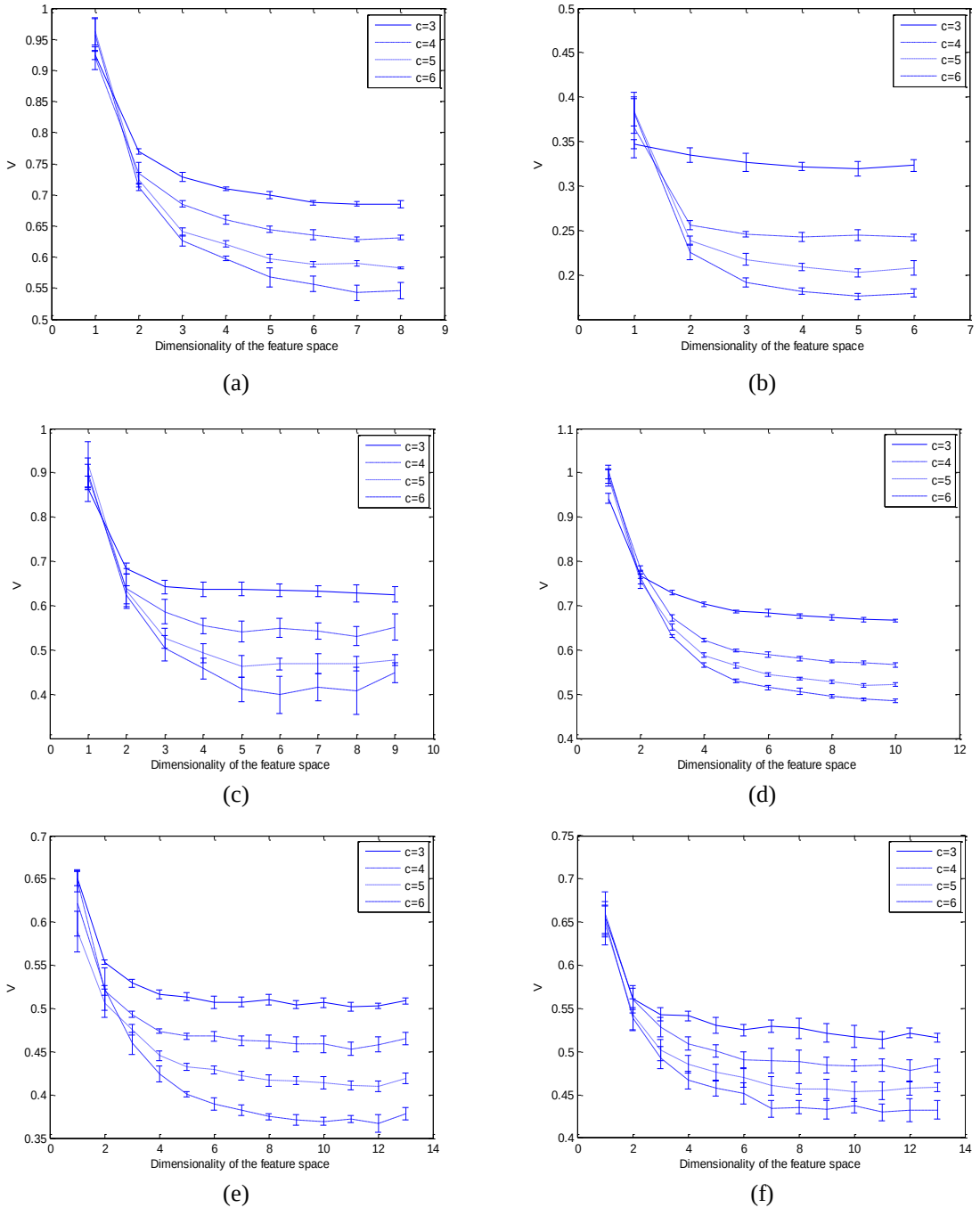


Figure 3-6: Plots of V versus the dimensionality of the features space for selected levels of granularity of information (number of clusters): (a) Pima dataset, (b) Auto dataset, (c) Vowel dataset, (d) Glass dataset (e) Housing dataset, and(f) Wine dataset. Shown are the average values of V as well as the standard deviations of the fitness function.

Table 3-2: The ratio of V to V_{ref} for all data sets

Dimensionality	V/V_{ref}					
	Auto	Pima	Glass	Vowel	Housing	Wine
4	1.01	1.10	1.02	1.17	1.12	1.08
5	0.98	1.04	0.92	1.09	1.06	1.06
6	1.00	1.02	0.89	1.06	1.03	1.04
7		0.99	0.93	1.05	1.01	1.00
8		1.00	0.91	1.02	0.99	1.01
9			1.00	1.01	0.98	1.00
10				1.00	0.98	1.01
11					0.98	0.99
12					0.97	1.00
13					1.00	1.00

Equally interesting are the resulting reduced feature spaces. The results are shown in Tables 3-3 to 3-6. Overall, the reduced feature space is almost the same for different numbers of cluster used, especially for data with dimensionality of the feature space being less 10. For example, Table 3-3 and Table 3-4 show that the features selected are almost the same for different number of clusters applied. However, referring to Table 3-5 and Table 3-6, we note that some of the selected feature spaces are not the same when considering different numbers of clusters.

Table 3-3: Best subsets of features for c=3, c=4, c=5 and c=6 (Pima data)

Dimensionality	c=3	c=4	c=5	c=6
1	4	4	4	4
2	4,8	4,8	4,8	**5**,8
3	4,5,8	4,5,8	4,5,8	4,5,8
4	1,4,5,8	1,4,5,8	1,4,5,8	1,4,5,8
5	1,2,4,5,8	1,2,4,5,8	1,2,4,5,8	1,2,4,5,8
6	1,2,4,5,6,8	1,2,4,5,6,8	1,2,4,5,6,8	1,2,4,5,6,8
7	1,2,4,5,6,7,8	1,2,4,5,6,7,8	1,2,4,5,6,7,8	1,2,4,5,6,7,8
8	all	all	all	all

Table 3-4: Best subsets of feature for c=3, c=4, c=5 and c=6 (Auto-MPG data)

Dimensionality	c=3	c=4	c=5	c=6
1	2	2	2	2
2	2,6	**1,**6	2,6	2,6
3	1,2,6	1,2,6	1,**5**,6	1,**5**,6
4	1,4,5,6	1,4,5,6	1,4,5,6	1,4,5,6
5	1,3,4,5,6	1,3,4,5,6	1,3,4,5,6	1,3,4,5,6
6	all	all	all	all

Table 3-5: Best subsets of feature for c=3, c=4, c=5 and c=6 (Glass data)

Dimensionality	c=3	c=4	c=5	c=6
1	4	8	8	8
2	1,8	3,7	1,3	1,3
3	1,2,8	1,3,8	3,7,9	3,7,9
4	1,2,3,8	1,2,3,8	1,3,8,9	1,3,8,9
5	1,2,3,8,9	1,2,3,8,9	1,2,3,8,9	1,2,3,8,9
6	1,2,3,4,5,7	1,2,3,7,8,9	1,2,3,7,8,9	1,2,3,7,8,9
7	1,2,3,4,5,7,8	1,2,3,4,5,7,8	1,2,3,5,7,8,9	1,2,3,4,5,7,9
8	1,2,3,4,5,7,8,9	1,2,3,4,5,7,8,9	1,2,3,4,5,7,8,9	1,2,3,4,5,7,8,9
9	all	all	all	all

Table 3-6: Best subsets of feature for c=3, c=4, c=5 and c=6 (Housing data)

Dimensionality	c=3	c=4	c=5	c=6
1	3	5	3	3
2	2,10	2,10	2,3	3,5
3	2,7,10	2,5,9	2,3,7	2,5,9
4	2,7,8,10	2,7,10,12	2,5,10,12	2,5,9,12
5	2,6,7,8,10	2,6,7,10,12	2,8,10,12,13	2,6,7,9,12
6	2,6,7,8,10,12	2,3,6,7,9,12	2,3,7,9,12,13	2,6,7,8,10,12
7	2,3,6,7,8,9,12	2,3,7,8,9,11,13	2,3,7,8,9,12,13	2,3,6,7,9,11,12
8	2,3,6,7,8,9,12,13	2,3,6,7,8,9,11,13	2,3,7,8,9,11,12,13	2,3,6,7,8,9,11,12
9	2,3,5,6,7,8,9,11,12	2,3,6,7,8,9,10,12,13	2,3,6,7,8,9,11,12,13	2,3,6,7,8,9,11,12,13
10	2,3,5,6,7,8,9,10,11,12	2,3,6,7,8,9,10,11,12,13	2,3,5,6,7,8,9,11,12,13	2,3,5,6,7,8,9,11,12,13
11	2,3,5,6,7,8,9,10,11,12, 13	2,3,5,6,7,8,9,10,11,12, 13	2,3,5,6,7,8,9,10,11,12, 13	2,3,5,6,7,8,9,10,11,12, 13
12	2,3,4,5,6,7,8,9,10,11,1 2,13	2,3,4,5,6,7,8,9,10,11,1 2,13	2,3,4,5,6,7,8,9,10,11,1 2,13	2,3,4,5,6,7,8,9,10,11,1 2,13
13	all	all	all	all

In most of the data sets, we observe that the subsets of the reduced feature spaces exhibit an interesting "nesting" property meaning that the extended feature space builds upon the feature space constructed so far. For instance, for the Pima data, we have the subsets of features

{feature 4} $\subset$ {feature 8, feature 4}
$\subset$ {feature 8, feature 5, feature 4},

where the corresponding features are: 4. Triceps skin fold thickness (mm), 8. Age (years), 5. 2-Hour serum insulin (mu U/ml).

For the Auto MPG data, we arrive at the series of nested sets of features:

{feature 2} $\subset$ {feature 6, feature 2} $\subset$ {feature 6, feature 2, feature 1}.

where the corresponding features are: 2. displacement, 6. model year, 1. Number of cylinders.
The optimal subsets of the reduced feature spaces (viz. the subsets for which V is close to V_{ref} and the increase of the feature space does not produce any significant reduction in the values of the reconstruction error)
In all cases, we can conclude that the original feature space exhibits redundancy and some features could be easily eliminated without causing any tangible increase in the values of the reconstruction error. The results are shown in Table 3-7 to 3-12.

Table 3-7: Best subsets of features – Pima data

Dimensionality	Best Feature Subset	V± σ
1	4	0.9572±0.0260
2	5 8	0.7130±0.0054
3	4 5 8	0.6265±0.0090
4	1 4 5 8	0.5978±0.0040
5	1 2 4 5 8	0.5674±0.0157
6	1 2 4 5 7 8	0.5568±0.0125
7	1 2 4 5 6 7 8	0.5425±0.0126
8	1 2 3 4 5 6 7 8	0.5458±0.0129

Table 3-8: Best subsets of features – Auto-MPG data

Dimensionality	Best Feature Subset	V± σ
1	2	0.3821±0.0228
2	2 6	0.2250±0.0084
3	1 5 6	0.1917±0.0049
4	2 3 5 6	0.1817±0.0035
5	1 3 4 5 6	0.1759±0.0035
6	1 2 3 4 5 6	0.1796±0.0046

Table 3-9: Best subsets of features - Glass data

Dimensionality	Best Feature Subset	V± σ
1	2	0.8996±0.0332
2	1 3	0.6241±0.0201
3	3 7 9	0.5038±0.0283
4	1 2 3 9	0.4578±0.0234
5	2 4 5 7 9	0.4109±0.0271
6	2 3 4 5 7 9	0.3985±0.0422
7	1 2 3 5 7 8 9	0.4161±0.0312
8	1 2 3 4 5 7 8 9	0.4078±0.0531
9	1 2 3 4 5 6 7 8 9	0.4484±0.0215

Table 3-10: Best subsets of features -Vowel data

Dimensionality	Best Feature Subset	V± σ
1	9	1.0021±0.0157
2	2 8	0.7628±0.0140
3	2 8 9	0.6309±0.0027
4	2 4 8 9	0.5657±0.0058
5	2 3 4 8 9	0.5297±0.0048
6	1 2 6 7 8 9	0.5149±0.0045
7	1 2 5 6 7 8 9	0.5063±0.0068
8	1 2 4 5 6 7 8 9	0.4954±0.0043
9	1 2 3 4 5 6 7 8 9	0.4883±0.0038
10	1 2 3 4 5 6 7 8 9 10	0.4843±0.0043

Table 3-11: Best subsets of features -Housing data

Dimensionality	Best Feature Subset	V± σ
1	3	0.6217±0.0382
2	3 5	0.5225±0.0244
3	2 5 9	0.4596±0.0128
4	2 5 10 12	0.4242±0.0092
5	2 6 7 9 12	0.4006±0.0035
6	2 6 7 8 10 12	0.3892±0.0071
7	2 3 6 7 9 11 12	0.3826±0.0062
8	2 3 6 7 8 9 11 12	0.3746±0.0037
9	2 3 6 7 8 9 11 12 13	0.3711±0.0058
10	2 3 5 6 7 8 9 11 12 13	0.3693±0.0046
11	2 3 5 6 7 8 9 10 11 12 13	0.3716±0.0041
12	2 3 4 5 6 7 8 9 10 11 12 13	0.3672±0.0102
13	1 2 3 4 5 6 7 8 9 10 11 12 13	0.3782±0.0071

Table 3-12: Best subsets of features - Wine data

Dimensionality	Best Feature Subset	V± σ
1	7	0.6510±0.0186
2	7 10	0.5382±0.0126
3	3 7 10	0.4930±0.0131
4	3 7 10 13	0.4670±0.0103
5	1 3 6 7 10	0.4577±0.0086
6	3 8 9 10 11 13	0.4516±0.0121
7	3 6 8 9 10 11 13	0.4342±0.0098
8	2 3 7 8 9 10 12 13	0.4354±0.0077
9	1 2 3 4 8 9 10 12 13	0.4328±0.0112
10	1 2 3 5 6 8 9 10 11 12	0.4374±0.0081
11	1 2 3 5 6 7 8 9 10 11 13	0.4299±0.0098
12	1 2 3 4 5 6 8 9 10 11 12 13	0.4320±0.0134
13	1 2 3 4 5 6 7 8 9 10 11 12 13	0.4326±0.0106

While in the minimization of the reconstruction error we did not use any class information (class labels, which could serve as a certain component of supervision), we can assess the performance

of the optimized reduced spaces by determining the corresponding classification error (or classification accuracy). Making use of the class labels, we assign each cluster to a certain class (which is determined by the majority of patterns allocated to the given cluster). Then we count the number of misclassified patterns, viz. the patterns, which were allocated to clusters with different class annotation than these patterns themselves).

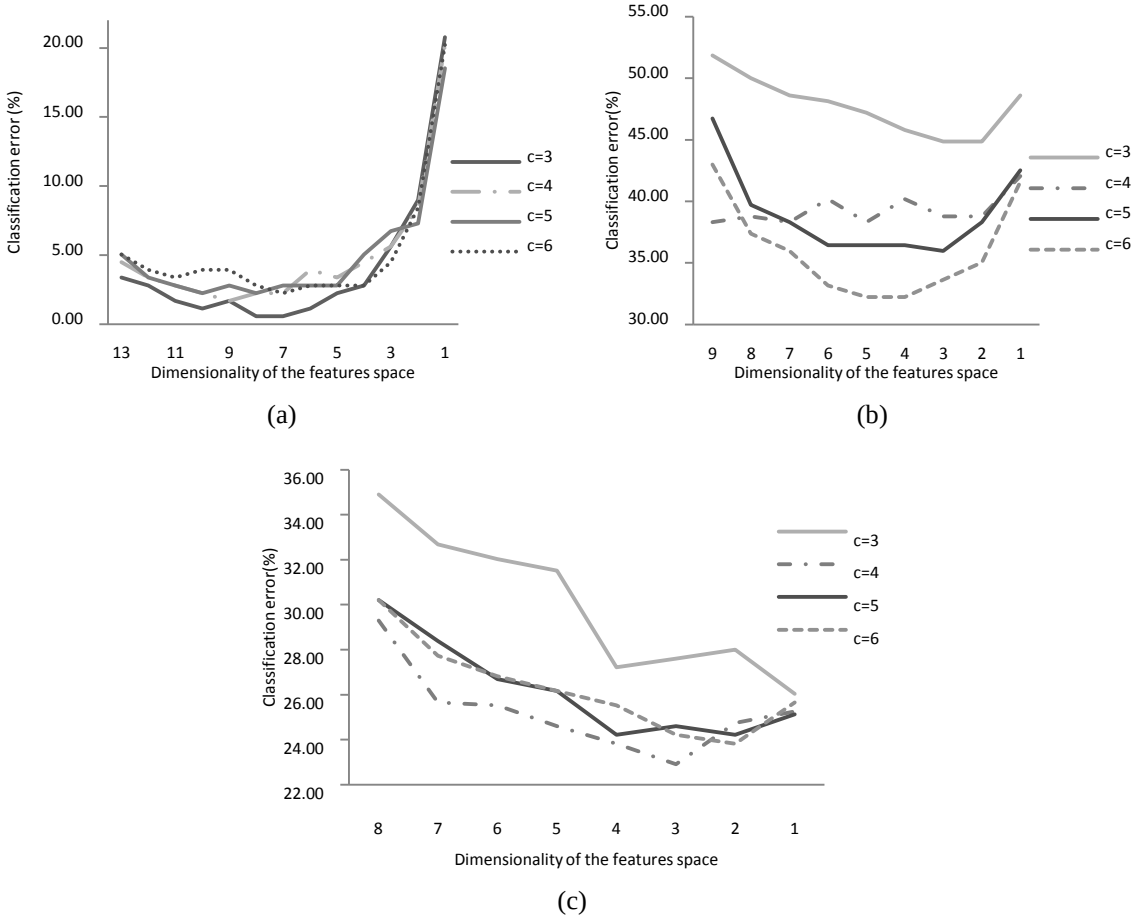


Figure 3-7: Plot of classification errors versus the dimensionality of the reduced feature space for selected levels of granularity (number of clusters, c): (a) Wine dataset, (b) Glass dataset, and (c) Pima dataset

The behavior of classification error varies from one data set to another. In some of them it is quite apparent that the use of the entire feature space is not beneficial at all. This happens for the Wine, Pima and Glass data sets (see Figure 3-7). In all these cases we observe that there is an optimal subset of features where the classification error attains a far lower value than the one reported for the entire feature space. The concise summary of classification results is reported in Table 3-13.

Table 3-13: Minimal classification errors in reduced feature spaces

Data	*Number of cluster*	*Classification error (all features)*	*Classification error (selected features)*	*Reduction of classification error*	*Number of selected features*	*Set of selected features*
Wine	3	3.37	1.00	0.30	7	{1,3,4,6,7,10,13}
Glass	6	42.99	32.24	0.74	4	{3,4,5,8}
Pima	4	29.30	22.92	0.78	3	{2,6,7}

Along with the use of GA as the optimization vehicle, we experimented with another population-based technique that is PSO. To arrive at a sound comparative framework, the numbers of generations as well as the size of the population were the same as in case of GA. Two data sets were experimented with, that Pima and Auto MPG. Our intent is to compare the results produced by the two methods as well as compare the computational effectiveness of the methods themselves. The obtained results in terms of the best subsets of features are reported in Tables 3-14 and 3-15 are the same as those produced by the GA.

Table 3-14: Best subsets of features (c=4) – Pima data

Dimensionality	Best Feature Subset	V± σ
1	4	0.9103±0.0260
2	5 8	0.7466±0.0054
3	4 5 8	0.6773±0.0090
4	1 4 5 8	0.6603±0.0040
5	1 2 4 5 8	0.6420±0.0157
6	1 2 4 5 7 8	0.6263±0.0125
7	1 2 4 5 6 7 8	0.6300±0.0126
8	1 2 3 4 5 6 7 8	0.6315±0.0129

Table 3-15: Best subsets of features (c=6) – Auto MPG data

Dimensionality	Best Feature Subset	V± σ
1	2	0.4191±0.0497
2	2 6	0.2544±0.0155
3	1 5 6	0.2026±0.0159
4	2 3 5 6	0.1853±0.0210
5	1 3 4 5 6	0.1843±0.0103
6	1 2 3 4 5 6	0.1796±0.0046

From the computational point of view, the PSO approach is more efficient as the convergence of the search requires a significantly lower number of generations, see Figure 3-8. For instance, PSO required 10 generations versus 45 generations required by the GA optimization. Furthermore as the overall computing overhead of the PSO is lower, we see significant reductions in the overall computing time as illustrated in Table 3-16.

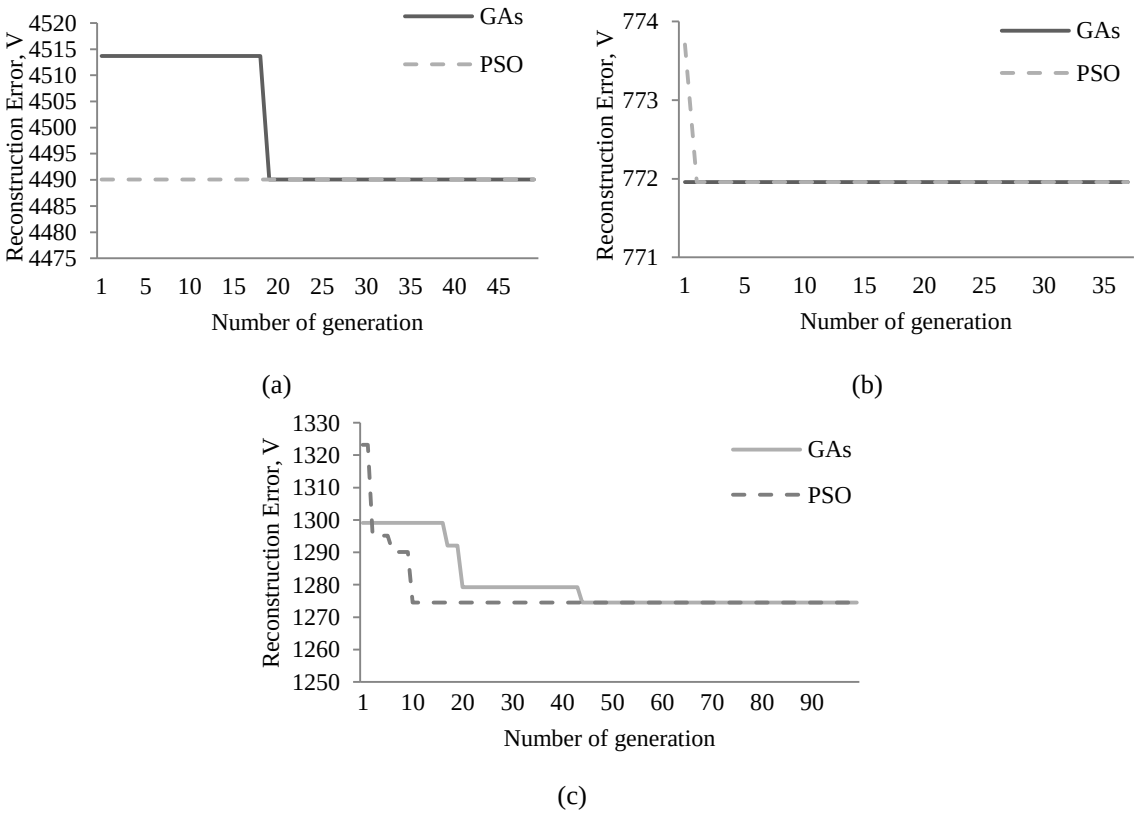


Figure 3-8: Reconstruction error in successive generations; (a) Pima dataset, (b) Auto MPG dataset, and (c) Wine dataset

Table 3-16: Computing time (in seconds) used in the PSO and GA optimization

Data set	PSO	GA
Pima	6714.2595	7240.5593
Auto MPG	854.94512	1591.6648

3.5 Conclusions

In this chapter, we proposed a way of feature selection based on the concept of structure retention, which is quantified in terms of the reconstruction criterion. The granulation-degranulation mechanism forms the underlying conceptual vehicle using which the quality of the granules in the original and the reduced feature space is quantified. The retention of the original structure is optimized (maximized) by solving a combinatorial selection problem of forming a reduced feature space. The mechanisms of Genetic Algorithms were used to solve the problem. The experiments identified several interesting findings. First, it is noticeable that the feature space could be quite significantly reduced with a minor deterioration of the reconstruction capabilities and this reduction does not depend upon the number of clusters themselves. Second, the sequence of successively reduced feature spaces exhibit an appealing "nesting" property in which the extended spaces are built upon the previous reduced versions.

There are a number of possible extensions and generalizations of the generic version of the structure retention – based feature selection. The concept could be cast in the setting of any clustering as the granulation-degranulation scheme applies (after some refinement) to any scheme of information granulation (and the ensuing clustering technique). Some further pursuits in investigating structural stability of clusters deserve more attention.

4.Feature and Data Reduction in Fuzzy Modeling via Cooperative PSO

In this chapter, a comprehensive framework is proposed to construct fuzzy models from the subset of numerical input-output data. First, we develop a data-driven fuzzy modeling framework for a high dimensional large dataset, which is capable of generating a rule-base automatically from numerical data. Second, we integrate the concept of feature selection and data selection together in the unified form to further refine the fuzzy models. In this regard, the PSO technique is applied in order to search for the best subset of data. In order to increase the effectiveness of the PSO techniques, we introduce a new Cooperative PSO method based on the information granulation approach. Third, we develop a flexible setup to cope with the optimization of variables and data to be used in the design of the fuzzy model. The proposed approach allows the user to choose the predetermined fraction of variables and data that can be used to construct the fuzzy models. This chapter is organized as follows. First, we give some introduction about feature and data reduction in Section 4.1. Next, we briefly elaborate on the selected approaches to data and feature space reduction (Section 4.2). The proposed fuzzy modeling framework along with its main algorithmic developments is presented in Section 4.3. Experimental studies are presented in Section 4.4. Then in Section 4.5 we proposed feature and data selection for Nearest Neighbors classification via the Cooperative PSO and conclusions are provided in Section 4.6.

4.1 Feature and data reduction

The data and feature reduction activities are advantageous to fuzzy models in terms of both the effectiveness of their construction and the interpretation of the resulting models, their realization deserves particular attention. The formation of a subset of meaningful features and a subset of essential instances is discussed in the context of fuzzy rule-based models. The dimensionality problem can be addressed by reducing the constructed fuzzy rules. The reduction method plays two important roles. It increases the effectiveness of the learning algorithm, since the learning algorithm will concentrate only on the most useful subset of data. It also improves the computational efficiency as the learning algorithm involves only a subset of data smaller than the original dataset (M. Setnes, R. Babuska, et al. 1998). This reduction can be realized by removing the redundant fuzzy rules by exploiting a concept of fuzzy similarity (Jin 2000, M. Setnes, R. Babuska, et al. 1998, Chen and Linkens 2004). Evolutionary algorithms have also been used for

building compact fuzzy rules (Wang, et al. 2005, Berlanga, et al. 2010, Alcala-Fdez and R. Alcala 2010, Chen, et al. 2007). An evolutionary algorithm is used to tune the structure and the rules' parameter of the fuzzy systems (Delgado, Zuben and Gomide 2004, Xiong and Litz 2002). However, in numerous cases, some variables are not crucial to the realization of the fuzzy model. A suitable way to overcome this problem is to implement feature selection before constructing the fuzzy models. Therefore, during the last decade, feature selection methods in conjunction with constructing fuzzy models for reducing the curse of dimensionality were developed (Gaweda, Zurada and Setiono 2001, Hadjili and Wertz 2002, Sindelar and Babuska 2004, Zarandi, Turken and Rezaee 2004, Du and Zhang 2008, Ghazavi and Liao 2008, Zhang, et al. 2011, Wan, et al. 2005). This process reduces the fuzzy rule search space and increases the accuracy of the model.

As mentioned above, forming the best input data as the training set to construct the fuzzy modeling is also important. However, as far as we know there is no research that have been done to simultaneously select the best subset of features and input data for constructing the fuzzy model. Most of the research is focused on reducing the fuzzy rules and the process of simplifying the system is done once the design has been completed. Here we propose a method that reduces the complexity of the system starting from the design stage. Whereas, the process of constructing the antecedent and the consequent parts of the fuzzy model are realized using the best subset of input data.

4.2 Selected approaches to feature and data reduction

In general, reduction processes involve Feature Selection (FS), Instances (data) Selection (IS), and a combination of these two reduction processes: Feature and Instances Selection (FIS). Feature selection is a subject of the main reduction pursuits. The goal of FS, which is commonly encountered in problems of system modeling and pattern recognition, is to select the best subset of features so that the model formed in this new feature (input) space exhibits the highest accuracy (classification rate) being simultaneously associated with the increased transparency of the resulting construct (Guyon and Elisseeff 2003). The process aims to discard irrelevant and/or redundant features (Lui and Yu 2005, Blum and Langley 1997). The reader can refer to (Guyon and Elisseeff 2003, Lui and Yu 2005, Lui and Motoda 2008, Chui 1996) for more details.

Instances selection (IS), another category of reduction approaches, is concerned with the selection of the relevant data (instances) reflective of the knowledge pertinent to the problem at hand (Olvera-Lopez, et al. 2010, Lui and Motoda, Instance Selection and Construction for Data Mining 2001). The three main functions forming the essence of IS includes enabling, focusing and cleaning.

In this study, as stated earlier, instead of approaching feature selection and instances selection separately, we focus on the integration of feature selection and instances selection in the construction of the fuzzy models. Both processes are applied simultaneously to the initial dataset, in order to obtain a suitable subset of feature and data to construct the parameters for the fuzzy model. In the literature, some methods for integrating feature and instances selection are more focused on a class of classification problems (Ishibuchi, Nakashima and Nii 2001, Derrac, Garcia and Herrera 2009).

The ideas of feature and data reduction as well as hybrid approaches have been discussed in the realm of fuzzy modeling. Table 4-1 offers a snapshot at the diversity of the existing approaches and the advantages gained by completing the reduction processes.

Table 4-1: A summary of selected studies in data and feature reduction in fuzzy modeling

Reference	Feature reduction technique	Dataset, fuzzy model and data	Original data used in modeling		Number of selected features	Number of resulting rules
			number of instances	number of features		
(Gaweda, Zurada and Setiono 2001)	The use of sensitivity analysis - determination of essential features	Box-Jenkins gas furnace	296	10	3	2
(Hadjili and Wertz 2002)	Deviation criterion (DC): to measure the change in fuzzy partition. Removal of features that do not significantly change the fuzzy partition	Nonlinear systems in noisy environment	250	3	1	4
		Nonlinear dynamical system excited by a sinusoidal signal	800	10	6	8
		Run-out cooling table in a hot strip mill	1000	17	5	12
(Zarandi, Turken and Rezaee 2004)	Heuristic method to select features.	Nonlinear System	50	4	2	4
		Supplier Chance Management dataset	300	9	5	5
(Du and Zhang 2008)	Evolutionary optimization	Box-Jenkins gas furnace	296	10	3	4
		MR damper	5000	11	6	10

		identification				
(Ghazavi and Liao 2008)	1.Mutual correlation methods 2. Gene selection criteria 3. The relief algorithm	Wisconsin breast cancer	569	30	3	250 (3)
		PIMA Indian diabetes	768	8	3	125 (3)
		Welding flaw identification	399	25	3	-
(Zhang, et al. 2011)	Iterative Search Margin Based Algorithm (Simba)	Wisconsin breast cancer	699	9	5	3
		Wine	178	13	4	5
		Iris	150	4	3	3
		Ionosphere	351	34	10	4

4.3 PSO-integrated feature and data reduction in fuzzy rule-based models

Some recent studies (Ishibuchi, Nakashima and Nii 2001, Derrac, Garcia and Herrera 2009, Cano, Herrera and Lozano 2003) have employed population-based optimization techniques to carry out search for the best subset of variables and data for solving the application problems, but all of them were carried out to solve the classification problem. Therefore, in this study, we use population-based technique for selecting the best subset of feature and data for the regression problem. Here, we implement Particle Swarm Optimization (PSO) techniques to intelligently search for the best subset of features and data (instances).

In this research, we employed the PSO-based method to handle two optimization tasks namely, (1) a selection of the optimal subset of features and (2) a selection of the optimal subset of instances based on the concept of information granularity. In order to reduce the computational complexity of using the standard PSO, we employed Cooperative PSO method to simultaneously solve the two optimization tasks. The motivation behind the use of cooperative PSO, as advocated in (van den Bergh and Engelbrecht 2004), is to deal effectively with the dimensionality of the search space, which becomes a serious concern when a large number of data with a large dimensionality are involved. This curse of dimensionality is a significant impediment negatively impacting the effectiveness of standard PSO. The essence of the cooperative version of PSO is essentially a parallel search for optimal subset of features and its optimal subset of instances. The cooperative strategy is achieved by dividing the candidate solution vector into components, called

sub-swarm, where each sub-swarm represents a small part of the overall optimization processes. By doing this, we implement the concept of divide and conquer to solve the optimization problem, so that the process will become more efficient and fast (van den Bergh 2002).

The essence of the cooperative version of PSO is to split the data into several groups so that each group is handled by a separate PSO. The main design question involves splitting the variables into groups. A sound guideline is to keep the related (associated) variables within the same group. Obviously, such relationships are not known in advance. Several possible methods are available for addressing this issue on more detail in the context of the problem at hand.

(a) As we are concerned with a collection of features and data (instances), a natural way to split the variables would be to form two groups (K=2), one for the features (n) and another one for the instances (M). This split would be legitimate if the dimensionality of both subsets were quite similar.

(b) In some situations, one of the subsets (either the data or the features) might be significantly larger than the other one. We often encounter a large number of data, but in some situations, a large number of features might be present (for instance, in microarray data analysis). This particular collection of data or features is then split into K groups. Clustering such items is a viable algorithmic approach. Running K-Means or Fuzzy C-Means produces clusters (group) of variables that are used in the individual PSO.

(c) In case both subsets are large, the clustering is realized both for the features and data and the resulting structure (partition) is used to run cooperative PSO

As the problem of feature-data reduction is inherently combinatorial nature, PSO provides an interesting and computationally viable optimization alternative. In the following sub-sections, we start with a general optimization setting and then discuss the PSO realization of the search process (here, a crucial design phase is a formation of the search space with a suitable encoding mechanism). Although the proposed methodology is of a general nature, we concentrate on rule-based models, which are commonly present in fuzzy modeling, to help offer a detailed view of the overall design process.

4.3.1 An overall reduction process

As is usual in system modeling, we consider a supervised learning scenario in which we encounter in a finite set of training data $(\mathbf{x}_k, t_k)$, k=1, 2,…, M. By stressing the nature of the data and their dimensionality, the data space along with n-dimensional feature vectors can be viewed as a Cartesian product of the data and features $\mathbf{D} \times \mathbf{F}$. The essence of the reduction is to arrive at

the Cartesian product of the reduced data and feature spaces, **D**' × **F**', where, **D'** ∈ **D** and **F**' ∈ **F**. The cardinality of the reduced spaces is equal to M' and n' where M' < M and n' <n.

The overall scheme of the reduction process outlining a role of the PSO-guided reduction is illustrated in Figure 4-1. The scheme can be divided into two important parts and can be described as follows:

(a) Reduction process via PSO: A reduction process tackles both feature reduction and data reduction simultaneously. PSO algorithm is use to search for the best feature and data for constructing the fuzzy model. The size of the selected features (n') and data (M') is provided in advance by the user. After the PSO meets the maximum generation, the process is stopped, and the last best subset of features and data is the best subset of data for constructing the fuzzy model.

(b) Evaluation process: The Fuzzy C-Means algorithm is used to convert the numerical data into the information granules. Here, the information granulation process deals only with the subset of the data and features (**D'** x **F'**). Next, the consequent parameter **a** constructed from the fuzzy models is used to evaluate the performance of the selected data and features. At this stage we access the performance of the constructed fuzzy model in terms of their capability to fit the model by using the all instances in the original data set.

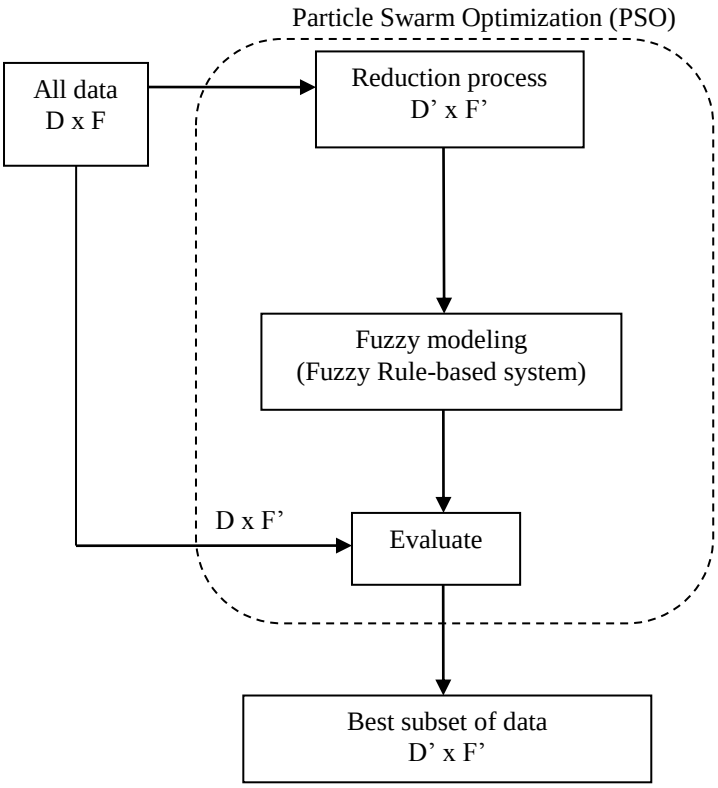


Figure 4-1: The scheme of the proposed data and reduction for fuzzy modeling

As it becomes apparent, the original space $\mathbf{D}\times\mathbf{F}$ is reduced, and in this Cartesian product a fuzzy model, denoted by FM, is designed in the usual way (we elaborate on the form of the fuzzy model in the subsequent section). Its design is guided by a certain objective function Q expressed over all elements of original instances. The quality of the reduced space is assessed by quantifying the performance of the fuzzy model operating over the original, non-reduced space. The same performance index as used in the construction of the fuzzy model in the reduced space is used to describe the quality of the fuzzy model:

$$Q = \sqrt{\frac{1}{M}\sum_{\mathbf{x}_k \in D\times F'} (FM(\mathbf{x}_k) - t_k)^2} \tag{4-1}$$

Note that the summation shown above is taken over all the elements forming the data space **D**. By taking another look at the overall reduction scheme, it is worth noting that the reduction is realized as in the wrapper mode, in which we use a fuzzy model to evaluate the quality of the reduction mechanism.

4.3.2 The PSO –based representation of the search space

The reduction of the data and feature spaces involves a selection of a subset of the data and a subset of the features. Therefore, the problem is combinatorial in its nature. PSO is used here to form a subset of integers which are indexes of the data or features to be used in the formation of **D**'× **F**'. For instance, **D**' is represented as a set of indexes $\{i_1, i_2, \ldots, i_{M'}\}$ being a subset of integers {1, 2, …, M}. From the perspective of the PSO, the particle is formed as a string of real numbers in [0, 1] of the length of n+M, effectively, the search space is a hypercube $[0,1]^{n+M}$. The first substring of length n represents the features; the second one (having M entries) is used to optimize the subset of data. The particle is decoded as follows. Each substring is processed (decoded) separately. The real number entries are ranked. The result is a list of integers viewed as the indexes of the data. The first M' entries out of the M-position substring are selected to form **D**'. The same process is applied to the substring representing the set of features. An overall decoding scheme is illustrated in Figure 4-2.

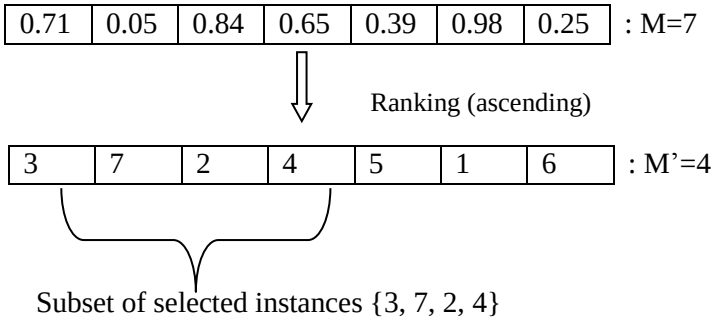


Figure 4-2: From a particle in $[0,1]^M$ search space to a subset of instances

The information given by the PSO is used to represent the subset of features and data to construct the data-driven fuzzy models. Then, the numerical data are represented in terms of a collection of information granules (a fuzzy sets) produced through some clustering (fuzzy clustering). The information about the granules (clusters) is then used to construct the fuzzy models.

In the cooperative PSO, the formation of the search space is realized in a more sophisticated way. The cooperative facet involves mainly exchanging information about the best positions found by the different sub-swarms. Here, we present a new cooperative PSO (CPSO) algorithm for the data and feature reduction process. The selection of the number of cooperating swarms is important because it will affect the performance of the cooperative PSO model. Sub-swarm 1 represents the features' column and sub-swarm 2 represents the instances' row of the particular data set. Figure 4-3 illustrates the main difference between standard PSO and cooperative PSO. The standard PSO contains one swarm with a large dimension of search space. In contrast, for the

cooperative PSO, we divide the search space into two sub-swarms: sub-swarm 1 for feature representation and sub-swarm 2 for instances representation. All the sub-swarms share the same basic particles definition illustrated in Figure 4-2.

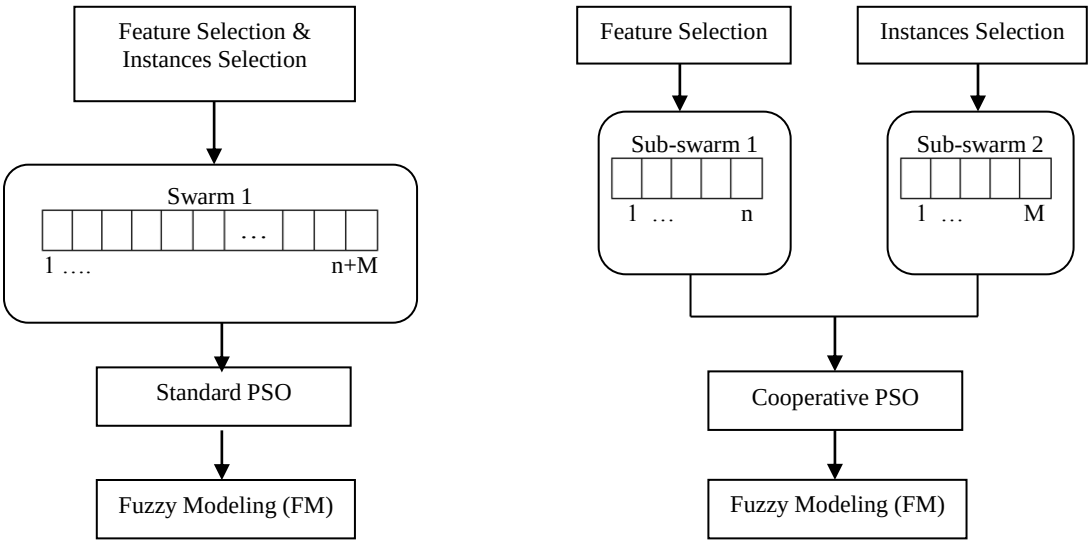


Figure 4-3: The particle scheme of the "standard" PSO (a) and cooperative PSO (b)

In general, the dimensionality for the data (instances) selection is higher than that of the feature selection. In order to reduce the impact of the curse of dimensionality, we decompose the data into several groups by using the information granulation approach. In this research, we used the Fuzzy C-Means (FCM) to construct the information granules. Therefore, the number of decomposition groups is actually the number of the clusters (C) used in the FCM. For example, if we want to decompose the data into three groups, we use the number of clusters equal to three. As a result, instead of having only two sub-swarms, we introduce more sub-swarms that represent different groups of data (see Figure 2-9). As mentioned earlier, we apply the concept of information granulation to decompose the data group. In order to identify the selected data in each decomposed group, we use the information granules (membership degrees) values to identify the index of the instances in each group. We employ a winner-takes-all scheme to determine a single group for each granule, i.e. the index of the instances in each of the decomposition group related to the information granule that gets the highest degrees of activation. We denote the set of data associated with the i-th granules by X_{io};

$$X_{io} = \{ x_k \in X \mid U_{iok} = \max_i U_{ik} \} \text{ for } 1 \leq k \leq M \text{ and } 1 \leq i \leq c \quad , \tag{4-2}$$

where X_{io} is the decomposition groups, U_{ik} is the information granules for each data, x_k is the data (instances), M and c are the number of data and the level of information granulation, respectively.

4.4 Experimental Studies

In this section, we report our results from a set of experiments, using several Machine Learning data sets (see http://www.ics.uci.edu/~mlearn/MLRepository.html and http://lib.stat.cmu.edu/datasets/). The main objective of these experiments is to show the abilities of the proposed approach, quantify the performance of the selected subsets of features and instances, and arrive at some general conclusions. A concise summary of the data sets used in the experiment is presented in Table 4-2. All the data concern continuous output.

Table 4-2: Description of data used in the experiments (S is the ratio of the number of data versus the number of features)

Data set	*Abbreviation*	*Number of features*	*Number of data*	*Sparsity ratio, S*
Air Pollution PM10	PM10	7	500	71.43
Boston Housing	Housing	13	506	38.92
Body Fat	Body Fat	14	252	18.00
Parkinson Tele-monitoring	Parkinson	17	5875	345.59
Computer Activity	Computer	21	8192	390.09

4.4.1 Parameter setup

The values of the PSO and CPSO parameters were set using the standard form as follows. The values of the inertia weight, w were linearly from 1 to 0 over the course of optimization. The values of the cognitive factor, c_1 and social factor c_2 were set to 1.49 and 1.49, respectively. In Table 4-3, we also list the numeric values of the parameters of the PSO and CPSO environment. As to the size of the population and the number of generations, we used a larger population and a larger number of generations in the generic version of the PSO than in the CPSO because of the larger search space this algorithm operates in.

The number of sub-swarms used in the optimization method is also shown in Table 4-3. The PSO method comprises only a single swarm whose individuals concatenate features and instances. In contrast, for the CPSO, we divided the search space into several sub-swarms that can cooperate with each other and where the individuals in the sub-swarms are used to represent a portion of the search space. The $CPSO^1$ contains two sub-swarms that cover the data and features, respectively. In $CPSO^{2,}$ we used three sub-swarms to represent data point; in the data used here, the number of data is larger than the number of features, so a better balance of the dimensionality of the spaces is achieved. The data (instances) search space is divided into three sub-swarms, and the decomposition process is realized by running fuzzy clustering (each cluster forms a sub-swarm). In the table we used a smaller size of generation compared to particles size. This is because in (Shi and Eberhart 1999) Shi and Eberhart mentioned that the population size does not

exhibit any significant impact on the performance of the PSO method. However, the size of particles is high given the size of the search space. Here we require more particles to capture the large search space of instances selection for using the standard PSO. As a result we can find the best solution faster than using a smaller particles size. On the other hand, the number of particle is decreased when we implement the CPSO method. This is because the original large search space is divided into several groups and the processes of searching the best subset are done in parallel.

Table 4-3: The values of the parameters used in the experiments; $CPSO^1$ – swarms located in the feature space. $CPSO^2$ – swarms located in the instance (data) space

Optimization Method	*Sub -swarms*	*Generation*	*Particles*
PSO	1	50	300
$CPSO^1$	2	50	100
$CPSO^2$	4	30	50

4.4.2 Results of the experiments

In the experiments, we looked at the performance – an average root mean squared error (RMSE) obtained for the selected combinations of the number of features and data (instances). The results obtained for the Housing data, PM10 data, and Parkinson data for c = 4 and c = 3 clusters are summarized in Tables 4-4 to 4-6, respectively. The experiments were repeated 10 times, and the reported results are the average RMSE values. We also report the values of the standard deviation of the performance index to offer a better insight into the variability of the performance. It is noticeable that the standard deviation is reduced with the increase of the data involved and the decrease of the dimensionality of the feature space.

Table 4-4: Results for Housing data; the number of clusters is set to 4, c=4; S is the ratio of the number of selected data versus the number of selected features

Feature	*Data=10%* *(#of data=51)*		*Data=20%* *(#of data=101)*		*Data=30%* *(#of data=152)*		*Data=40%* *(#of data=202)*	
	S	RMSE	S	RMSE	S	RMSE	S	RMSE
10%	51.0	6.341 ± 0.253	101.0	6.262 ± 0.171	152.0	5.777 ± 0.207	202.0	6.227 ± 0.351
20%	17.0	6.664 ± 0.233	33.7	5.389 ± 0.253	50.7	5.191 ± 0.185	67.3	4.884 ± 0.118
30%	12.8	6.127 ± 0.245	25.3	5.468 ± 0.290	38.0	4.853 ± 0.183	50.5	4.574 ± 0.078
40%	10.2	6.321 ± 0.518	20.2	5.122 ± 0.245	30.4	4.626 ± 0.075	40.4	4.362 ± 0.190
50%	7.3	7.126 ± 0.835	14.4	5.046 ± 0.312	21.7	4.574 ± 0.206	28.9	4.018 ± 0.109
60%	6.4	8.133 ± 0.782	12.6	5.120 ± 0.189	19.0	4.504 ± 0.207	25.3	4.052 ± 0.196
70%	5.7	9.379 ± 0.984	11.2	5.003 ± 0.232	16.9	4.345 ± 0.134	22.4	3.949 ± 0.125
80%	5.1	10.57 ± 2.251	10.1	5.107 ± 0.262	15.2	4.232 ± 0.173	20.2	3.67 ± 0.093
90%	4.3	24.05 ± 7.681	8.4	5.324 ± 0.207	12.7	4.173 ± 0.181	16.8	3.809 ± 0.080
100%	3.9	44.39 ± 17.65	7.8	5.409 ± 0.201	11.7	4.082 ± 0.047	15.5	3.781 ± 0.055

	Data=50% *(#of data=253)*		*Data=60%* *(#of data=304)*		*Data=70%* *(#of data=354)*		*Data=80%* *(#of data=405)*	
	S	RMSE	S	Feature	S	RMSE	S	RMSE
10%	253.0	6.389 ± 0.063	304	6.483 ± 0.214	354.0	6.610 ± 0.211	405.0	6.387 ± 0.026
20%	84.3	4.805 ± 0.047	101	4.882 ± 0.080	118.0	4.906 ± 0.040	135.0	5.011 ± 0.077
30%	63.3	4.619 ± 0.294	76	4.45 ± 0.092	88.5	4.398 ± 0.035	101.3	4.56 ± 0.053
40%	50.6	4.172 ± 0.123	60.8	4.06 ± 0.099	70.8	4.126 ± 0.105	81.0	4.18 ± 0.177
50%	36.1	3.916 ± 0.224	43.4	3.927 ± 0.089	50.6	4.009 ± 0.093	57.9	4.085 ± 0.132
60%	31.6	3.912 ± 0.120	38	3.935 ± 0.129	44.3	3.934 ± 0.035	50.6	3.923 ± 0.117
70%	28.1	3.721 ± 0.071	33.8	3.722 ± 0.065	39.3	3.722 ± 0.066	45.0	3.787 ± 0.046
80%	25.3	3.617 ± 0.108	30.4	3.659 ± 0.128	35.4	3.568 ± 0.064	40.5	3.567 ± 0.086
90%	21.1	3.652 ± 0.050	25.3	3.569 ± 0.028	29.5	3.555 ± 0.017	33.8	3.533 ± 0.016
100%	19.5	3.687 ± 0.038	23.4	3.654 ± 0.015	27.2	3.631 ± 0.021	31.2	3.615 ± 0.011

Feature	*Data=90%* *(#of data=455)*		*Data=100%* *(#of data=506)*	
	S	RMSE	S	RMSE
10%	455.0	6.655 ± 0.061	506	7.437 ± 0
20%	151.7	5.149 ± 0.071	169	5.039 ± 0.096
30%	113.8	4.536 ± 0.026	127	4.578 ± 0.01
40%	91.0	4.343 ± 0.026	101	4.233 ± 0.079
50%	65.0	4.077 ± 0.092	72.3	4.005 ± 0.082
60%	56.9	3.931 ± 0.102	63.3	3.803 ± 0.088
70%	50.6	3.799 ± 0.041	56.2	3.668 ± 0.043
80%	45.5	3.645 ± 0.065	50.6	3.526 ± 0.049
90%	37.9	3.541 ± 0.015	42.2	3.550 ± 0
100%	35.0	3.605 ± 0.015	38.9	4.023 ± 0

Table 4-5: Results for PM10 dataset -c=3

Feature	*Data=10%* *(#of data=50)*		*Data=20%* *(#of data=100)*		*Data=30%* *(#of data=150)*		*Data=40%* *(#of data=200)*	
	S	RMSE	S	RMSE	S	RMSE	S	RMSE
10%	50.0	0.931 ± 0.036	100.0	0.979 ± 0.018	150.0	0.983 ± 0.006	200.0	0.985 ± 0.010
20%	50.0	0.896 ± 0.034	100.0	0.98 ± 0.013	150.0	0.987 ± 0.009	200.0	0.994 ± 0.008
30%	25.0	0.825 ± 0.087	50.0	0.902 ± 0.04	75.0	0.918 ± 0.007	100.0	0.916 ± 0.003
40%	16.7	0.829 ± 0.023	33.3	0.877 ± 0.007	50.0	0.877 ± 0.010	66.7	0.862 ± 0.009
50%	12.5	0.802 ± 0.029	25.0	0.816 ± 0.008	37.5	0.822 ± 0.012	50.0	0.826 ± 0.028
60%	12.5	0.804 ± 0.027	25.0	0.818 ± 0.013	37.5	0.825 ± 0.014	50.0	0.834 ± 0.016
70%	10.0	0.783 ± 0.030	20.0	0.781 ± 0.031	30.0	0.804 ± 0.017	40.0	0.782 ± 0.039
80%	8.3	0.768 ± 0.024	16.7	0.769 ± 0.007	25.0	0.776 ± 0.017	33.3	0.768 ± 0.024
90%	8.3	0.774 ± 0.026	16.7	0.771 ± 0.017	25.0	0.767 ± 0.014	33.3	0.796 ± 0.010
100%	7.1	0.786 ± 0.012	14.3	0.758 ± 0.017	21.4	0.764 ± 0.007	28.6	0.765 ± 0.015

Feature	*Data=50%* *(#of data=250)*				*Data=60%* *(#of data=300)*				*Data=70%* *(#of data=350)*				*Data=80%* *(#of data=400)*			
	S	RMSE			S	RMSE			S	RMSE			S	RMSE		
10%	250.0	0.998	±	0.021	300.0	1.036	±	0.022	350.0	1.071	±	0.016	400.0	1.088	±	0.002
20%	250.0	1.004	±	0.018	300.0	1.025	±	0.032	350.0	1.075	±	0.009	400.0	1.090	±	0.003
30%	125.0	0.918	±	0.010	150.0	0.920	±	0.005	175.0	0.922	±	0.003	200.0	0.937	±	0.001
40%	83.3	0.865	±	0.004	100.0	0.869	±	0.008	116.7	0.887	±	0.004	133.3	0.892	±	0.006
50%	62.5	0.843	±	0.024	75.0	0.870	±	0.023	87.5	0.891	±	0.022	100.0	0.906	±	0.004
60%	62.5	0.841	±	0.006	75.0	0.879	±	0.012	87.5	0.898	±	0.006	100.0	0.897	±	0.006
70%	50.0	0.806	±	0.014	60.0	0.832	±	0.016	70.0	0.851	±	0.010	80.0	0.856	±	0.004
80%	41.7	0.796	±	0.010	50.0	0.805	±	0.016	58.3	0.826	±	0.013	66.7	0.839	±	0.007
90%	41.7	0.777	±	0.019	50.0	0.815	±	0.017	58.3	0.820	±	0.014	66.7	0.843	±	0.001
100%	35.7	0.772	±	0.016	42.9	0.795	±	0.004	50.0	0.808	±	0.010	57.1	0.818	±	0.005

Feature	*Data=90%* *(#of data=450)*				*Data=100%* *(#of data=500)*			
	S	RMSE			S	RMSE		
10%	450.0	1.099	±	0.003	500.0	1.116	±	0
20%	450.0	1.098	±	0.003	500.0	1.116	±	0
30%	225.0	0.948	±	0.005	250.0	0.964	±	0
40%	150.0	0.902	±	0.002	166.7	0.915	±	0
50%	112.5	0.905	±	0.002	125.0	0.925	±	0
60%	112.5	0.907	±	0.002	125.0	0.925	±	0
70%	90.0	0.864	±	0.003	100.0	0.900	±	0
80%	75.0	0.847	±	0.001	83.3	0.878	±	0
90%	75.0	0.851	±	0.003	83.3	0.878	±	0
100%	64.3	0.824	±	0.003	71.4	0.883	±	0

Table 4-6: Results for Parkinson data- c = 3

Feature	*Data=10%* *(#of data=346)*				*Data=20%* *(#of data=691)*				*Data=30%* *(#of data=1037)*				*Data=40%* *(#of data=1382)*			
	S	RMSE			S	RMSE			S	RMSE			S	RMSE		
10%	346	6.388	±	0.182	691	6.221	±	0.064	1037	6.393	±	0.140	1382	6.495	±	0.106
20%	173	6.183	±	0.110	346	5.857	±	0.031	519	5.932	±	0.023	691	5.972	±	0.008
30%	115	6.152	±	0.151	230	5.799	±	0.061	346	5.703	±	0.067	461	5.708	±	0.045
40%	86	6.328	±	0.456	173	5.900	±	0.087	259	5.886	±	0.342	346	6.175	±	0.358
50%	69	6.991	±	0.525	138	7.585	±	0.755	207	6.448	±	0.580	276	7.493	±	0.303
60%	58	8.357	±	0.164	115	8.088	±	0.028	173	8.069	±	0.078	230	7.960	±	0.021
70%	49	8.401	±	0.068	99	8.074	±	0.064	148	8.087	±	0.015	197	8.008	±	0.011
80%	43	8.419	±	0.091	86	8.257	±	0.048	130	8.187	±	0.017	173	8.092	±	0.020
90%	38	8.500	±	0.106	77	8.258	±	0.024	115	8.199	±	0.017	154	8.139	±	0.013
100%	35	8.560	±	0.026	69	8.249	±	0.033	104	8.262	±	0.013	138	8.223	±	0.007

Feature	*Data=50%* *(#of data=1728)*		*Data=60%* *(#of data=2074)*		*Data=70%* *(#of data=2419)*		*Data=80%* *(#of data=2765)*	
	S	RMSE	S	RMSE	S	RMSE	S	RMSE
10%	1728	6.644 ± 0.137	2074	6.644 ± 0.137	2419	6.515 ± 0.006	2765	6.406 ± 0.005
20%	864	6.247 ± 0.137	1037	6.247 ± 0.137	1210	6.066 ± 0.018	1382	5.970 ± 0.002
30%	576	6.077 ± 0.048	691	6.077 ± 0.048	806	5.964 ± 0.276	922	5.740 ± 0.089
40%	432	6.461 ± 0.261	518	6.461 ± 0.261	605	6.299 ± 0.563	691	6.308 ± 0.677
50%	346	8.057 ± 0.037	415	8.057 ± 0.037	484	8.160 ± 0.018	553	8.110 ± 0.012
60%	288	8.104 ± 0.021	346	8.104 ± 0.021	403	8.147 ± 0.017	461	8.101 ± 0.002
70%	247	8.123 ± 0.009	296	8.123 ± 0.009	346	8.158 ± 0.011	395	8.107 ± 0.007
80%	216	8.177 ± 0.008	259	8.177 ± 0.008	302	8.194 ± 0.008	346	8.138 ± 0.005
90%	192	8.200 ± 0.009	230	8.200 ± 0.009	269	8.217 ± 0.004	307	8.157 ± 0.006
100%	173	8.239 ± 0.006	207	8.239 ± 0.006	242	8.235 ± 0.002	276	8.223 ± 0.001

The visualization of the results in the form of a series of heat maps see Figure 4-4 to 4-6, helps us arrive at a number of qualitative observations as well as to look at some quantitative relationships. In most cases, the performance index remains relatively low in some regions of the heat map. This finding demonstrates that the available data come with some evident redundancy, which exhibits a negative impact on the designed model. For the PM10 data, there is a significantly reduced performance of the model when for a low percentage of data, the number of features starts growing. This effect is present for different numbers of clusters. The same tendency is noticeable for the other data sets. There is a sound explanation to this phenomenon: simply, the structure formed by fuzzy clustering does not fully reflect the dependencies in the data (due to the effect of the sparsity of the data), and this problem, in turn results in the deteriorating performance of the fuzzy model. In this case, one would be better off to consider a suitable reduced set of features. In all cases experimented with, we noted an optimal combination of features and data that led to the best performance of the model. Table 4-7 summarizes the optimal combinations of features and data.

Table 4-7: The optimal % of features and data for different clusters

Data set and number of clusters	*% of features*	*% of data*
Pima with c=3	70	40
Pima with c=4	100	20
Pima with c=5	100	20
Pima with c=6	100	40
Pima with c=7	100	80
Housing with c=3	80	70

Housing with c=4	80	50
Housing with c=5	80	100
Housing with c=6	80	80
Housing with c=7	90	100
Body Fat with c=3	30	30
Body Fat with c=4	100	70
Body Fat with c=5	90	70
Body Fat with c=6	90	90
Parkinson with c=3	30	30

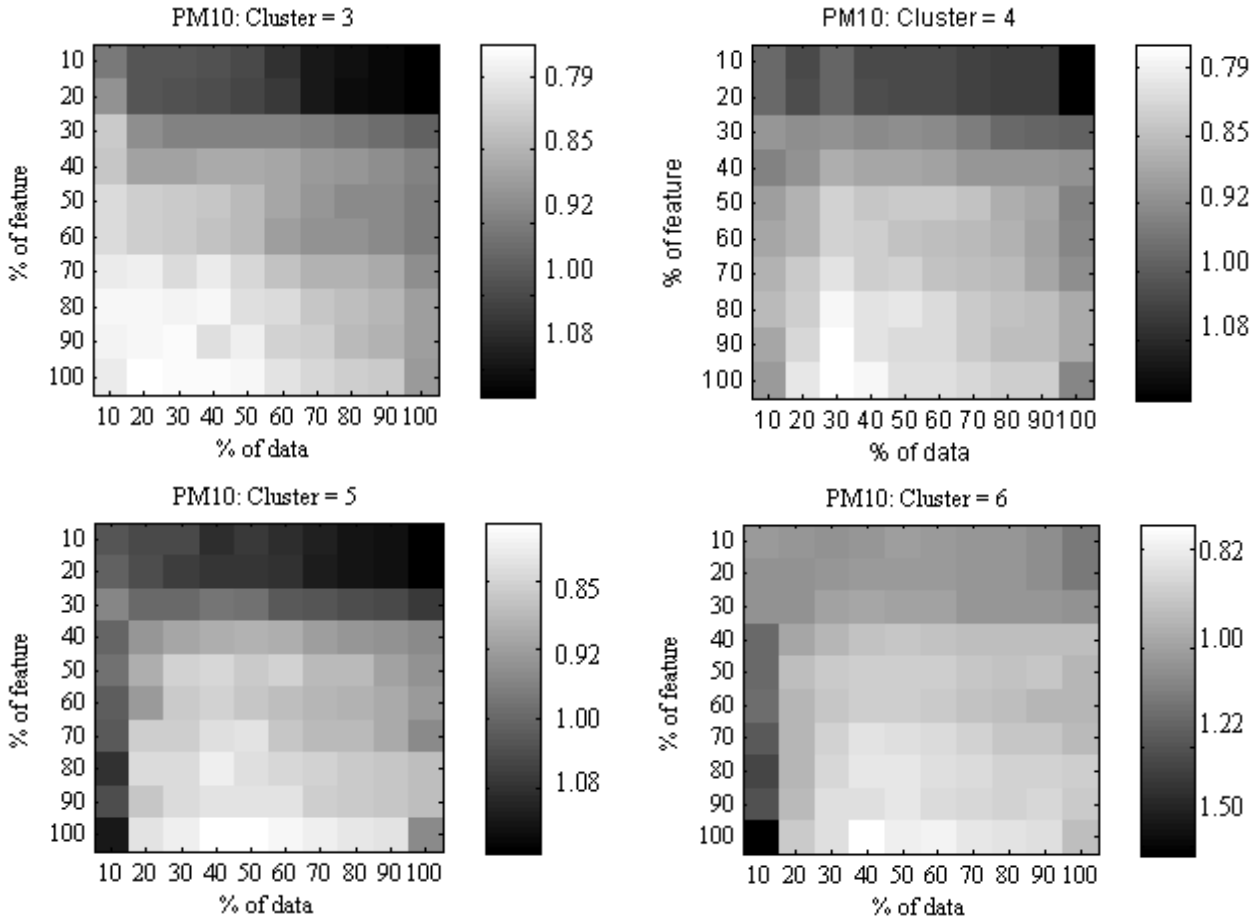


Figure 4-4: Heat map for PM 10 data for c varying in-between 3 to 6

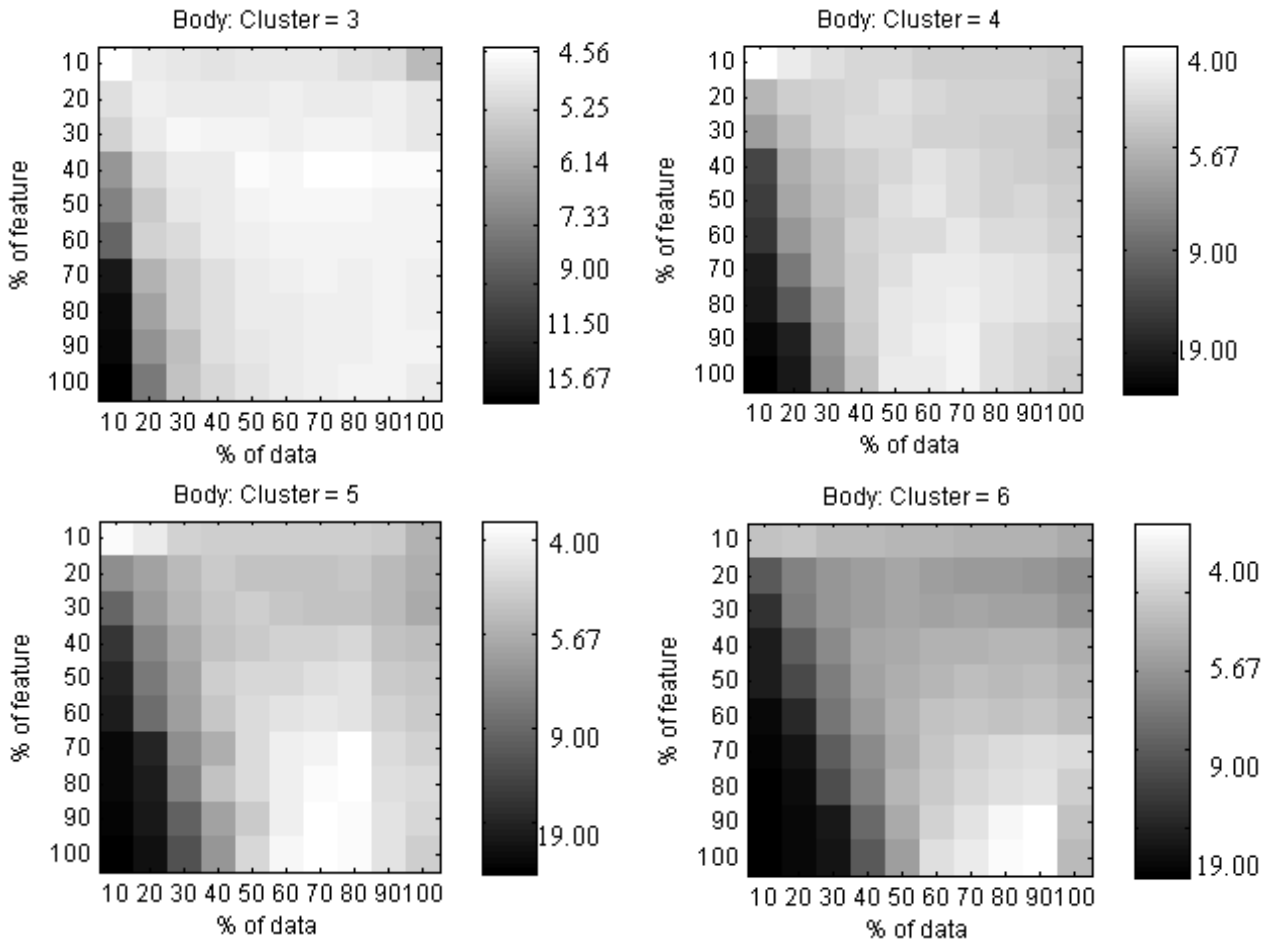
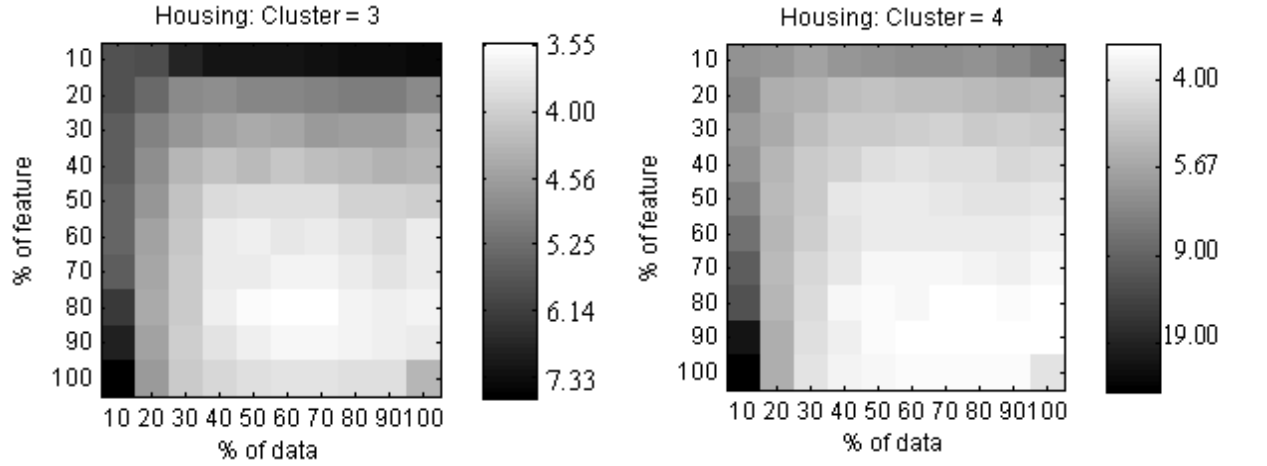


Figure 4-5: Heat map for Body fat data for c=3 to 6



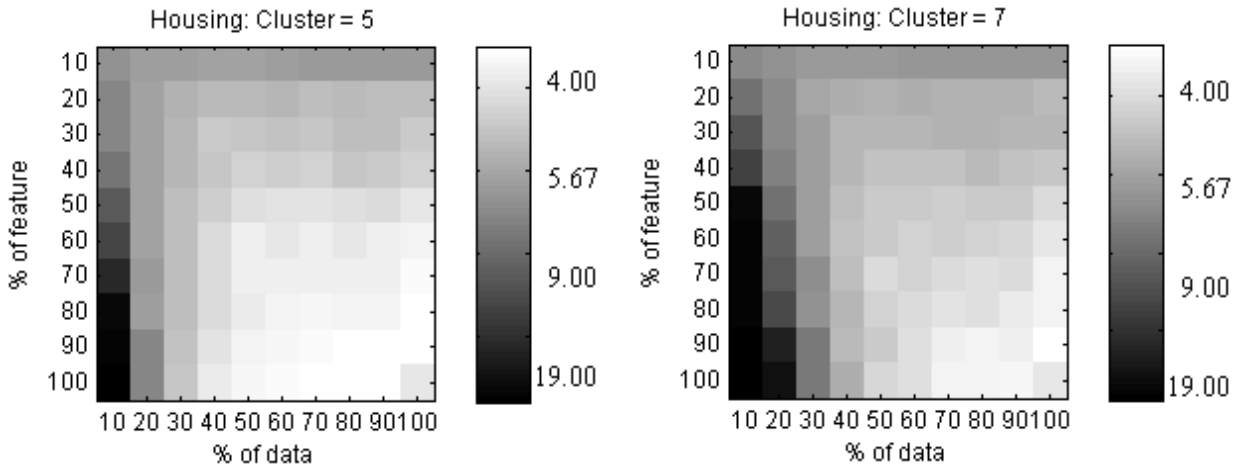
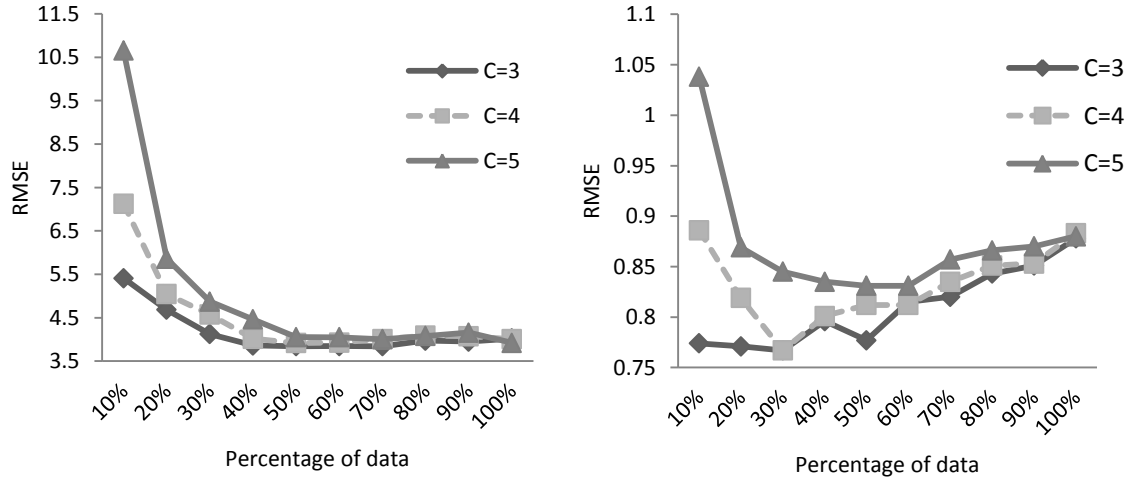
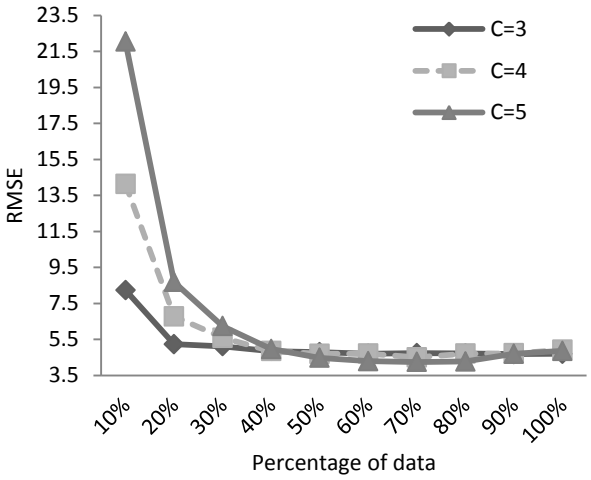


Figure 4-6: Heat map for Housing data for c=3, 4, 5, and 7

The relationships between the percentage of data used and the resulting RMSE values are displayed in Figures 4-7 and 4-8. Some interesting tendencies are worth noting. A critical number of data are required to form a fuzzy model. Increasing the number of data does not produce any improvement as the curves plotted on Figures 4-7(a), 4-7(b) and 4-7(c) achieve a plateau or even some increase of the RMSE is noticeable.

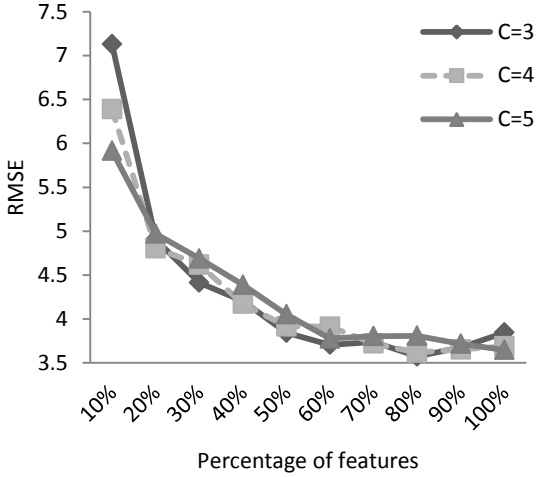


(a) (b)

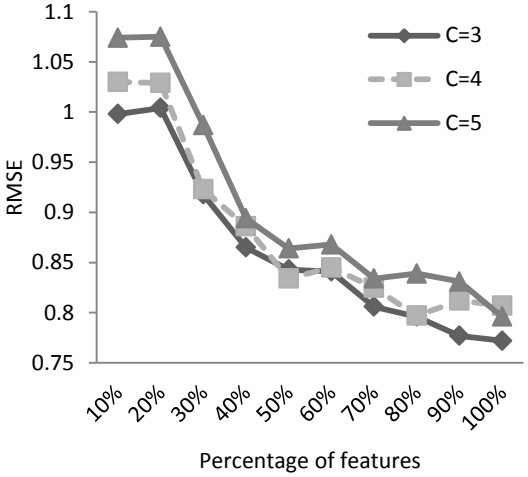


(c)

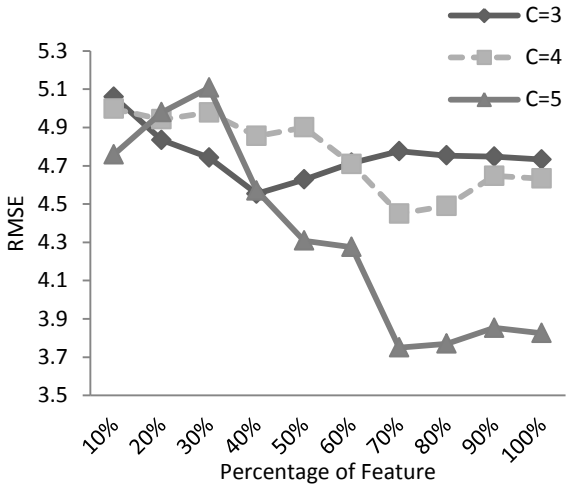
Figure 4-7: The values of RMSE versus the percentage of data for selected number clusters: (a) Housing data, (b) PM10 data, and (c) Body Fat data



(a)



(b)



(c)

Figure 4-8: Plots of RMSE versus the percentage of features for selected number clusters: (a) Housing data, (b) PM10 data, and (c) Body Fat data

Considering a fixed percentage of the data used, we look at the nature of the feature sets. Overall, the selected subsets of features are almost the same for different numbers of the clusters being used (see Table 4-8 to 4-10). Furthermore, we observe that in most cases, the reduced feature spaces exhibit an interesting "nesting" property, meaning that the extended feature space constructed subsumes the one formed previously. For example, for the Housing data, we obtain the following subsets of features:

{feature 6}$\subset$ {feature 6, feature 9, feature 13}
$\subset$ { feature 6,feature 9,feature 10,feature 13}

Here, the corresponding features are as follows: 6. Average number of rooms per dwelling, 9. Index of accessibility to radial highways, 13. Percentage of lower status population, and 10. Full-value property-tax rate per $ 10,000. This combination is quite convincing.
For the PM10 data, we arrive at a series of nested collections of features:

{feature 1}$\subset$ {feature 1, feature 7}
$\subset$ {feature 1, feature 6, feature 7}
$\subset$ {feature 1, feature 2, feature 6, feature 7}

where the corresponding features include: 1. The concentration of PM10 (particles), 7. Hour of experiment per day, 6. Wind direction, and 2. The number of cars per hour.

Table 4-8: Best subsets of features for Housing data

F	D=10%	D=30%	D=50%	D=70%	D=90%
10%	12	6	6	12	12
20%	5,12,13	6,9,13	6,9,10	6,9,10	6,9,10
30%	6,7,9,13	5,6,9,11	4,6,9,13	1,6,9,10	2,3,12,13
40%	1,3,6,10,13	4,6,9,10,13	6,9,10,12,13	3,5,6,9,10	1,4,6,9,10
50%	3,6,7,8,9,10,11	1,2,6,9,10,11,13	1,3,5,6,8,9,11	1,5,6,9,10,11,12	1,6,9,10,11,12,13
60%	3,5,6,7,8,9,12,13	3,5,6,7,8,9,10,11	1,3,5,6,7,9,10,11	1,3,5,6,7,9,10,11	1,3,6,7,8,9,11,13
70%	3,5,6,7,8,9,10,12,13	1,3,4,5,6,9,10,11,13	1,3,5,6,7,9,10,11,13	1,3,5,6,7,9,10,11,13	1,3,6,7,8,9,10,11,13
80%	1,3,4,5,6,8,9,10,11, 13	1,3,5,6,8,9,10,11,12 ,13	1,3,5,6,7,8,9,10,11, 13	1,3,5,6,7,8,9,10,11, 13	1,3,5,6,7,8,9,10,11, 13
90%	1,2,3,4,5,6,7,8,10,1 1,12,13	1,3,4,5,6,7,8,9,10,1 1,12,13	1,3,4,5,6,7,8,9,10,1 1,12,13	1,3,4,5,6,7,8,9,10,1 1,12,13	1,3,4,5,6,7,8,9,10,1 1,12,13

Table 4-9: Best subsets of features for PM10 data

F	D=30%	D=40%	D=50%	D=60%	D=70%	D=80%	D=90%
10%	1	1	1	1	1	1	1
20%	1	1	1	1	1	1	1
30%	1,7	1,7	1,7	1,7	1,6	1,6	1,6
40%	1,6,7	1,6,7	1,6,7	1,6,7	1,6,7	1,6,7	1,6,7
50%	1,2,6,7	1,2,6,7	1,2,6,7	1,2,6,7	1,2,6,7	1,2,6,7	1,2,6,7
60%	1,2,6,7	1,2,6,7	1,2,6,7	1,2,6,7	1,2,6,7	1,2,6,7	1,2,6,7
70%	1,2,3,6,7	1,2,3,6,7	1,2,3,6,7	1,2,3,6,7	1,2,3,6,7	1,2,4,6,7	1,2,4,6,7
80%	1,2,3,4,6,7	1,2,3,5,6,7	1,2,3,4,6,7	1,2,3,4,6,7	1,2,3,4,6,7	1,2,3,4,6,7	1,2,3,4,6,7
90%	1,2,3,4,6,7	1,2,3,4,6,7	1,2,3,4,6,7	1,2,3,5,4,7	1,2,3,4,5,7	1,2,3,4,6,7	1,2,3,5,4,7

Table 4-10: Best subsets of features for Body fat data

F	D=50%	D=60%	D=70%	D=80%
10%	1	1	1	1
20%	1,6,7	1,3,7	1,3,7	1,3,7
30%	1,6,7,9	1,7,8,9	1,3,7,9	1,3,7,9
40%	1,6,7,8,9,12	1,3,4,7,8,9	1,3,6,7,8,9	1,7,8,9,11,12
50%	1,2,6,7,8,12,14	1,4,8,9,11,12,14	1,3,7,8,9,12,14	1,3,4,5,7,8,12
60%	1,2,6,7,8,11,12,14	1,3,4,7,8,9,11,14	1,3,5,7,9,11,12,14	1,3,5,7,8,11,12,14
70%	1,3,4,5,6,8,9,11,12,14	1,3,4,5,7,8,9,11,12,14	1,3,4,5,7,8,9,11,12,14	1,3,4,5,7,8,9,11,12,14
80%	1,3,4,5,6,7,8,9,10,12,14	1,3,4,5,6,7,8,9,10,12,14	1,3,4,5,6,7,8,9,11,12,14	1,3,4,5,6,7,8,9,11,12,14
90%	1,2,3,4,5,6,8,9,10,11,12,13,14	1,3,4,5,6,7,8,9,10,11,12,13,14	1,3,4,5,6,7,8,9,10,11,12,13,14	1,23,4,5,6,7,8,9,10,11,12,14

Turning to the comparative analysis of performance of the swarm optimization methods, we summarize the obtained results in Figure 4-9. For all data, the CPSO performed better than the standard PSO. Although both algorithms show the same tendency when the percentage of feature is 100 % however, the RMSE produced by the CPSO is lower than the one obtained when running the PSO. Furthermore, the CPSO algorithm is more stable than the standard PSO. In most cases, the standard deviations of error produced by the CPSO are smaller than the results obtained for the standard PSO (see Table 4-11).

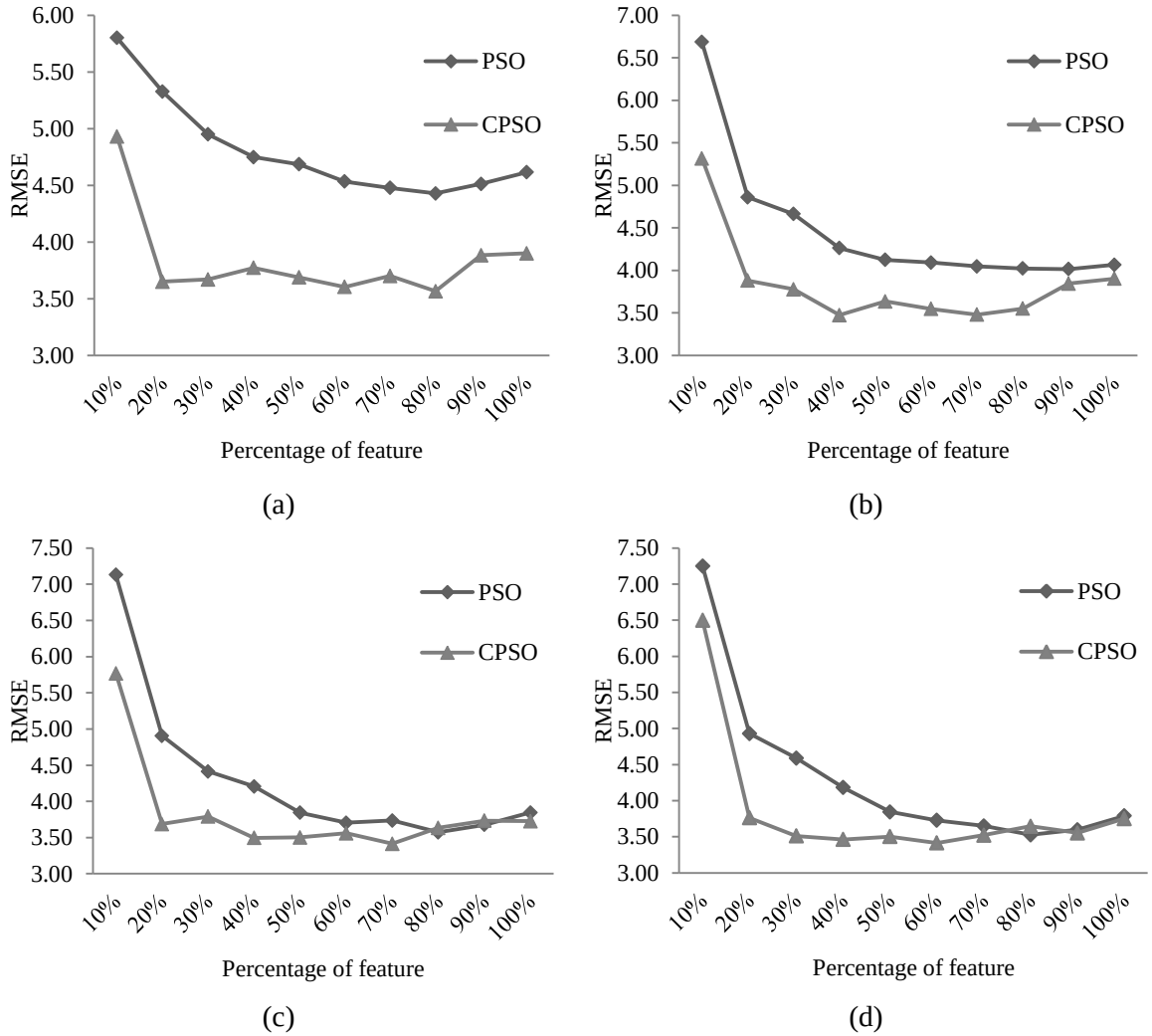


Figure 4-9: Values of RMSE versus the percentage of features selected when running PSO and CPSO – the use of the housing dataset: (a) 20 % of selected data, (b) 30 % of selected data, (c) 50 % of selected data, and (d) 70 % of selected data

Table 4-11: Standard deviations for PSO and CPSO (Housing and PM10 data sets)

Housing (D= 50%)		PM10 (D=50%)	
PSO	CPSO	PSO	CPSO
0.066	0.072	0.021	**0.007**
0.192	**0.015**	0.018	**0.009**
0.199	**0.039**	0.010	**0.007**
0.11	**0.093**	0.004	**0.007**
0.115	**0.079**	0.024	**0.008**
0.091	**0.071**	0.006	**0.009**
0.058	0.094	0.014	**0.010**
0.044	0.064	0.010	**0.007**
0.053	**0.042**	0.019	**0.009**
0.021	0.042	0.016	**0.009**

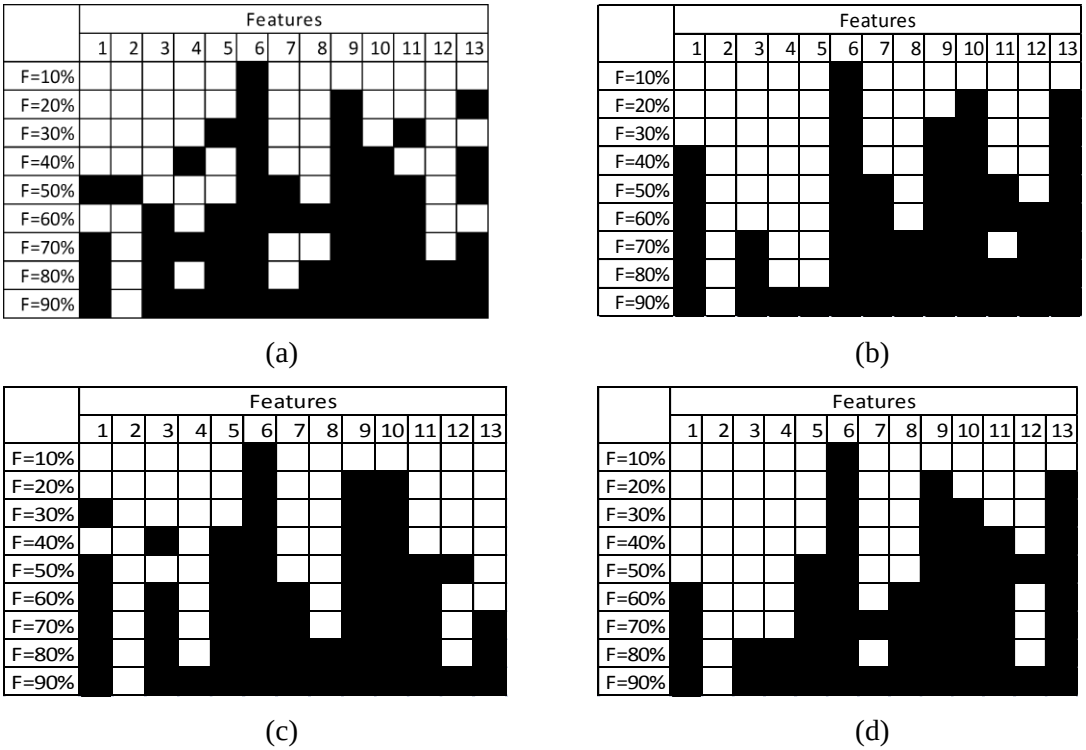


Figure 4-10: Comparison of sets of features being selected by using PSO and CPSO^2 for Housing data dataset: (a) PSO method with 30 % of selected data, (b) CPSO^2 method with 30 % of selected data, (c) PSO method with 70 % of selected data, and (d) CPSO^2 method with 70 % of selected data

Figure 4-10 shows the subsets of the features selected for different percentages of the features used in construction of the fuzzy model. The CPSO algorithm is more consistent while selecting the increasing number of features. For example, features 6 and 13 were selected when using both 30% and 70 % of data. In contrast to the selection made with the PSO algorithm, the subset of the features selected here is not as stable, especially when using only 30% of data.

Table 4-12: percentage of improvement of the RMSE obtained when using CPSO over the results formed by the PSO; Housing data set

F	D=10%	D=20%	D=30%	D=40%	D=50%	D=60%	D=70%	D=80%	D=90%
10%	13	15	20	26	19	18	10	19	11
20%	31	31	20	24	25	16	24	23	17
30%	33	26	19	20	14	15	24	18	21
40%	28	21	19	14	17	17	17	18	19
50%	34	21	12	7	9	9	9	12	8
60%	23	21	13	7	4	7	8	9	7
70%	19	17	14	4	9	6	4	8	8
80%	33	19	12	4	2	3	3	8	9
90%	22	14	4	5	2	1	2	2	1
100%	17	4	4	7	3	1	1	1	1

Table 4-12 presents the percentage of the improvement when using the CPSO algorithm compared to the PSO algorithm. Note that in this percentage we included all different combinations of the features' percentages and the data percentages being used. The percentage of the improvement is higher when dealing with a smaller percentage of features and data. For example, the percentage of improvement is 34% for 10% of the instances and 50% of the features selected while the percentage of improvement is less than 10% for 60% of instances and features used. These results occurred because the PSO method has to deal with a large search space for selecting a small subset of features and instances. In contrast to the search space for CPSO, the large search space is decomposed into multiple sub-swarms that reduce the dimensionality of the original search space.

Table 4-13 to 4-16 show the comparison of RMSE when using the proposed method and the standard fuzzy modeling method. Here the standard fuzzy model is constructed without using any feature and instances selection and the holdout method is used to select the data based on the percentage given. The experiment for using the standard fuzzy modeling is repeated for 25 times. If we analyze the tables, we can observe that our proposed method outperforms the standard method of constructing the fuzzy model from the dataset. This can be seen clearly when using the CPSO method to search for the best subset of feature and instances. For example, in Table 4-13 if we used the CPSO method the RMSE for using 70% of data is 3.413, whereas the RMSE for the standard method is 8.312. The same tendency occurs for all datasets used here.

Table 4-13: The comparison of RMSE obtained when using standard PSO, CPSO, and standard fuzzy model with holdout method for Housing data with C=3

% of data	Standard PSO		Cooperative PSO[1]		Holdout Method	
	% of	MSE	% of feature	MSE	%of	MSE
30	90	4.015	40	3.473	100	17.593
40	80	3.699	70	3.464	100	10.803
50	80	3.573	70	3.414	100	9.907
60	80	3.556	70	3.435	100	8.507
70	80	3.527	60	3.413	100	8.312
80	80	3.654	90	3.449	100	8.164
90	80	3.679	90	3.615	100	7.641

Table 4-14: The comparison of RMSE obtained when using standard PSO, CPSO, and standard fuzzy model with holdout method for Body fat data with C=3

%of data	Standard PSO		Cooperative PSO[1]		Holdout Method	
	% of	MSE	% of feature	MSE	%of	MSE
30	30	4.677	30	4.6847	100	11.586
40	30	4.717	30	4.5409	100	8.291
50	40	4.617	30	4.5136	100	7.548
60	50	4.636	40	4.4289	100	7.073
70	40	4.548	40	4.4234	100	6.658
80	40	4.553	40	4.4233	100	6.239
90	40	4.582	40	4.3771	100	6.102

Table 4-15: The comparison of RMSE obtained when using standard PSO, CPSO, and standard fuzzy model with holdout method for PM10 data with C=3

%of data	Standard PSO		Cooperative PSO[1]		Holdout Method	
	% of	MSE	% of feature	MSE	%of	MSE
30	100	0.764	80	0.7338	100	2.100
40	100	0.765	80	0.7432	100	2.018
50	90	0.777	90	0.7790	100	2.030
60	80	0.805	80	0.7769	100	1.986
70	90	0.820	80	0.8052	100	2.001
80	80	0.839	90	0.8206	100	1.983
90	70	0.847	90	0.8417	100	1.976

Table 4-16: The comparison of RMSE obtained when using CPSO and standard fuzzy model with holdout method for Computer data with C=3

%of data	Cooperative PSO		Holdout Method	
	% of	MSE	%of feature	MSE
30	40	16.446	100	17.453
40	30	14.712	100	17.524
50	30	14.350	100	17.680
60	40	14.935	100	17.837
70	40	16.237	100	17.918
80	40	15.122	100	18.351

Figure 4-11 shows the comparison plot between the proposed method and the standard fuzzy modeling. In most of the cases, the proposed method better performance.

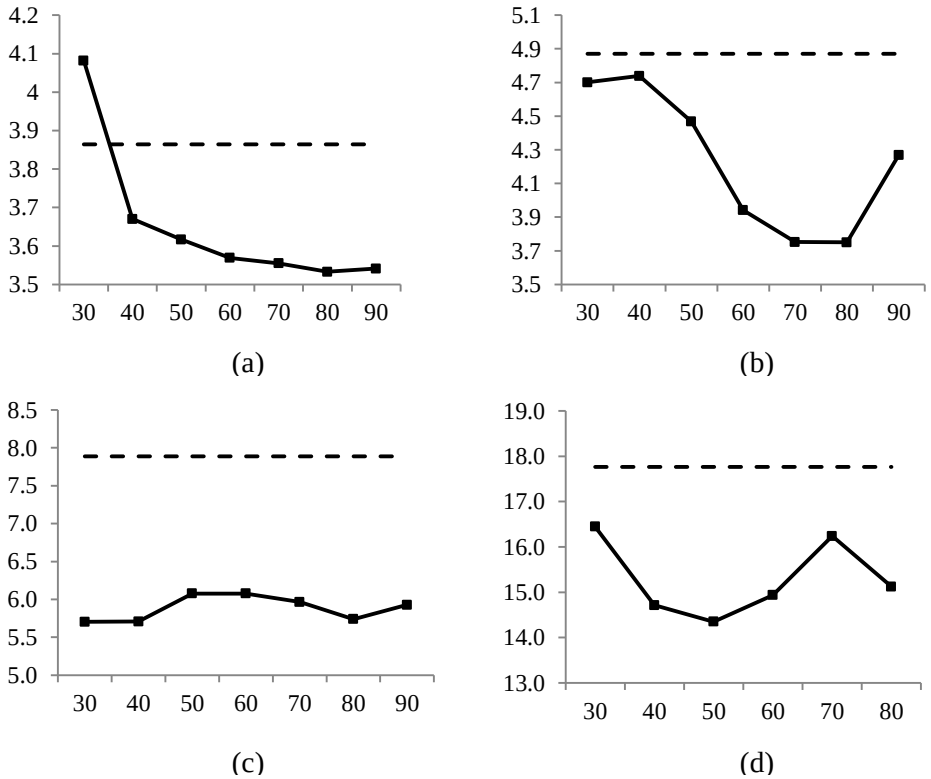


Figure 4-11: Comparison of RMSE by using proposed method and standard fuzzy model (dotted line): (a) Housing dataset with c=4, (b) Body fat dataset with c=5, (c) Parkinson dataset with c=3, and (d) Computer dataset with c=3

It becomes clear that one is able to reduce the input data in terms of the number features and instances. Moreover, the flexibility of choosing the reduction level helps the user focus on the most essential subsets of data and features (variables). The knowledge acquired about the best subset of data can be used for future data collection. In addition, the user can put more effort analyzing only the best subset of data that give more impact to the overall prediction.

4.5 Feature and Instances Selection for Nearest Neighbor Classification via Cooperative Binary PSO

The Nearest Neighbor (NN) Classifier is one of the well known non-parametric classification approach. It was proposed by Cover and Hart (Cover and Hart 1967) and is applied in many application domains such as pattern recognition, data mining, and machine learning. This method is popular because it is easily being implemented and it is conceptually straightforward comparing to other supervised learning method. It is a non-parametric classifier. Therefore, there is no model perform in the training phase. The effectiveness of the classification process is based on the instances contained in the training set. Thus, the classification rate is relied on the quality of the training set. Moreover, applying this approach to a real world problem suffers from several problems such as they are computationally expensive classifiers since that the whole training set must be stored in the computer for classify the unseen data. In addition, NN classifier is in-tolerant with the irrelevant features. Facing vast amount of data in the real world applications, the use of some reduction mechanism becomes a necessity.

In the literature, most of the works on improving the NN are by selecting the subset of instances. Selecting proper instances in the training dataset can improve the classification rate and computational complexity of the classifier. Moreover in some cases, the training dataset may contain noisy or redundant instances. Therefore by using the instances selection, the training data are the subset of the most useful set of instances. For example, in (Arturo, J and Kittler 2010), the authors proposed an instances selection algorithm based on nearest neighbor rule called the Condensed Nearest Neighbor Rule (CNN). This method focused on finding a subset such that every member of the original dataset was closer to a member of the subset of the same class than to a member of the subset of different class. An improvement of this method was proposed in (Ritter, et al. 1975). In this method, called the Selective Nearest Neighbor (SNN), each member in the original dataset must be nearer to a member of the dataset of the same class than any member of the original dataset of a different class. Another popular method applying the same

approach is proposed in (Wilson and Martines 2000) and is called DROP (Decremental Reduction Optimization).

In this research, we proposed an alternative method to improve the performance of the NN by simultaneously select the best subset of feature and instances for the training data. The integration of feature and data reduction process is guided by using Cooperative Binary PSO. The cooperative version of binary PSO is simple but efficient method for searching the best subset of feature and instances simultaneously. Here, we divide the candidate solution (particles) into several sub-components called sub-swarms. The first component is dealing with feature selection and the rest are dealing with the instances selection. The cooperative behaviors among the sub-swarms improve the selection process for both feature and instances. The selection of the feature is directly from the first sub-swarm of the candidate solution. On the other hand, for the selection of instances we divide the search space by classification label given by the original dataset. Therefore, each classification group is guided by one sub-swarm. Finally, all sub-swarms work cooperatively with each other to come out with the best solution.

Generally, the performance of the classifier is based on the classification rate. Therefore, our performance index is based on the classification rate for NN. The other aspect, our framework focused to overcome the limitation of the NN when dealing with the large dataset. In this regard, Cooperative Binary PSO is applied in order to find the best subset of data to be used for training the NN classifier. This framework was implemented and tested on 14 datasets from the Machine Learning Database Repository and StaLIB.

4.5.1 The Proposed Methodology

In this research, we employed Cooperative Binary PSO (CBPSO) method for searching for the best subset of feature and instances. The search space for the CBPSO method is based on the classification label given in the dataset. For example in Iris dataset, we have three classification groups; therefore, total number of sub-swarms for searching the best subset is four. Here, we have four different search spaces that we solve individually by using CBPSO method. The motivation behind the use of cooperative version of PSO, as advocated in (van den Bergh and Engelbrecht 2004) is to deal effectively with the dimensionality of the search space, which becomes a serious concern when a large dimensionality of the feature space and instance are involved. The essence of the cooperative version of PSO is essentially a parallel search for optimal subset of feature and instances. The cooperative strategy is achieved by dividing the candidate solution vector into components, called sub-swarm, where each sub-swarm represents a small part of the overall optimization processes. By doing this, we implement the concept of

divide and conquer to solve the optimization problem, so that the process will become more efficient and fast.

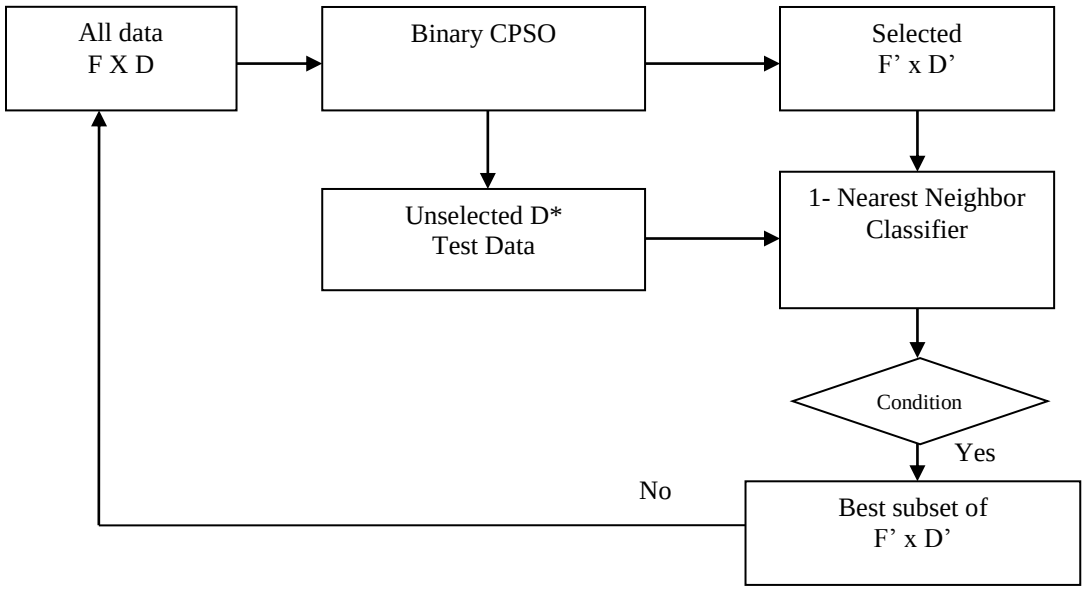


Figure 4-12: General framework of integration FS and IS

Figure 4-12 illustrates the framework of the feature and instances selection by using the Binary Cooperative PSO (FISCBPSO). The framework can be divided into three main parts and can be described below:

1: Reduction process by CBPSO: The Cooperative BPSO is the main tool for searching the best instances for solving the classification problem. By using the cooperative method, we can reduce the size of the search space for the PSO method. Here, each class will be put in one sub-swarm. The total number of sub-swarms is based on the total number of classification label. For example, for iris dataset the number of sub-swarm is equal to 3. Therefore, by dividing the search space into their own classification group, we can reduce the complexity of the algorithm and improve the computational time especially for the large dataset.

2: 1-Nearest Neighbor Classifier is a simple and effective method comparing to other learning method. The selected data is then used as the training data for 1-NN classifier, and the unselected data are the test data.

3: The generating of the new subset of instances will be stop when the algorithm met the stop condition. Finally, the subset of instances with the highest accuracy will be selected as the best subset of instances.

The FISCBPSO uses the fitness function that focused on the main objective that is to improve the classification rate. There are two criteria to be considered in the optimization process. The first one is concerned about the selection of the features, and the second one is the selection of the instances. The classification rate can be computed as the following:

$$ClassRate(F,I) = \frac{\#\ Instances\ classified\ correctly}{\mathrm{M}}, \quad (4\text{-}3)$$

where F is the sub-swarm for feature selection, I, are the sub-swarms of the instances selection and M is the number of instances in the training set.

4.5.1.1 The description of the particles and its representation

Next, we discuss the representation of the particle in each of the sub-swarm. The length of the particle in the first sub-swarm is equal to the total number of features in the dataset. Then for the rest sub-swarms the length is based on the total number of sample in the classification label. The value in each element in the particle is a binary number, which are the values of 0 and 1. In order to make the particle representation related to feature and instances selection, a value of 1 means that it selected features or instances and vice versa. For example, given a list of features in the first sub-swarm F= {F1, F2, F3, F4, F5} and n = 5, a sub-swarm may look like:

X(1) = { 0, 0, 1, 1, 1 },
X(2) = { 1, 0, 1, 0, 1 },
X(3) = { 0, 0, 1, 0, 1 }

4.5.2 Experimental Studies

In this section, we elaborate on a set of experiments, in which we used several classification data sets from machine learning repository (see http://www.ics.uci.edu/~mlearn/MLRepository.html and http://lib.stat.cmu.edu/ datasets/). The main objective of these experiments is to show the abilities of the proposed method and quantify the performance of the selected features and instances. A brief summary of the data sets used in the experiment is presented in Table 4-17 and 4-18.

Table 4-17: Data description

Data set	*Number of features*	*Number of data*	*Number of Classes*
Bupa	7	645	2
Ionosphere	34	351	2
Mammography	6	961	2
Pima	7	768	2
Wisconsin (Wis)	32	699	2
Image	19	210	7

Iris	4	150	3
Glass	9	214	7
Zoo	17	101	7
Ecoli	8	336	8
Yeast	8	1484	10

Table 4-18: High dimensional datasets

Data set	*Number of features*	*Number of data*	*Number of Classes*
Movement_Libras	90	360	15
Spambase	57	4597	2
Satimage	36	6435	7
Sonar	60	208	2

The values of the BCPSO parameters were set using the standard form as follows. The value of the inertia weight w was set equal to 0.7. The values of the cognitive factor, c_1 and social factor c_2 were set to 1.49 and 1.49, respectively (Jiang, Luo and Yang 2007). The number of sub-swarms used in the BCPSO is based on the number of classes for dataset used in the experiments. Here, we divided the search space into several sub-swarms that can cooperate with each other and where the individuals in the sub-swarms are used to represent a portion of the search space.

In this section, we experimentally evaluate the proposed framework. Table 4-19 shows the result achieved by using the instances selection method to the dataset in Table 4-17. The best results in accuracy are highlighted in bold. The proposed instances selection method outperforms the Adaptive Search Algorithm (CHC) and Generational Genetic Algorithm (GGA) (Derrac, Garcia and Herrera 2009) especially when dealing with the dataset with the large number of class.

Table 4-19: ISCBPSO vs. IS algorithm (Instances selection only)

Dataset	*ISCBPSO*	*1-NN*	*CHC*	*GGA*
Bupa	**0.7400**	0.6122	0.6963	0.6825
Mammo	0.8416	0.7377	0.8429	**0.8558**
Pima	0.7519	0.7033	**0.7595**	0.8313
Sonar	0.9293	0.8632	**0.9467**	0.6789
Wis	**0.9644**	0.9569	0.8539	0.8948
Iris	**1.0000**	0.9548	0.9272	0.8797
Glass	**1.0000**	0.7077	0.7560	0.6050
Zoo	**1.0000**	0.9280	0.9806	0.6630

Table 4-20: ISCBPSO method vs. IS algorithm (accuracy)

Dataset	*ISCBPSO*	*1-NN*	*ARB1*	*ARB2*
Ecoli	**0.9281**	0.8085	0.8571	0.863
Yeast	**0.5925**	0.5088	0.5223	0.5108

In Table 4-20 displays the result of using Yeast and Ecoli dataset. We purposely chosen this data because of the nature of these dataset classification problems is such that the given features are not sufficient to distinguish between classes. The yeast dataset appears to be a difficult classification problem. It contained 1484 instances representing ten classes. The same problem appears with the Ecoli dataset. However, the number of instances is small. For both dataset, our method gives the best accuracy result compared to the other two methods. We compared the result with ARB1 and ARB2 methods (Marwah and Boggess 2002). The proposed method easily identify the best instances for the training set that can provide a better accuracy result using the unseen data. Figure 4-13 displays the data selected distribution based on their classification label for Glass data set, Yeast data set and Ecoli data set.

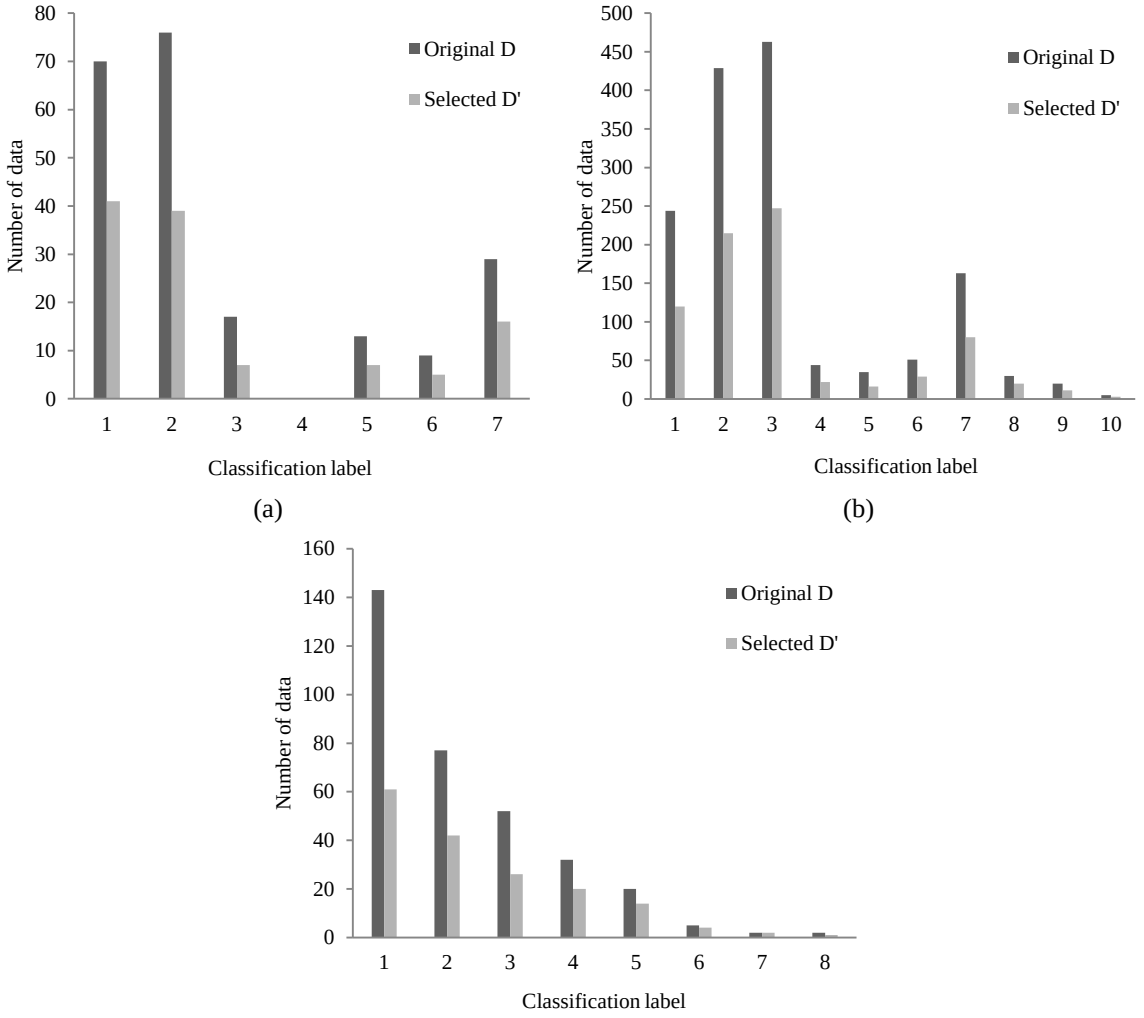


Figure 4-13: The distribution of the selected data vs. the original data ; (a) Glass dataset, (b)Yeast dataset, and (c) Ecoli dataset.

Table 4-21: FISCBPSO vs. ISCBPSO

Dataset	*IS & FS*	*IS*
Bupa	**0.7418**	0.7400
Mammo	**0.8759**	0.8416
Pima	**0.7615**	0.7519
Wisconsin	**0.9650**	0.9644
Ionos	**0.9572**	0.9322
Image	**0.9533**	0.9462

Table 4-22: Reduction ratio achieved

Dataset	*Feature*	*Instances*
Bupa	0.43	0.51
Mammo	0.50	0.26
Pima	0.43	0.54
Wisconsin	0.50	0.50
image	0.58	0.49

In Table 4-21 shows the comparison results between two data reduction techniques. The integration of feature and instances selection outperforms the accuracy of using only instances selection. Equally interesting are the resulting of reduction ratio of feature and instances for each of the dataset used in the experiments. The results are shown in Table 4-22. Overall, the ratio for feature and instances are around 0.5. That means that we only used around 50 % of the original data as the training dataset.

In the second experiment, we used 4 high dimensional datasets display in Table 4-18. The accuracy results obtained over these datasets shows in Table 4-23. Here, we compare our proposed method with Adaptive Search Algorithm (CHC) and Interactive Genetic Algorithm (IGA) for selection the feature and instances. In all cases, our method outperformed the other two methods. Figure 4-14 displays the data selected distribution based on their classification label. In most cases, our method only selects 50% of the original data.

Table 4-23: FISCBPSO vs. FIS algorithm (accuracy) using large dataset

Dataset	*IFS-BCPSO*	*CHC*	*IGA*	*1-NN*
spam	**0.9166**	0.9071	0.9112	0.8945
libras	**0.8971**	0.6583	0.7234	0.8194
sati	**0.9126**	0.8611	0.8383	0.9058
Sonar	**0.9340**	0.7561	0.7878	0.8555

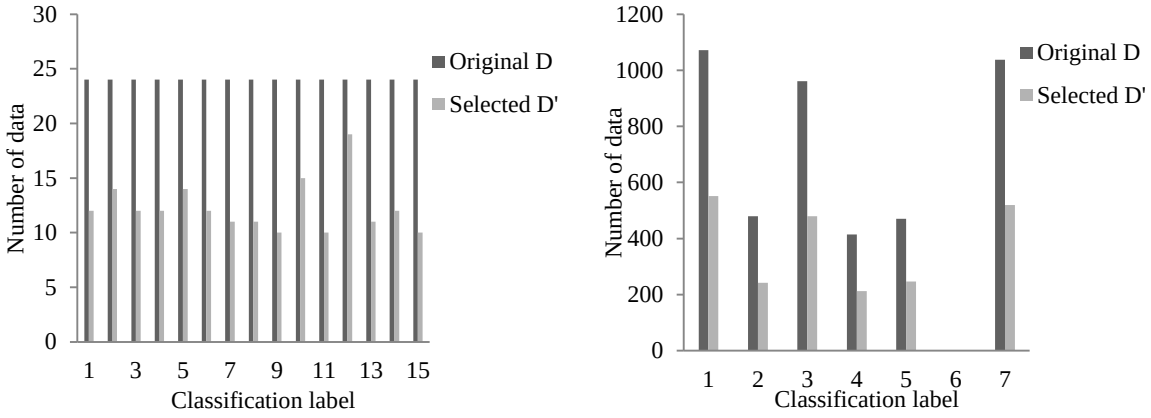


Figure 4-14: The distribution of the selected data vs. the original data; (a) Libras dataset and (b) Sati mage dataset

4.6 Conclusions

In this chapter, we proposed a simple framework for constructing fuzzy modeling from high dimensional and large data. This framework has several advantages that make it better suited than other frameworks for sharing various real-life problems. Firstly, the simultaneously feature and instances selection is easily adapted to construct the structure of the fuzzy model. Secondly, the best selected subset of data obtained with this framework is capable of representing the original large data set. Thirdly, we construct an optimal (or sub-optimal) collection of features and data based on the PSO. In addition, a cooperative PSO is developed in order to overcome the limitation of using standard PSO when dealing with a high dimensional search space. The size of the selected features and data used to construct the fuzzy model can be adjusted based upon the feedback provided in terms of the performance of the model constructed for the currently accepted.

The effectiveness of the framework was validated by using four well-known regression data sets. The experiment results showed that the proposed fuzzy modeling framework is able to handle high dimensionality and a large data set simultaneously. Moreover, the curse of dimensionality problem in fuzzy modeling was substantially reduced. In the future work one could concentrate on improving the Cooperative PSO by fine tuning the parameters of the method such as e.g., the cognitive and social parameter.

Next, we have proposed an alternative approach for feature and data reduction problem based on the cooperative BPSO technique for classification problems. Here, the particle for CBPSO is divided into several components, called sub-swarms. The first sub-swarm is for selecting the

feature and the rest are for selecting the instances. The employment of a cooperative approach allows our method to integrate the process of feature reduction and data reduction simultaneously. Moreover, this component by component optimization allows fine tuning of each component by each particle. The results achieved by our proposed method in the experimental study performed have shown that it offers better data selection than using other existing method. In addition the method that we proposed is efficient and comprehensible.

In the next chapter, we discuss an alternative method for improving the complexity of the fuzzy system by reducing the number of rules. Here, we focus on designing the compact fuzzy rules based on the original fuzzy rule-based system. The granular fuzzy rule is come from the main concept in information granularity that emphasis on the generality of the granular representation.

5. The development of granular rule-based systems: A study in structural compression

In this chapter, we develop a comprehensive design process of granular fuzzy rule-based systems. These constructs arise as a result of a structural compression of fuzzy rule-based systems in which a subset of originally existing rules is retained. This chapter is organized as follows. In Section 5.1, we explain about the complexity reduction in fuzzy rules-based system. Next, in Section 5.2 we discuss the underlying concept. In the sequel, we discuss an optimization criterion quantifying the performance of the granular rules. A suite of protocols of allocation of information granularity is presented. In Section 5.4, we describe the PSO environment using which the granular fuzzy rule-based system is constructed. In Section 5.5, experimental studies are given. Finally, conclusions and some prospects of further research are presented in Section 5.6.

5.1 Complexity reduction in fuzzy rules-based system

There have been several efforts from the fuzzy rule based systems community to strike a balance between reducing the model complexity and increasing the model performance. For example in (Setnes, Babuska, Kaymak, & van Nauta Lemke, 1998) a merging strategy was suggested to eliminate redundant fuzzy rules by using a measure of similarity. Genetic Algorithm also been used for eliminating the redundant rule and for identifying the important rules (Krone, Krause and Slawinski 2000, Roubos and Setnes 2001, Ishibuchi and Yamamoto 2004). In some cases the method of Singular Value Decomposition is used to reduce the number of rules (Baranyi and Yam 2000, Tanaka, Taniguchi and Wang 2000). In addition, even though it is conceivable that removal of redundant or less important rules from the original fuzzy rule based systems can result in a compact fuzzy system, but the generality ability of the reduced rule is not an easy to achieve.

The rules are viewed as descriptors of individual, local pieces of knowledge, especially when forming a global mapping from the space of conditions to the space of conclusions. When dealing with a large number of rules, emerges an interesting and practically viable question about a reduction of the number of rules so that a small subset of the most representative rules can be formed. The reduction process is important because of two main reasons. First, the smaller number of rules enhances their readability meaning that the transparency of the reduced model becomes enhanced. Second, computing overhead is reduced. Starting from the set of rules "if **x** is A_j then y is B_j" j=1, 2,…, N, the reduction of the model leads to the subset of rules "if **x** is A_i then

y is B_i" i=1, 2, …, I where I<<N. Surprisingly, the reduced rules do not reflect a fact they are the subset of the original far larger collection of rules. Intuitively, we might have anticipated that the reduced rule set reflects the reduction aspect by having a level of abstraction of the fuzzy sets standing in the condition parts of the rules being elevated. In other words, the reduced set of rules comprises the conditional statements of the form "if x is $\boldsymbol{G}(A_i)$ then y is B_i " The increased level of abstraction (generality) is realized by forming a granular augmentation of the original fuzzy set A_j by generalizing it to the granular fuzzy set $\boldsymbol{G}(A_j)$ viz. an interval fuzzy sets, fuzzy set of type-2, shadowed fuzzy sets, probabilistic (fuzzy) sets (Bargiela and Pedrycz 2003) and other generalizations. In a nutshell, the term granular fuzzy set stands for the generalization of the fuzzy set in which the original numeric value of membership, say $A_j(x)$ at point "x" is generalized to the granular value (interval, fuzzy set in [0,1], probability density function, etc.). This granular nature of the proposed construct is directly associated with the reduced number of rules to compensate for the reduction of the rule base, the rules have to be made more abstract (viz. granular). Figure 5-1 illustrates granular fuzzy rules in general. In the figure we can visualize the process of the rule reduction by selecting subset of rules and the process rule generalization by the constructing of the granular rules.

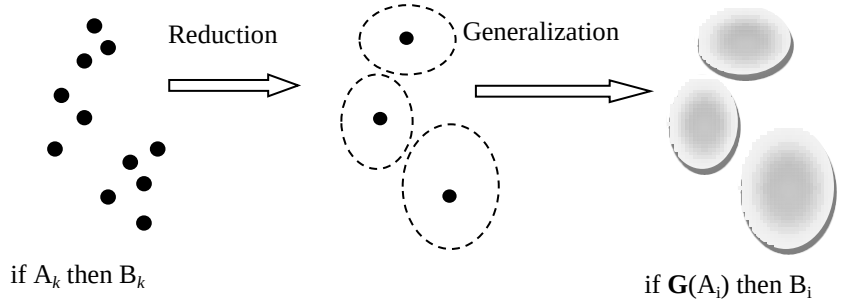


Figure 5-1: Reduction of rule base by selection and a granular extension (generalization) of the representative subset of rules. The granular constructs are shown as shadowed disks.

Assuming that the reduced set of rules has been formed viz. the collection of rules "if **x** is $\boldsymbol{G}(A_j)$ then y is B_j" has been decided upon, the fundamental question arises as to the formation of the granular fuzzy sets. The underlying design principle is that of an optimal allocation of information granularity. The values of the membership grades are non-numeric, say they become intervals or membership functions. Given a certain predetermined level of information granularity α we allocate it among the elements of the original fuzzy set (by making it *granular*) so that a balance of information granularity is met and a certain optimization criterion is maximized. The optimization criterion used to guide the process of granularity allocation expresses an extent to

which the results of inference process realized with the use of all the rules are "covered" by the results formed by the reduced rule-based system.

The development of the granular rule based system comprises two important and intertwined phases, namely a selection of a subset of the rules and a formation of the optimal allocation of information granularity. Given the combinatorial character of the first phase and a nonlinear nature of the overall process of granularity allocation, in the study we consider a particle swarm optimization environment (PSO) as well as its generalized cooperative version.

5.2 From fuzzy rule-based models to granular fuzzy rule-based models: the concept

The essence of fuzzy rule-based systems is inherently associated with the inference schemes of approximate reasoning

x is A (5-1)
if x is A_i then y is B_i, i=1,2,.., N
---
y is B

where B is a fuzzy set of conclusion to be determined. A and A_i are defined in a finite input space **X**, dim (**X**) = n while B_i and B are expressed in the output space **Y** of dimensionality "m". The set of indexes of the rules is denoted by **N**; in this case it is simply a set of N natural numbers indexing the rules, **N** ={1, 2, …, N}.

There is a wealth of realizations of the inference schemes with a large number of optimization mechanisms (Alcala, Ducange, et al. 2009, Pedrycz, Knowledge-based Clustering: From Data to Information Granules 2005). In a nutshell, though, the inference scheme is realized by determining the activation levels of the individual rules (their condition parts) implied by some A. This is typically done by computing a possibility measure of A and A_i, poss(A, A_i). Denoting the possibility value by λ_i, the conclusion B is taken as a union of B_i weighted by the activation levels.

Now let us envision that instead of the entire collection of rules, we consider a subset of **I** rules in anticipation that this smaller collection can be deemed sufficient as being formed by a collection of the most representative rules out of N rules. Of course, the term *representativeness* has to be clarified and quantified as well as made operational. What is also quite intuitive is a fact that the rules forming the subset need to be made more abstract to compensate for the fact that

they need to capture the entire set. Operationally, by making them more abstract (general) means that we form the condition parts of the selected rules more general. This, in effect, implies that instead of A_j occurring in the selected rule, we consider a certain granular abstraction of A_j, say $\boldsymbol{G}(A_j)$ where $\boldsymbol{G}(.)$ stands for the granular version of A_j. All in all, this generalization gives rise to the granular fuzzy rules

$$\text{-if } \boldsymbol{G}(A_j) \text{ then y is } B_j \tag{5-2}$$

j=1, 2, …, I. Now **I** is a collection of indexes coming from **N** identifying the subset of rules, that is $I = \{j_1, j_2, ..., j_I\}$.

The ensuing inference scheme comes in the form

$$\begin{aligned} &x \text{ is } A \\ &\text{If x is } \boldsymbol{G}(A_j) \text{ then y is } B_j \\ &\text{----------} \\ &\text{y is } \boldsymbol{G}(B) \end{aligned} \tag{5-3}$$

It is worth noting that the granular format of the condition of the rule entails a granular format of the conclusion so we obtain the granular counterpart $\boldsymbol{G}(B)$ instead of the fuzzy set B.

The granular version of A_j , $\boldsymbol{G}(A_j)$ can be articulated in different ways. In a nutshell the granularity of A_j results in non-numeric membership values. Several main alternatives are outlined in Table 5-1.

Table 5-1: Selected formal models of granular versions of fuzzy set A- membership grade A(x) for fixed element of the universe of discourse

Interval granulation	$\boldsymbol{G}(A(x)) = [a_1(x), a_2(x)]$
Fuzzy set-based granulation	$\boldsymbol{G}(A(x)) = F_{A(x)}(u), \quad u \in [0,1]$ where F is a fuzzy set defined in the unit interval
Probability-based granulation	$\boldsymbol{G}(A(x)) = p_{A(x)}(u), \quad u \in [0,1]$ where p is a probability density function defined in the unit interval, $\int_0^1 p_{A(x)}(u)du = 1$

In the ensuing study, for the clarity of the presentation of the underlying concept, our focus is on interval (set-based) granulation. Thus we consider the interval-valued fuzzy sets, $\boldsymbol{G}(A_j)$, see also Table 5-1. In this case in the general inference scheme (5-3) the activation of $\boldsymbol{G}(A_j)$ results in an interval of activation values $[\lambda_j^-, \lambda_j^+]$. As a result, the conclusion becomes an interval fuzzy set $[B_j^-, B_j^+]$ with the bounds computed as

$$[B_j^-(y), B_j^+(y)] = [\max_{j=1,2,...I}(\lambda_j^- \wedge B_j(y)), \max_{j=1,2,...I}(\lambda_j^+ \wedge B_j(y))] \tag{5-4}$$

The development of the granular rule-based system entails two tightly connected design phases:

(a) selection of the subset of rules **I** out of the entire collection of rules

(b) generalization of the condition parts – fuzzy sets A_j are made granular

These two steps are intertwined and have to be discussed together. The first one is evidently of structural (combinatorial) character. The second one is about making the original fuzzy sets of condition granular. In what follows, let us formulate an optimization criterion used to guide an overall development process.

5.2.1 Inclusion measure as an optimization criterion

Let us assume that the set of rules **I** have been already formed (we discuss this development in the successive sections). The quality of these rules can be evaluated as follows: we consider the remaining N-I rules not present in the collection of rules being retained. We treat successive A_js present there as the inputs to the inference process (5-3). The result becomes an information granule, $\boldsymbol{G}(B_j)$. Intuitively, the quality of the granular rule-based system depends on how well the information granule $\boldsymbol{G}(B_j)$ "covers" original B_j considering that the granular rules form only a subset of the original rule base. The fundamental with this regard is the notion of coverage and its quantification. We introduce the following coverage index (measure)

$$\kappa = \frac{\sum_{y}\sum_{j} \text{incl}[B_j(y), \boldsymbol{G}(B_j(y))]}{(N - I)m} \tag{5-5}$$

where incl $(B_j(y), \boldsymbol{G}(B_j)(y))$ is a measure of inclusion of $B_j(y)$ in the granular counterpart produced by the inference scheme (5-3). The first summation standing in this formula is done over all elements of the finite output space over B_j and $\boldsymbol{G}(B_j)$ are defined whereas the second sum is carried out for all rules left out from the process of the generation of granular rules (whose number is N-I). The inclusion measure can be fully specified depending upon the assumed formalism used in the construction of granular rules. In the simplest case, where dealing with interval-valued membership functions, the double sum in the nominator of (5-5) is a count specifying how many times the membership value $B_j(y)$ is contained in the interval $[B_j^-(y), B_j^+(y)]$.

5.3 The design of optimal granular fuzzy rules

In this subsection, we discuss the process of constructing the granular rules. As mentioned in the previous section, the granular rules are realized in the form of interval-like values. The important

parameters that have the direct affect to the quality of the granular rules are the level of granularity (α) and its allocation in each of the constructed granular fuzzy rules. Here, the optimization method is the best approach to find the optimal allocation of information granularity. The quality of granular fuzzy rules can be assessed as the objective function for the optimization method. Let us assume a certain level of information granularity, α with values in the unit interval. This values associates with the given membership function grade $A_i(x)$ by forming an interval of length α distributed around $A_i(x)$ with eventual clipping of the range (if required). Figure 5-2 shows the different between the original fuzzy rule-based system and the granular fuzzy rules-based system. The membership function for the original fuzzy rules-based system depicted in Figure 5-2(a). Then the granular fuzzy rules is achieved by shifting the points on the Gaussian function to the left and to the right based on the level of granularity, as in Figure 5-2(b).

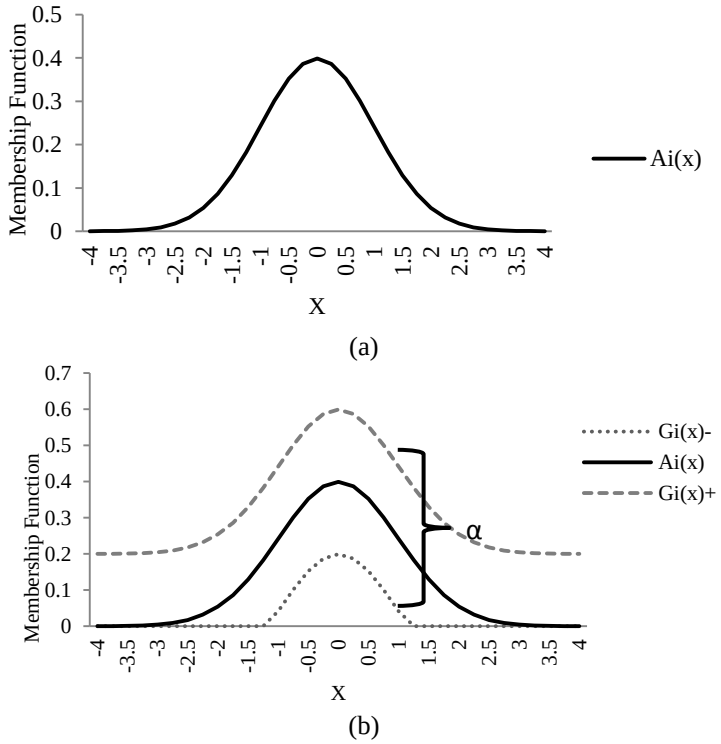


Figure 5-2: (a) Example of a FRBS, $A_i(x)$ and (b) a Granular Fuzzy Rules

In what follows, we start with a detailed discussion on the evaluation of the quality of the subset of the rules. We show that the quality of the granular fuzzy rules can be optimized by a suitable allocation of available information granularity (α). By considering that the reduced rule base comprises I rules and the fuzzy sets of condition are defined in the n-dimensional input space, we consider the quantity α × card(I) × n as an asset of information granulation to be distributed throughout the fuzzy sets of condition. More specifically, we allocate a certain level of

granularity (α) to each element of the input universe of the discourse and to each fuzzy set of condition of the rule standing in the reduced collection of rules.

5.3.1 Protocols of allocation of information granularity

As mention earlier, the granular rules constructing by shifting the points on the membership function of the selected original rules. This process is called the allocation of information granulation and it can be realized in several different ways depending how much diversity one would like to consider in the allocation process. We discuss the performance of each of the protocol in the context of rules if x is A_i then y is B_i. Recall that the dimensionality of the input space is "n" while the output space has "m" elements. In what follows, we discuss several protocols of allocation of information granularity:

Protocol 1(P_1*)*: A uniform allocation of information granularity for all membership degrees for the selected rules. The membership grades are replaced by later the length α. More specifically if *a* is the value $a \in [0, 1]$ then the corresponding interval membership values is $[a - \alpha/2, a + \alpha/2]$. Of course the overall balance of information granularity is satisfy that is $n \cdot \alpha$. No optimization is required.

Protocol 2(P_2*)*: A uniform allocation of information granularity with asymmetric position of interval. It is similar to P_1 however it exhibit flexibility as we allow the asymmetric allocation information granules (intervals) meaning that the membership values is $[a - \gamma \cdot \alpha, a + (1-\gamma) \cdot \alpha]$ where $\gamma \in [0, 1]$. The optimization concerned adjustment of the value of asymmetric (γ).

Protocol 3(P_3*)*: It comes as an augmentation of P_2. We admit asymmetric allocation of information granularity to individual membership grades. The membership grades a_i, i=1, 2, ..., n are generalized and assume the form of the interval $[a_i - \gamma_i\alpha, a_i + (1-\gamma_i)\alpha]$ where $\gamma_i \in [0, 1]$. In total, we have a vector of coefficients $[\gamma_1, \gamma_2, \ldots, \gamma_n]$.

Protocol 4(P_4*)*: A non-uniform allocation of information granularity with symmetrically distributed intervals of information granules. Here, the protocol involves individual intervals distributed symmetrically around a_i formed as follows,

$$[a_i - \alpha_i/2, a_i + \alpha_i/2] \tag{5-6}$$

The balance of information granularity is retained meaning that

$$\sum_{i=1}^{n} \alpha_i = n \cdot \alpha \tag{5-7}$$

Protocol 5(P_5*)*: A non-uniform allocation of information granularity with asymmetrically distributed intervals of information granules. Here, it generalizes P_3 in the sense that the constructed intervals are distributed asymmetrically. Thus α_i is replaced by the interval

$$[\, a_i - \alpha_i^-,\ a_i + \alpha_i^+] \tag{5-8}$$

With the balance of information granularity expressed as

$$\sum_{i=1}^{n} \alpha_i^{-} + \sum_{i=1}^{n} \alpha_i^{+} = n \cdot \alpha \tag{5-9}$$

In summary the search space explored by each of the protocols can be described as follows

Protocol	*Parameters*	*Dimensionality of search space*
P_1	α	no optimization
P_2	γ,α	optimization of γ, $\gamma \in [0, 1]$, (1)
P_3	α , γ_i i=1,2,...n	optimization of γ_1 , γ_2 ,... γ_n , (n)
P_4	α_i i=1,2,...n	optimization of α_1 , α_2 ,... α_n , (n)
P_5	α_i^- , α_i^+ i=1,2,...n	optimization of α_1^-, α_2^-,... α_n^- and α_1^+, α_2^+,... α_n^+ , (2n)

When dealing with two-input (or multivariable) rule-based systems the same protocols apply, however the condition on the retention of information granularity involves the condition $(n_1 + n_2)$ α where n_1 and n_2 are the dimensionality of the corresponding input spaces say "if x is A_i and Z is C_i then y is B_i". Here A_i is defined over a discrete space dimensionality n_1 and C_i is expressed over a space of dimensionality n_2.

The quality of the allocation protocol can be quantified by the coverage measure introduced in the previous section. Note also that κ is a non-decreasing function of α. The value of asymmetric (γ) is given in the form of increment value from 0 to 1. The increment size is given by the user and the best value will be selected for constructing the granular fuzzy rules.

Equally interesting, we illustrate the resulting granular realization of each protocol in Figure 5-3. Here we use Gaussian function to show the original rule, called A_i. Then the granular rules for the upper interval and the lower interval are G+ and G-, respectively. In Figure 5-3(a) we implement P_1 and from the graph we can see that the interval is uniform for all dimensions in the granular rules. Next in Figure 5-3(b) we show the representation of P_2 where the asymmetric (γ) is not balances. Here we used γ=0.2 and we can see that the interval is not uniform for all dimensions. The upper interval is wider than the lower interval. Then if we change the value of asymmetric into γ=0.75 as a result the graph is opposite representation from the first graph. In

Figure 5-3(c) the interval is not uniform because we implement P_4 for constructing the granular rules. Here we used different value of allocation of the information granularity (ε) for each dimension of the rule. Some of the interval is quite narrow because of the small values of the ε use to construct the granular rule. Finally, in Figure 5-3(d) display the representation of using P_5. Here we used different value of allocation of the information granularity (ε) and the asymmetric value is not balance between upper and lower interval of the granular fuzzy rules.

In this study, we are using the Particle Swarm Optimization techniques to search for the best subset of fuzzy rules and simultaneously find the optimal allocation of information granulation that maximum the performance index (coverage). The next subsection explains the procedure of PSO techniques as the main optimization tools in constructing the granular fuzzy rules.

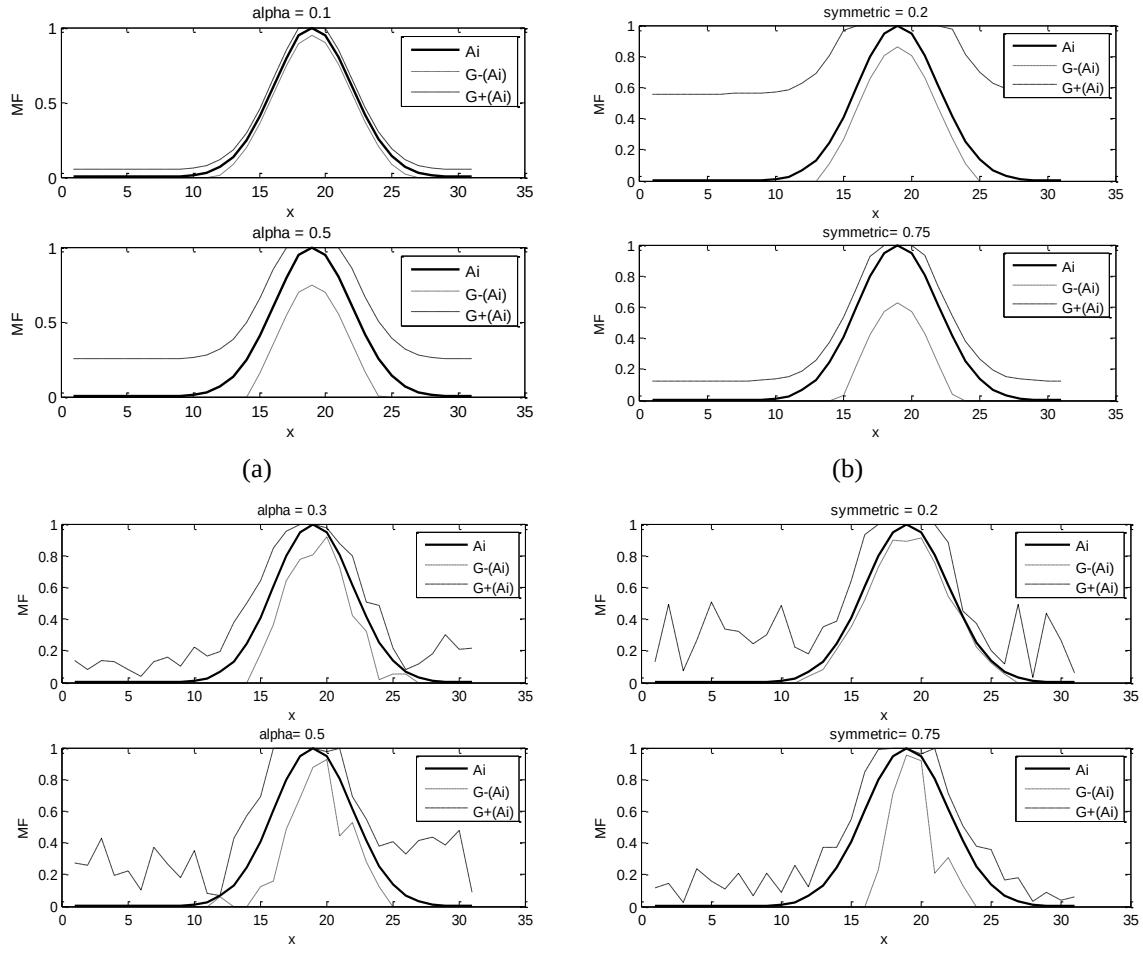


Figure 5-3: The protocols' visualization; (a) P_1, (b) P_2, (c) P_4, and (d) P_5

5.4 Particle swarm optimization as a design environment

The optimization process for granular fuzzy rules is based on the setting of a certain information allocation protocol. For protocol P_1 and protocol P_2, we need to optimize a single optimization task namely selection of the optimal rules, $\mathbf{I} = \{j_1, j_2, \ldots, j_I\}$. Whereas, for protocol P_3, protocol P_4, and protocol P_5 we add another optimization tasks namely, the optimal allocation of information granulation $[\alpha_1 ... \alpha_n]$. These two optimization tasks can be handled by the corresponding nested optimization process. In other words, for a subset of the rules generated by the optimization process at the upper level, one carries out the optimal allocation of information granularity by following a certain format of the assumed protocol. In this study, we implement the Particle Swarm Optimization for solving the optimization method when we use P_1 and P_2 of the allocation of information granularity. Next, for P_3, P_4 and P_5 we implement Cooperative Particle Swarm Optimization method for solving both optimization tasks simultaneously.

5.4.1 Particle Swarm Optimization and its variants

In this research, we employed PSO method for solving the optimization problem for constructing the optimal granular fuzzy rule-based system. The search space for the PSO method is based on the protocol for the allocation of information granularity. In Protocol P_1, the search spaces contain only the representation for the rule selection. Next for Protocol P_2 we add another search space that represents the optimal value of asymmetric (γ) for each dimensionality of the input rules. Then, for P_3 we still have the first search space for selecting the rule and the second search space represent the optimal value of allocation of information granularity (α) for each dimension of the input rules. P_4 is the combination of the P_2 and P_3; therefore, we have three different search spaces that we concatenate together as one particle for PSO method. Finally, in P_5 we need to find the optimal value of allocation of information granulation for each dimension of the rules for both intervals.

In order to deals with the large search spaces in P_2, P_3, P_4, and P_5, we employed Cooperative PSO method to solve all optimizations problem simultaneously. The motivation behind the use of cooperative PSO, as advocated in (van den Bergh and Engelbrecht 2004) is to deal effectively with the dimensionality of the search space, which becomes a serious concern when a large number of rules with a large dimensionality are involved. This curse of dimensionality is a significant impediment negatively impacting the effectiveness of standard PSO. The essence of the cooperative version of PSO is essentially a parallel search for optimal subset of rules and its optimal allocation values. The cooperative strategy is achieved by dividing the candidate solution

vector into components, called sub-swarm, where each sub-swarm represents a small part of the overall optimization processes (El-Abd 2006). The cooperative search between one sub-swarm to another is achieved by sharing the information of the global best position (P_{GB}) across all sub-swarm. Here, the algorithm has the advantage of taking two steps forward because the candidate solution come from the best position for all sub-swarm except only for the current sub-swarm being evaluated. Therefore, the algorithm will not spend too much time optimizing the rules or allocation values that have little effect of the overall solution. The rate at which each swarm converges onto the solution is significantly faster than the rate of convergence of the standard PSO.

Next, we discuss the representation of the particle in each of the optimization task. In the first optimization, the particle represents the rules in the original fuzzy rules. The length of the particle is equal to the total number of rules in the original fuzzy rules-based system. The value in each element in the particle is a real number, which is in the interval 0 to 1. In order to make the particle representation related to rule selection, we ranked the numbers in the particle in increasing order. Here, the ranked values for each element in the particle express the rule allocation in the original fuzzy rules-based system. For example, if the total number of rules is 25, and we only want to select 4 rules for constructing the granular fuzzy rules, then the first 4 entries denote the selected rules. In the second optimization process we need to find the optimal value for the information granulation allocation $[\alpha_1...\alpha_n]$. The length of the particle is equal to the dimensionality of each rule. Each element in the particle is represented by a real number that follows the constraints given in Eqn. (5-7).

5.4.2 Fitness function

Granular fuzzy rule-based system uses an objective function focused on the main objective that is to maximum the coverage of the subset of rules over the unselected rules. There are two criteria to be considered in the optimization process. The first one is concerned about the selection of the rules used to develop the granular rules, while the second one is the optimal value of the allocation of information granulation. The optimization process realized by considering the quality of the granular fuzzy rules constructed. Here, the quality of the granular rules can be quantified by counting how often the conclusion of the rule not being a part of the reduced set of rules is covered (contained) by the conclusion resulting from the reduced set of rules being now composed by granular fuzzy rules. Let us consider $A=A_j$ where the rule {if A_j then B_j} is not a part of the index set I. The fuzzy set A is processed by the rule in (5-2) resulting in the interval-valued conclusion $[B^-, B^+]$. We count the elements of the conclusion space where $B_j(y) \in [B_j^-(y),$

$B_j^+(y)$]. The process of repeating for all N-I rules that are outside the reduced rule set and a total count (s) is obtained. The following is the formula for calculating the coverage value:

$$\kappa(\alpha) = s / ((N-I)*m) \tag{5-10}$$

In an ideal situation, the coverage value is equal to 1, which becomes indicative of a complete inclusion of the conclusion of the original rule in the granular result of reasoning completed for the reduced rule base. In more realistic, the ratio gets lower than 1.

In addition, we introduced another objective function to evaluate the granular fuzzy rules based system called the area under the curve, AUC. As discuss above the value of κ depends upon the predetermined level of α, emphasized by the notation κ (α). Here, a monotonicity property is satisfied, namely κ (α) is a non-decreasing function. Higher values of α imply higher values of coverage of the fuzzy sets of conclusion. To achieve an overall assessment of the quality of the granular fuzzy rules, we integrate the corresponding values of κ (α), which results in a single index independent from the assuming level of granularity:

$$\mathrm{AUC} = \int_0^1 \kappa(\alpha)\mathrm{d}\alpha \tag{5-11}$$

This integral will be referred to as an area under curve, AUC. Here the selected rules are all the same for different value of alpha used to construct the granular fuzzy rules. Assume that the size of the reduced rule base, card (**I**), has been provided. Given a certain protocol of allocation of information granularity P, determine such **I**, called $\mathbf{I}_{opt}$ so that the value of κ becomes maximum value.

$$\mathrm{Max}_{\mathbf{I}}\ \kappa \tag{5-12}$$

The expression (5-12) leads to a combinatorial optimization problem and we implement the Particle Swarm Optimization method for solving both criteria mention above.

5.5 Experimental Studies

The experimental studies are concerned with a number of rule-based systems that are reported in the literature; see (Turksen and Berg 1992). The summary of these systems is presented in Table 5-2.

Table 5-2: Description of fuzzy rule-based system used in the study

Fuzzy Rule-Based System	*Abbreviation*	*Number of input variables*	*Number of rules*
Synthetic	Synthetic	1	8
Service Center Operation	Service	2	27
Mortgage Loan Assessment	Applicant	3	12
Aircraft Landing Control	Aircraft	2	20
Image1	Image1	1	11
Image2	Image2	1	9

In the ensuing experiments, the values of the parameters of PSO and CPSO were set up as follows. The inertia weight, "w" changes linearly from 1 to 0 over the course of optimization. The values of the cognitive factor, c_1 and social factor c_2 were set to 1.49 and 1.49, respectively (Eberhart and Shi 2001). In Table 5-3, we list the remaining details of the PSO and CPSO environment. As to the size of the population and the number of generations, their values are higher for the generic version of the PSO than the CPSO because of the higher dimensionality of the search space this algorithm operates in. We used 50 particles, 500 generations for the standard PSO and 30 particles and 250 generations for the CPSO method. In the CPSO, the number of sub-swarms is chosen based on the number of input variables used in the original rules. Table 5-3 displays the number of sub-swarm used in the CPSO method applied to 4 different optimization problems. The number of sub-swarms depends on the number of rules and input variables. By a proper allocation of each element to different sub-swarm, we improve the performance of the CPSO to complete search for the best rules and the allocation of each rule. Moreover, the search for the best solution is shorter compared to the search done when using only a single swarm (as encountered in the generic PSO).

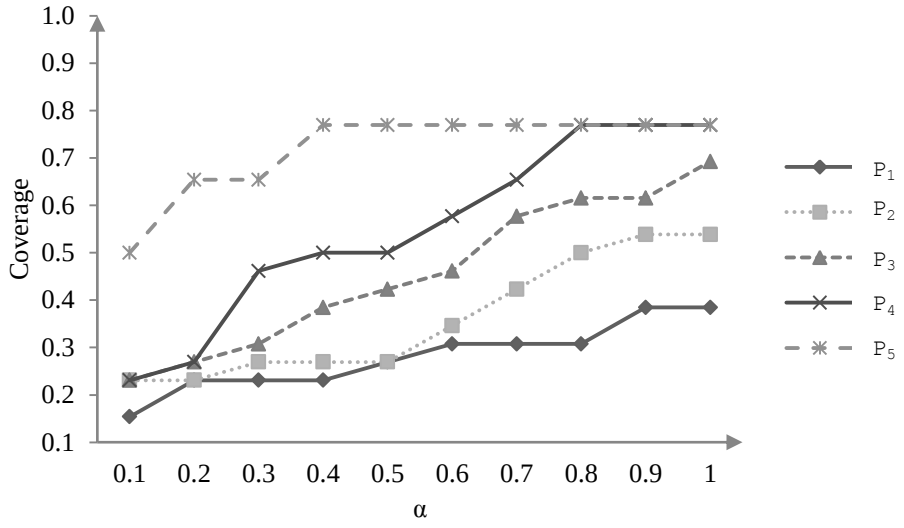
Table 5-3: The values of the parameters used in the experiments (RS= Rule Selection AT= Allocation tuning)

Optimization Method	*Sub -swarms*
Synthetic	2 :{ RS , AT}
Service	3 :{ RS , AT Input 1, AT Input 2}
Aircraft	3 :{ RS , AT Input 1, AT Input 2}
Application	4 :{ RS , AT Input 1, AT Input 2, AT Input 3}

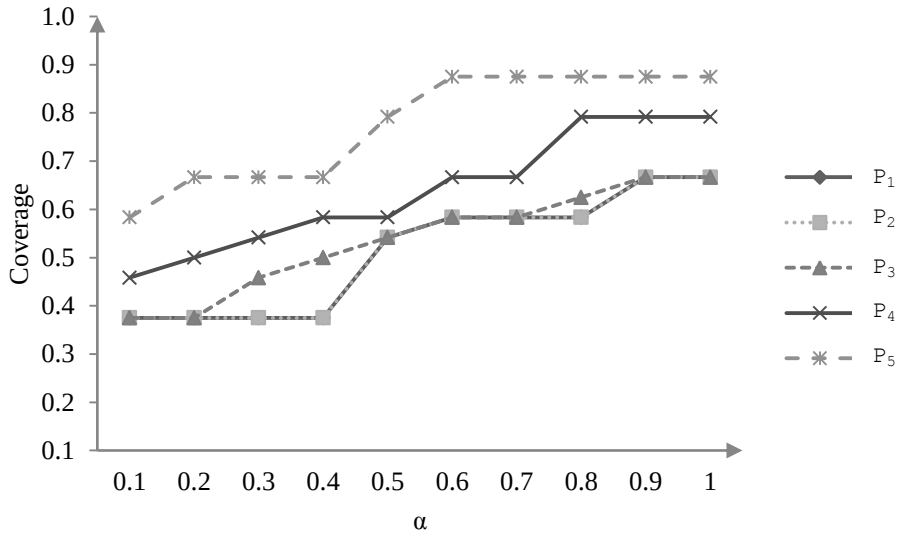
Synthetic fuzzy rule-based system- We consider the following collection of eight rules "if x is A_k then y is B_k" with fuzzy sets in the condition and conclusion part defined in the finite universes of discourse:

A_k	B_k
[0.1 0.9 0.5 0.2 0.1 0.0]	[0.0 0.3 0.5 0.8 1.0]
[0.7 1.0 0.6 0.3 0.2 0.0]	[1.0 0.7 0.3 0.2 0.0]
[0.9 0.9 1.0 0.2 0.0 0.0]	[0.1 0.9 0.9 0.4 0.2]
[0.0 0.3 0.5 0.9 1.0 0.0]	[0.0 0.4 0.9 1.0 0.5]
[1.0 0.9 0.5 0.2 0.1 0.0]	[0.0 0.3 0.5 0.8 1.0]
[0.6 0.3 0.2 1.0 0.5 0.7]	[0.5 0.9 1.0 0.5 0.2]
[0.2 0.3 1.0 0.2 0.5 0.7]	[0.0 0.3 0.5 0.8 1.0]
[0.0 1.0 0.5 0.3 0.0 0.0]	[0.3 1.0 0.2 0.0 0.0]

To illustrate the performance of the method, we start with a reduced set of two rules, I =2, that is N'= {7, 8}. These two rules were selected in an arbitrary fashion. The results are reported in Figure 5-4(a). There is a significant improvement when using protocol P_5 when compared the obtained results to the results produced by the remaining protocols. This is not surprising as this protocol offers a significant level of flexibility when allocating information granularity. The improvement is particularly visible for low values of α. Figure 5-4(b) shows the result using the optimal subset of two rules. Again there is a visible improvement in comparison with the results presented in Figure 5-4(a).



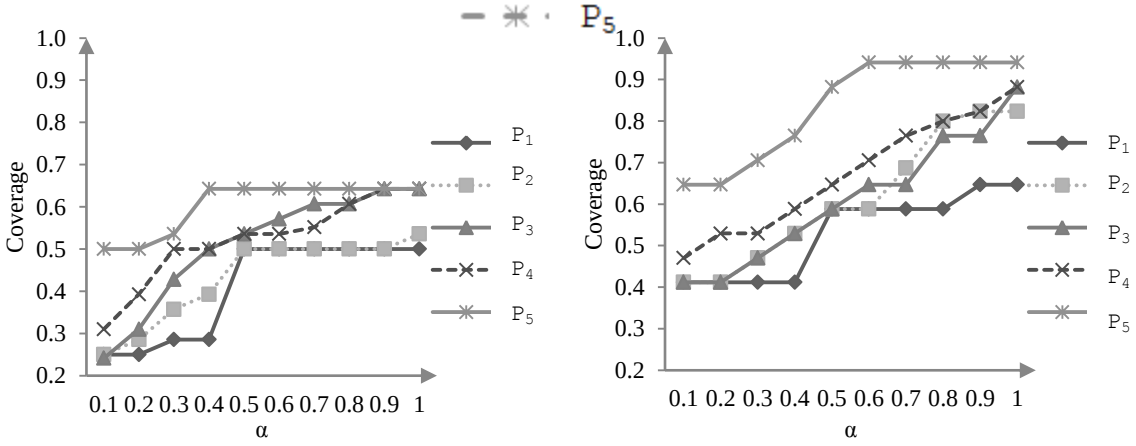
(a)



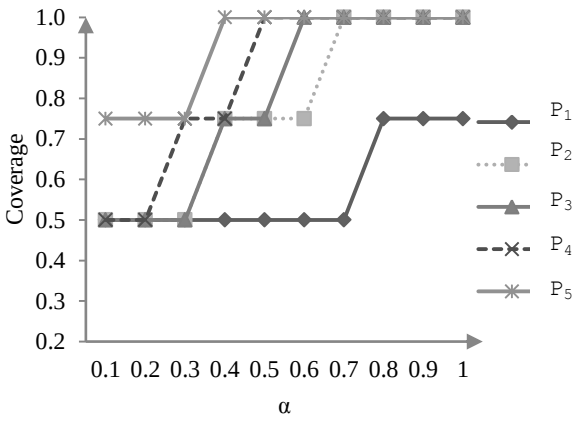
(b)

Figure 5-4: The coverage produced by the five protocols, (a) two arbitrarily selected rules, and (b) optimized two rules

Figure 5-5 illustrates the coverage values when using the PSO-optimized subsets of rules with N'=1, 4, and 7. The quality of results (ranging from the weakest coverage to the highest one) brings a ranking of the protocols ordered as P_1, P_2, P_3, P_4, and P_5 with P_1 producing the lowest coverage.



(a) (b)



(c)

Figure 5-5: The coverage produced by the five protocols, (a) N'=1, (b) N'=4, and (c) N'=7

In the sequel, Figure 5-6 shows a distribution of the allocation of granularity realized with the use of the protocol P_5; apparently, the distribution becomes non-uniform over the input space.

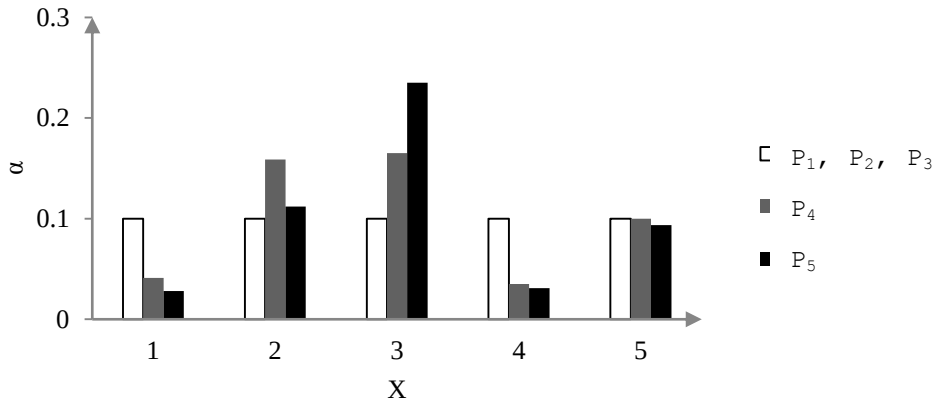


Figure 5-6: The allocation of information granularity for α=0.1

Figure 5-7 illustrates the values of coverage when using different number of rules. The coverage values are higher when increasing the number of selected rules. As illustrated in Figure 5-8, protocols of higher flexibility produce better coverage results.

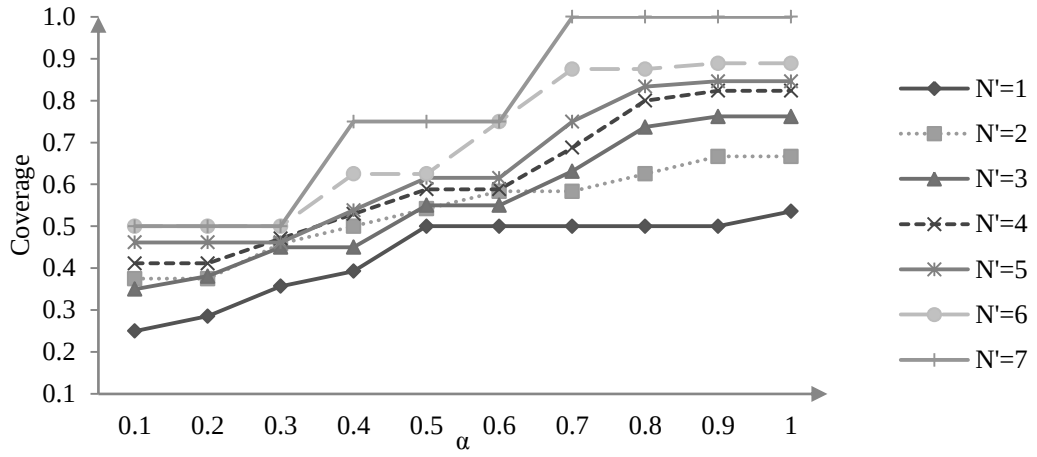


Figure 5-7: The coverage versus a different numbers of rules when using P_2

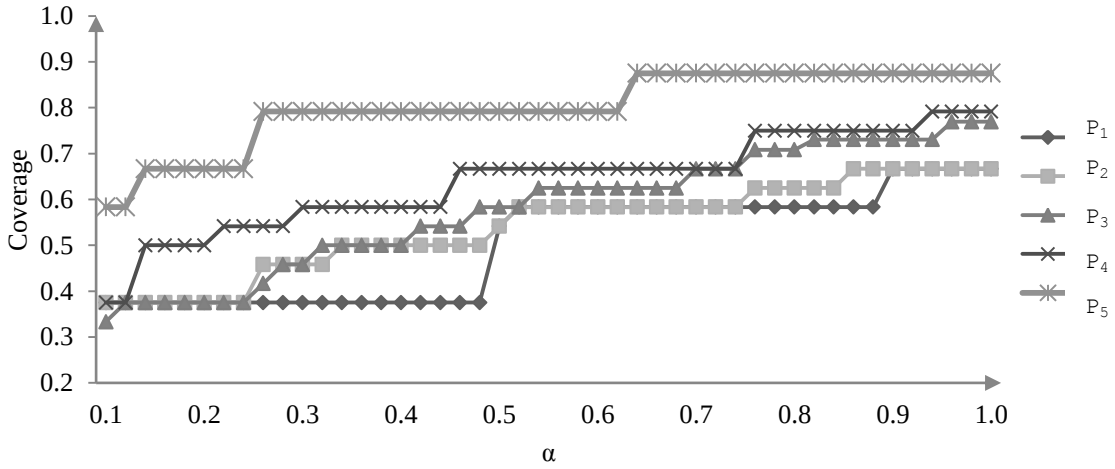


Figure 5-8: The coverage versus α obtained for different protocols

The overall performance expressed in terms of the AUC values is visualized in Figure 5-9. Again the superiority of the most flexible protocols is visible.

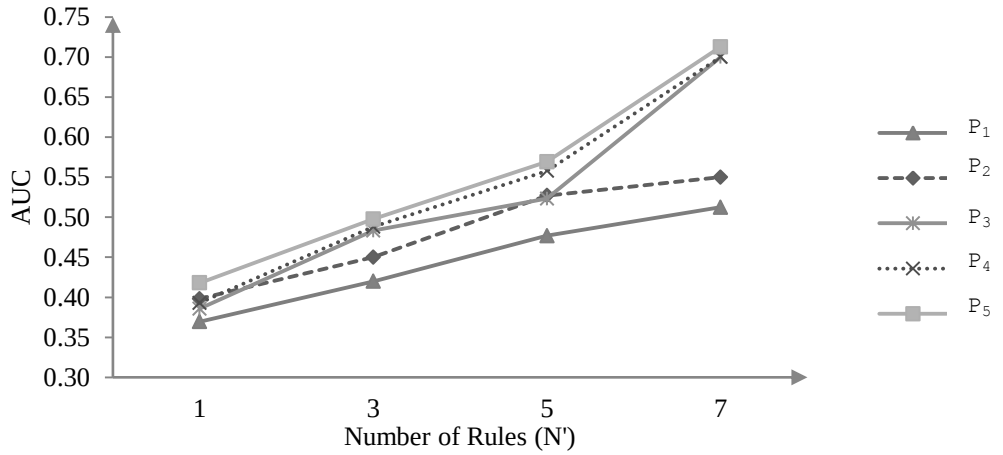


Figure 5-9: AUC as a function of the number of rules

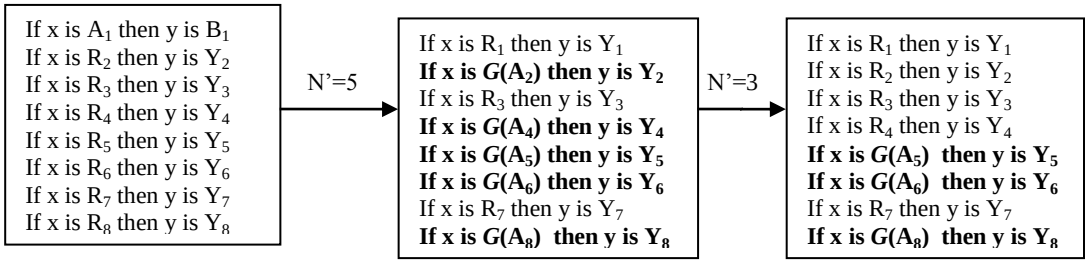


Figure 5-10: The selected subsets of rules (in boldface) obtained for different numbers of selected rules (for protocol P3)

The reduced list of rules is presented in Figure 5-10. Noticeable is a fact that with the reduction of the number of rules some of them are retained, say rule R_6 and R_8.

Mortgage applications assessment rule–based system-Assessment of a mortgage application normally based on evaluating the market value and location of the house, the applicant's asset and income, and repayment plan. A collection of rules is shown in Table 5-4.

Table 5-4: Rules for mortgage loan assessment

If (Asset is Low) and (Income is Low) then (Application is Low)
If (Asset is Low) and (Income is Medium) then (Application is Low)
If (Asset is Low) and (Income is High) then (Application is Medium)
If (Asset is Low) and (Income is Very High) then (Application is High)
If (Asset is Medium) and (Income is Low) then (Application is Low)
If (Asset is Medium) and (Income is Medium) then (Application is Medium)
If (Asset is Medium) and (Income is High) then (Application is High)
If (Asset is Medium) and (Income is Very High) then (Application is High)
If (Asset is High) and (Income is Low) then (Application is Medium)
If (Asset is High) and (Income is Medium) then (Application is Medium)
If (Asset is High) and (Income is High) then (Application is High)
If (Asset is High) and (Income is Very High) then (Application is High)

The results expressed in terms of the coverage treated as a function of the number of retained rules are summarized in Figure 5-11 and 5-12. The main trends are apparent. Furthermore the quantification of the improvements resulting from the increase of the number of rules involved is visible; a substantial jump is present when using more than 4 rules.

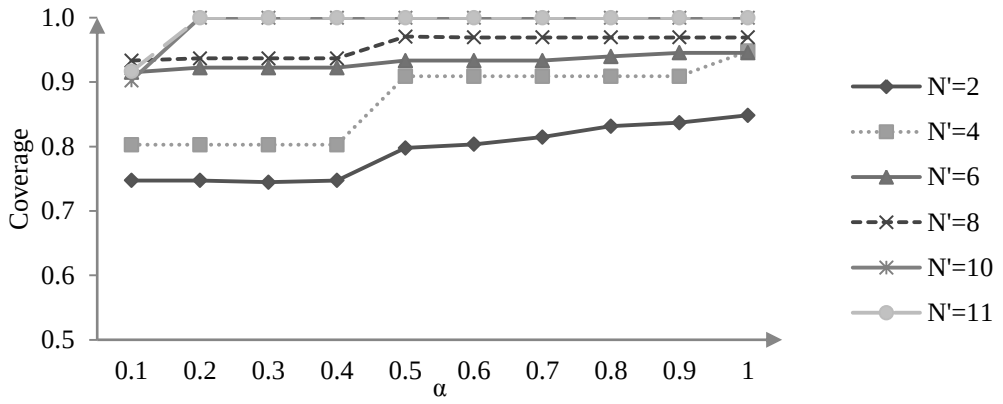


Figure 5-11: The coverage produced by the different number of rules using P_5

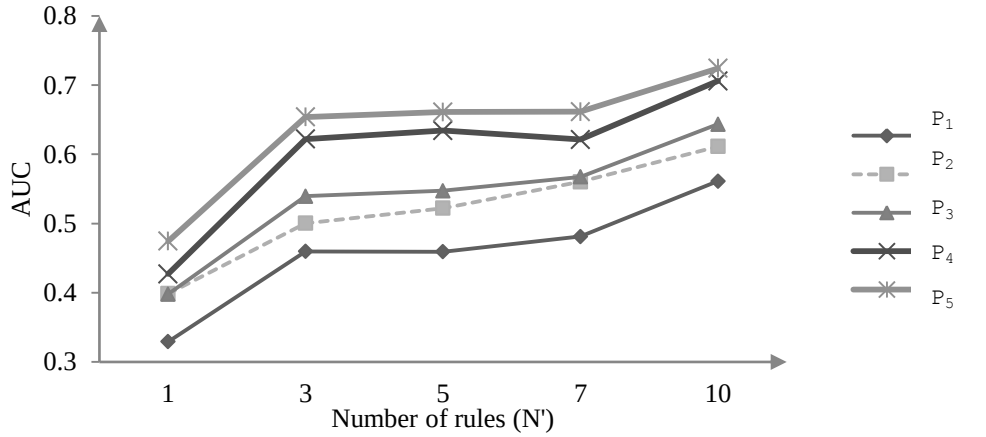


Figure 5-12: Area under curve AUC

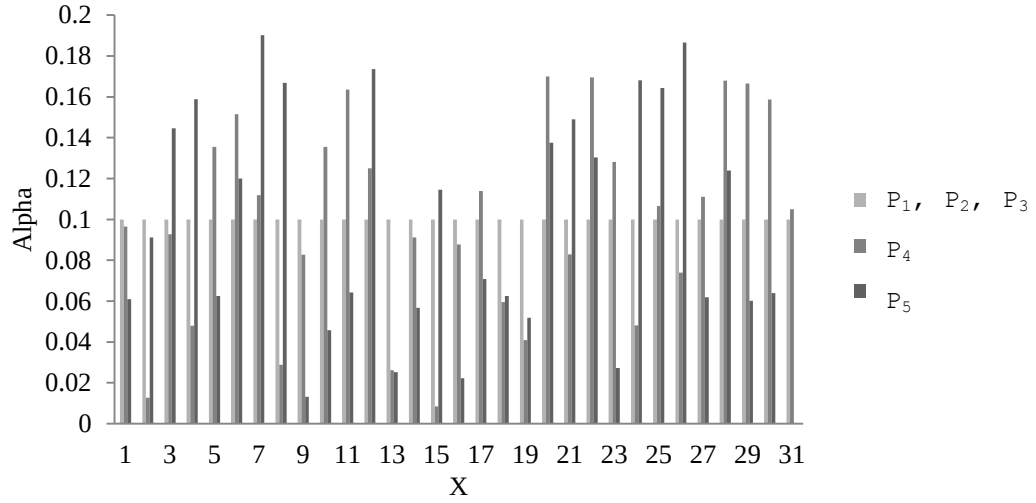


Figure 5-13: The allocation of information granularity for alpha, α=0.1

Figure 5-13 shows a distribution of the allocation of granularity realized with the use of the protocol P_1 to P_5; apparently, the distribution becomes non-uniform over the input space.

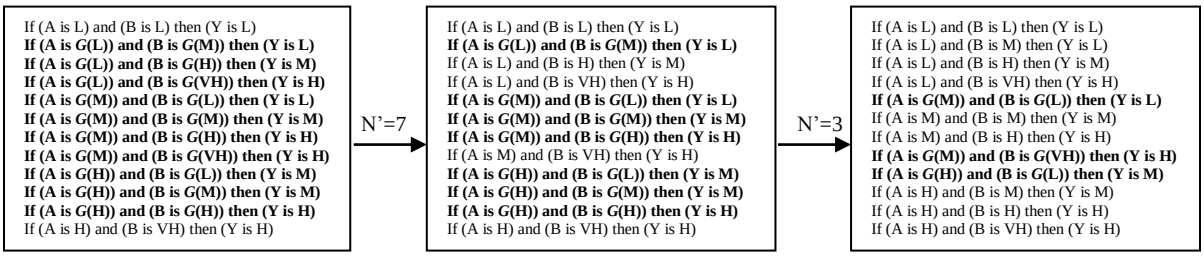


Figure 5-14: The selected rules when using P_3

The subset of selected rules when using P_3 is shown in Figure 5-14. As an example, we look at the reduction resulting in 3 rules:

*Rule 5: if (asset is **G**(medium)) and (income is **G**(low)) then (application is low)*

*Rule 8: if (asset is **G**(medium)) and (income is **G**(very-high)) then (application is high)*

*Rule 9: if (asset is **G**(high)) and (income is **G**(low)) then (application is medium)*

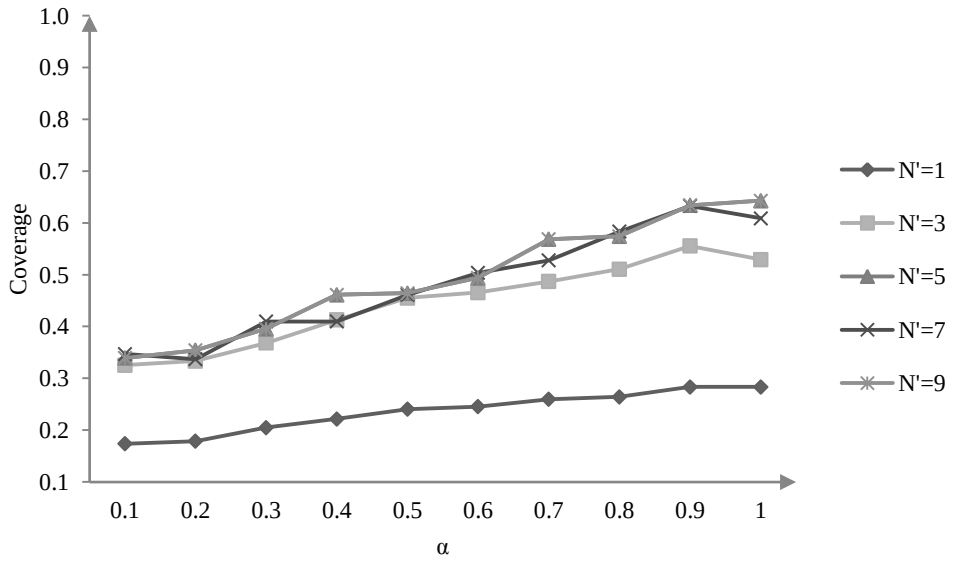
It is worth noting that the rules are representative of the three categories of applications.

Aircraft landing control problem -The aircraft landing control problem is dealing with the two important parameters called the velocity and the height. The main objective is to control the landing approach of an aircraft by desired downward velocity that is proportional to the square of the height. For example, at higher altitudes, a large downward velocity is desired and as the altitude (height) diminishes, the desired downward velocity gets smaller and smaller. Finally, as the height becomes vanishingly small, the downward velocity also goes to zero. Therefore, the aircraft will descend form altitude promptly, so that the touch down process is very gently to avoid damage. The pertinent rules are shown in Table 5-5.

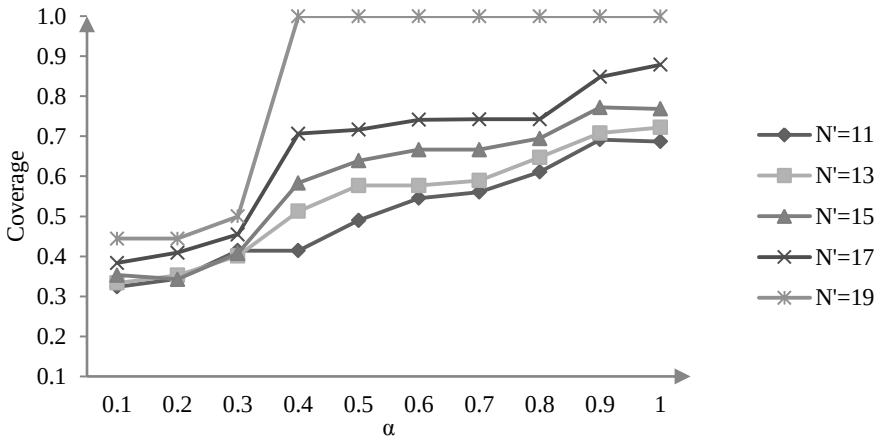
Table 5-5: Rules for the aircraft landing control problem

1. If (Height is L) and (Velocity is DL) then (Control force is Z)
2. If (Height is L) and (Velocity is DS) then (Control force is DS)
3. If (Height is L) and (Velocity is Z) then (Control force is DL)
4. If (Height is L) and (Velocity is US) then (Control force is DL)
5. If (Height is L) and (Velocity is UL) then (Control force is DL)
6. If (Height is M) and (Velocity is DL) then (Control force is US)
7. If (Height is M) and (Velocity is DS) then (Control force is Z)
8. If (Height is M) and (Velocity is Z) then (Control force is DS)
9. If (Height is M) and (Velocity is US) then (Control force is DL)
10. If (Height is M) and (Velocity is UL) then (Control force is DL)
11. If (Height is S) and (Velocity is DL) then (Control force is UL)
12. If (Height is S) and (Velocity is DS) then (Control force is US)
13. If (Height is S) and (Velocity is Z) then (Control force is Z)
14. If (Height is S) and (Velocity is US) then (Control force is DS)
15. If (Height is S) and (Velocity is UL) then (Control force is DL)
16. If (Height is NZ) and (Velocity is DL) then (Control force is UL)
17. If (Height is NZ) and (Velocity is DS) then (Control force is UL)
18. If (Height is NZ) and (Velocity is Z) then (Control force is Z)
19. If (Height is NZ) and (Velocity is US) then (Control force is DS)
20. If (Height is NZ) and (Velocity is UL) then (Control force is DS)

As before the main results are summarized in Figure 5-15 and 5-16.

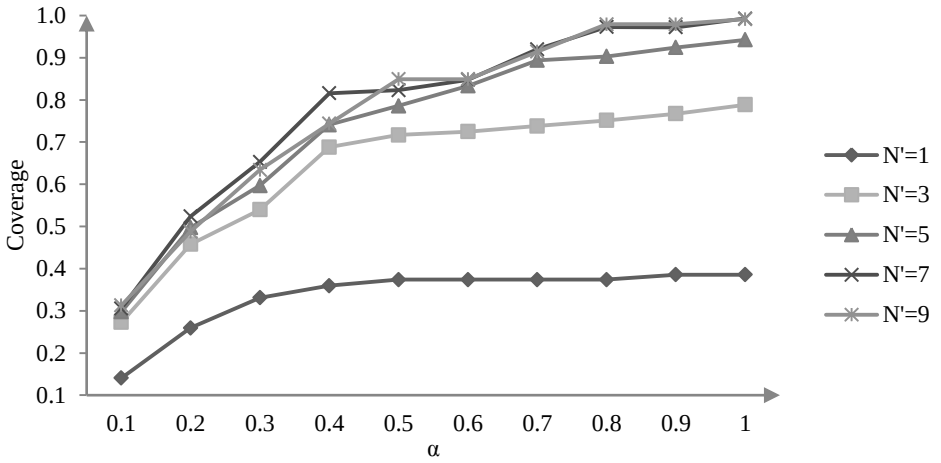


(a)

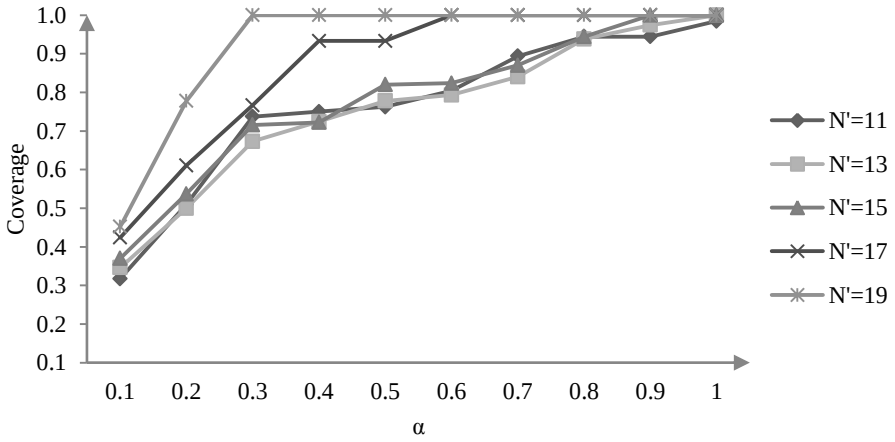


(b)

Figure 5-15: The plot of coverage K (α) regarded as a function of α using P1: (a) N'=1, 3, 5, 7, and (b) N'=11, 13, 14, 17, and 19



(a)



(b)

Figure 5-16: The plot of coverage K (α) regarded as a function of α using P4: (a) N'=1, 3, 5, 7, and (b) N'=11, 13, 14, 17, and 19

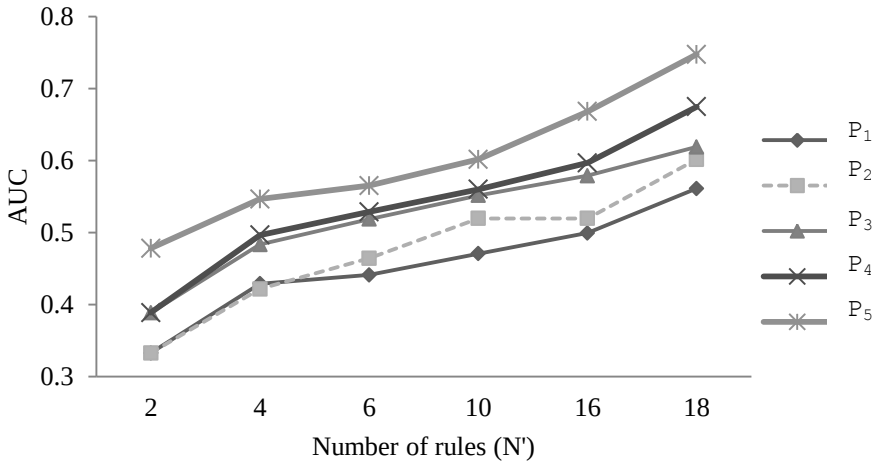


Figure 5-17: Area under curve AUC

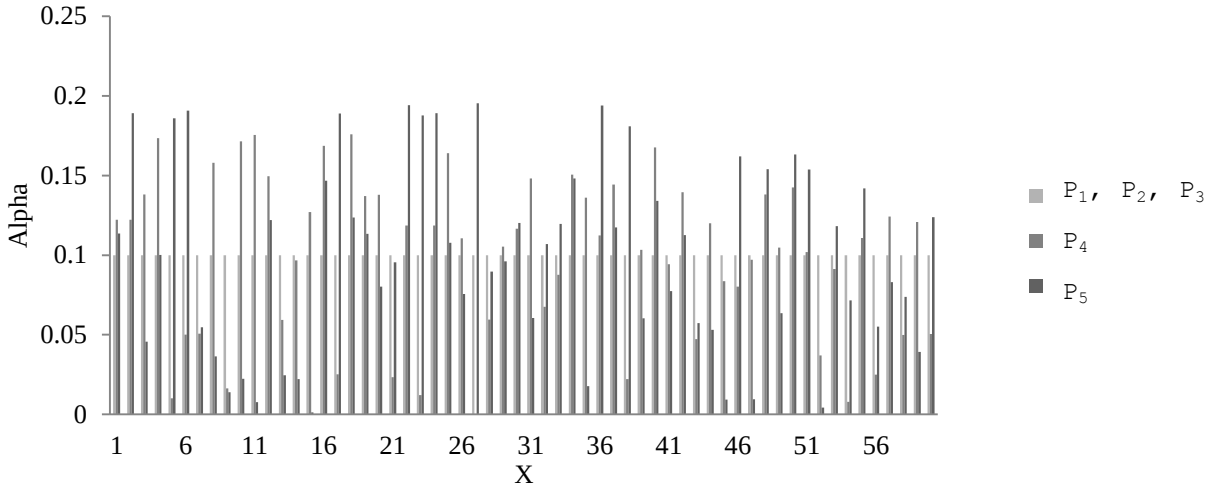


Figure 5-18: The allocation of information granularity for alpha, α=0.1

Figure 5-18 shows a distribution of the allocation of granularity realized with the use of the protocol P1 to P5; apparently, the distribution becomes non-uniform over the input space.

Equally interesting are the rules obtained through the reduction process. For instance, for α=0.5 and 4 rules we have,

*Rule 1 : If (Height is **G**(Low) and (Velocity is **G**(Down Large) then (Control force is Zeros)*

*Rule 10: If (Height is **G**(Medium) and (Velocity is **G**(Up Large) then (Control force is Down Large)*

*Rule 14 : If (Height is **G**(Small) and (Velocity is **G**(Up Small) then (Control force is Down Small)*

*Rule 16 : If (height is **G**(Near Zero) and (velocity is **G**(Down Large) then (Control is Up Large),*

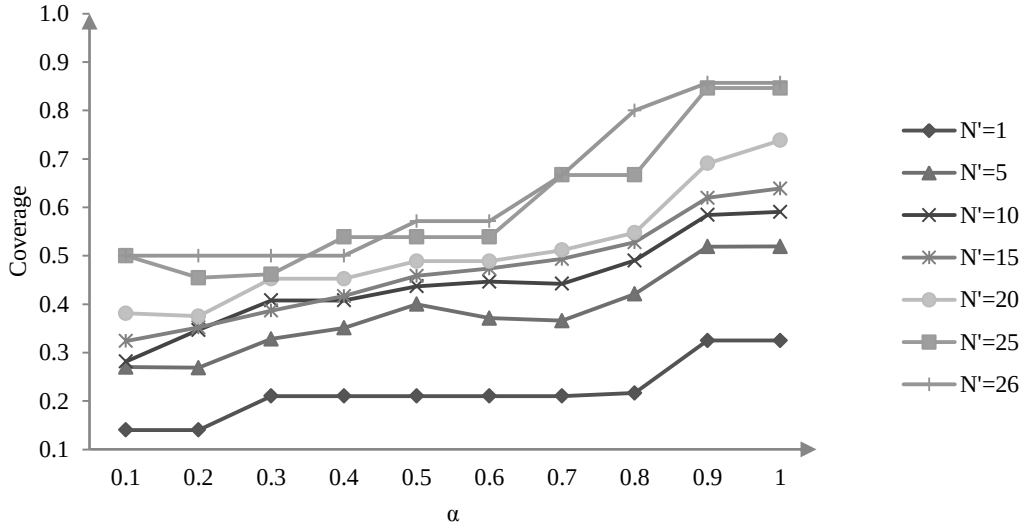
which capture the essence of the control strategy.

Service center operation data- The rules, having 3 inputs and a single output describing the functioning of the center are presented in Table 5-6. The overall number of the rules is 27.

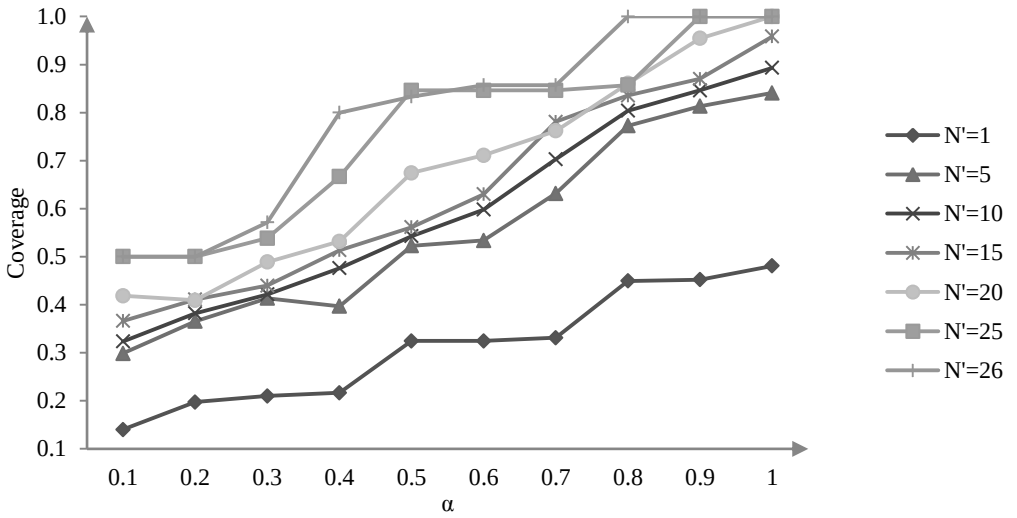
Table 5-6: Rules for the service center

1. If (Mean_Delay is VS) and (#of server is S) and (Utilization_Factor is L) then (# of spare is VS)
2. If (Mean_Delay is S) and (#of server is S) and (Utilization_Factor is L) then (# of spare is VS)
3. If (Mean_Delay is M) and (#of server is S) and (Utilization_Factor is L) then (# of spare is VS)
4. If (Mean_Delay is VS) and (#of server is M) and (Utilization_Factor is L) then (# of spare is VS)
5. If (Mean_Delay is S) and (#of server is M) and (Utilization_Factor is L) then (# of spare is VS)
6. If (Mean_Delay is M) and (#of server is M) and (Utilization_Factor is L) then (# of spare is VS)
7. If (Mean_Delay is VS) and (#of server is L) and (Utilization_Factor is L) then (# of spare is S)
8. If (Mean_Delay is S) and (#of server is L) and (Utilization_Factor is L) then (# of spare is S)
9. If (Mean_Delay is M) and (#of server is L) and (Utilization_Factor is L) then (# of spare is VS)
10. If (Mean_Delay is VS) and (#of server is S) and (Utilization_Factor is M) then (# of spare is S)
11. If (Mean_Delay is S) and (#of server is S) and (Utilization_Factor is M) then (# of spare is S)
12. If (Mean_Delay is M) and (#of server is S) and (Utilization_Factor is M) then (# of spare is VS)
13. If (Mean_Delay is VS) and (#of server is M) and (Utilization_Factor is M) then (# of spare is RS)
14. If (Mean_Delay is S) and (#of server is M) and (Utilization_Factor is M) then (# of spare is S)
15. If (Mean_Delay is M) and (#of server is M) and (Utilization_Factor is M) then (# of spare is VS)
16. If (Mean_Delay is VS) and (#of server is L) and (Utilization_Factor is M) then (# of spare is M)
17. If (Mean_Delay is S) and (#of server is L) and (Utilization_Factor is M) then (# of spare is RS)
18. If (Mean_Delay is M) and (#of server is L) and (Utilization_Factor is M) then (# of spare is S)
19. If (Mean_Delay is VS) and (#of server is S) and (Utilization_Factor is H) then (# of spare is VL)
20. If (Mean_Delay is S) and (#of server is S) and (Utilization_Factor is H) then (# of spare is L)
21. If (Mean_Delay is M) and (#of server is S) and (Utilization_Factor is H) then (# of spare is M)
22. If (Mean_Delay is VS) and (#of server is M) and (Utilization_Factor is H) then (# of spare is M)
23. If (Mean_Delay is S) and (#of server is M) and (Utilization_Factor is H) then (# of spare is M)
24. If (Mean_Delay is M) and (#of server is M) and (Utilization_Factor is H) then (# of spare is S)
25. If (Mean_Delay is VS) and (#of server is L) and (Utilization_Factor is H) then (# of spare is RL)
26. If (Mean_Delay is S) and (#of server is L) and (Utilization_Factor is H) then (# of spare is M)
27. If (Mean_Delay is M) and (#of server is L) and (Utilization_Factor is H) then (# of spare is RS)

The summary of the results is presented in Figure 5-19 and 5-20.



(a)



(b)

Figure 5-19: The plot of coverage κ (α) regarded as a function of using N'=1, 5, 10, 15, 20, 25, and 25: (a) P_1 and (b) P_2

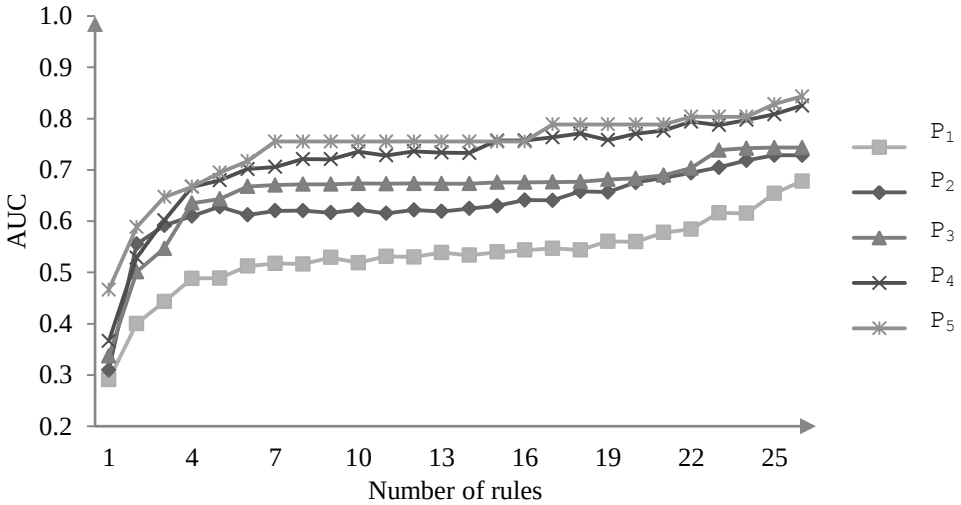


Figure 5-20: Area under curve AUC

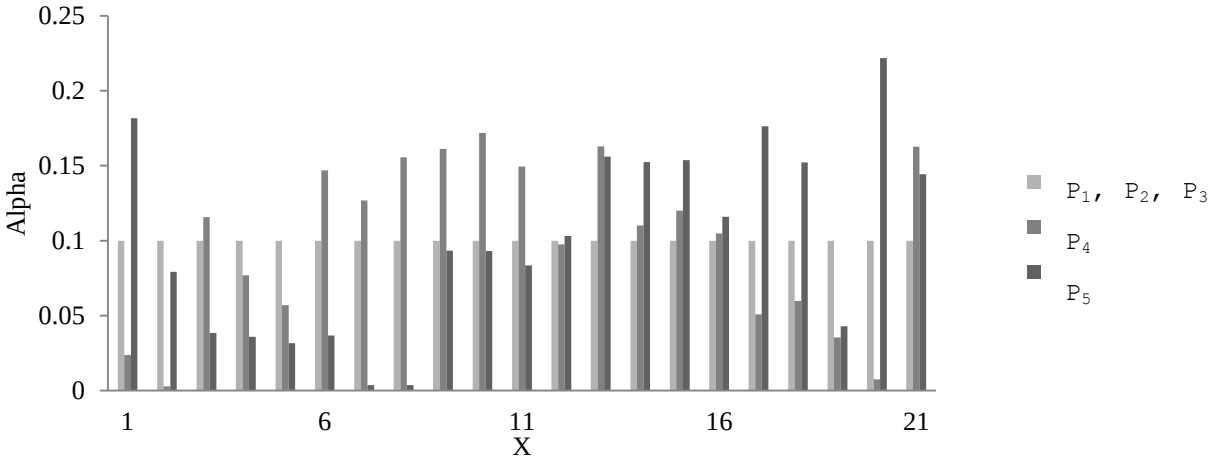


Figure 5-21: The allocation of information granularity for alpha, α=0.1

Figure 5-21 shows a distribution of the allocation of granularity realized with the use of the protocol P_1 to P_5; apparently, the distribution becomes non-uniform over the input space.

Figure 5-22 displays the selected of rules for using protocol P_1 with N'=15, N'=10 and N'=5. Form the figure we can see that in most cases the subset of the reduced rules is subset of the selected rules before. Here we observe that the subsets of the reduced rules exhibit an interesting "nesting" property meaning that the extended rules space builds upon the rules selected so far.

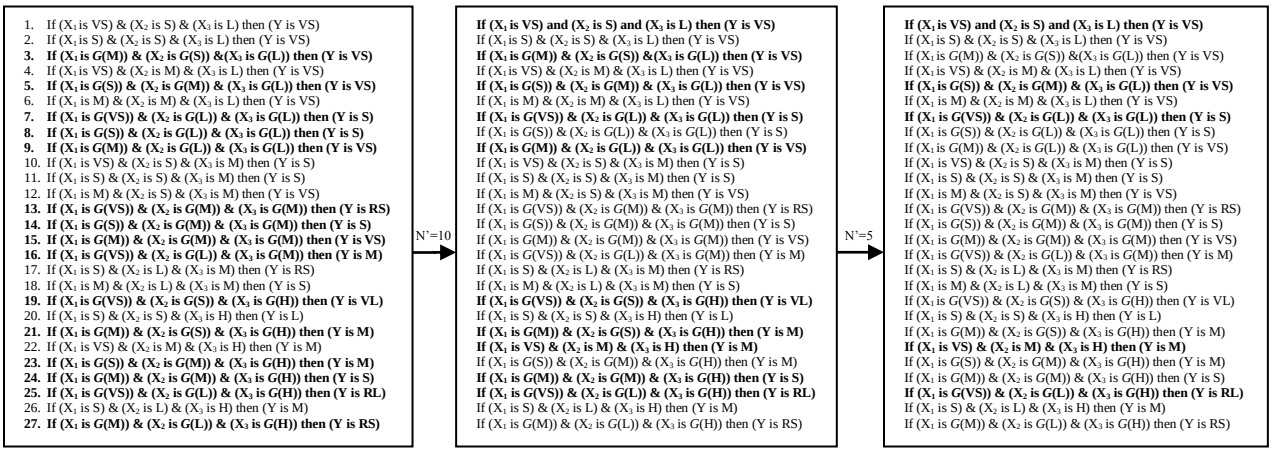
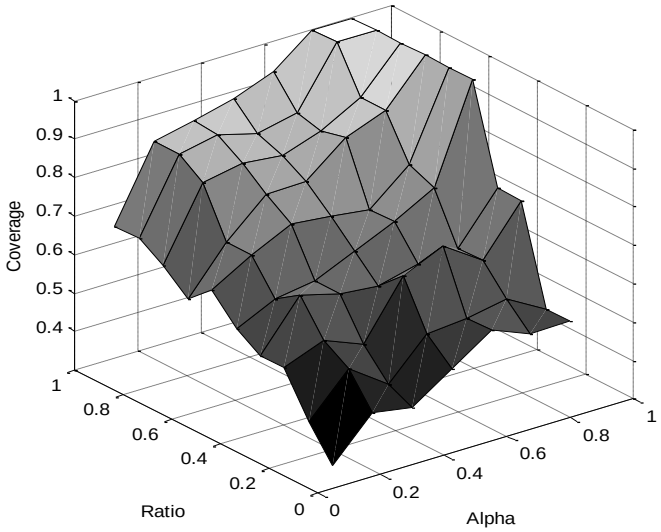
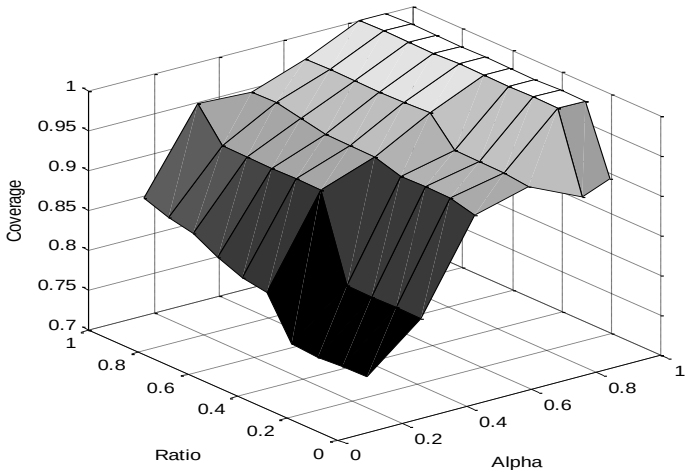


Figure 5-22: The selected rules for different number of selected rules using P1

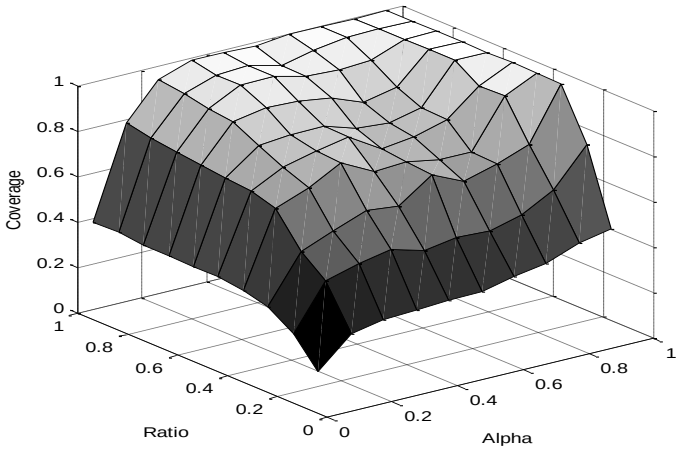
A concise summary of the results obtained for the series of experiments is presented in Figure 5-23. Here we visualize the coverage as a function of a fraction of rules retained (ratio). While the monotonicity character of this relationship is present.



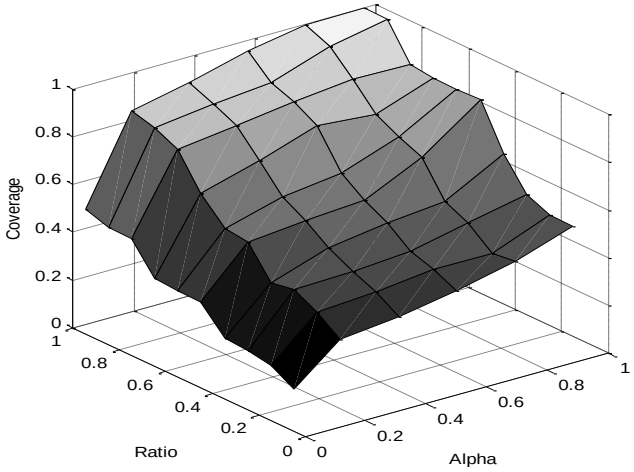
(a)



(b)



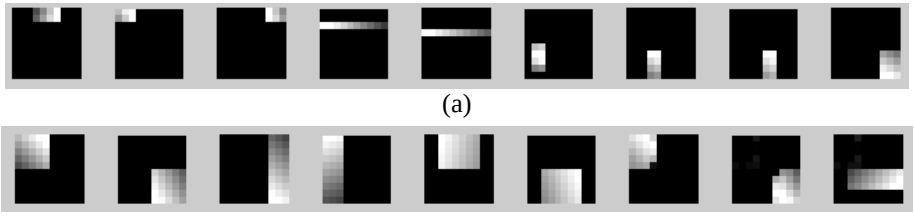
(c)



(d)

Figure 5-23: Coverage as a function of the fraction of rules retained for data: (a) Synthetic, (b) Applicant, (c) Aircraft, and (d) Service. In all cases protocol P5 was used

Image data- In what follows; we discuss a formation of a granular fuzzy associative memory (FAM). It is built by selecting an optimal subset of original rules building the memory in which the associated items A_k and B_k are stored (Kosko 1992). In Image 1 dataset, we consider 9 pairs of input-output patterns, see Figure 5-24. These grey-scale images $x^{\xi} \in [0,1]^{10x10}$, ξ=1,2,...,9. Each image is treated as a finite vector (with 81 coordinates).

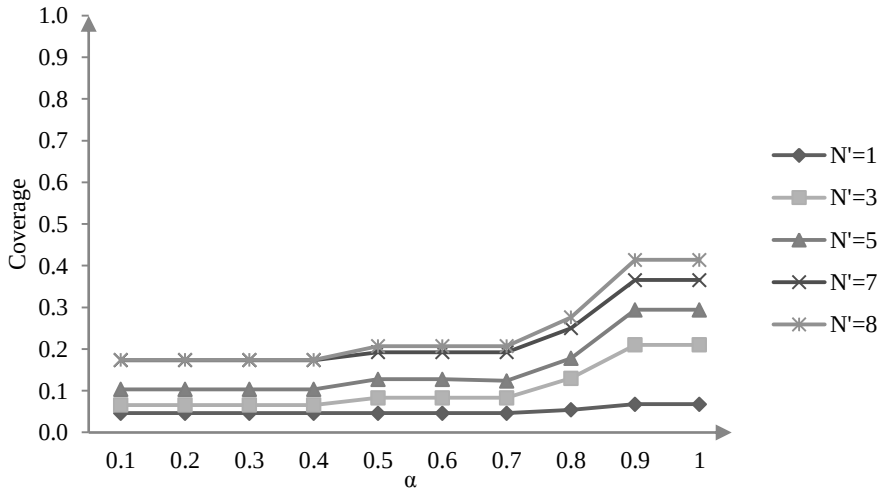


(a)

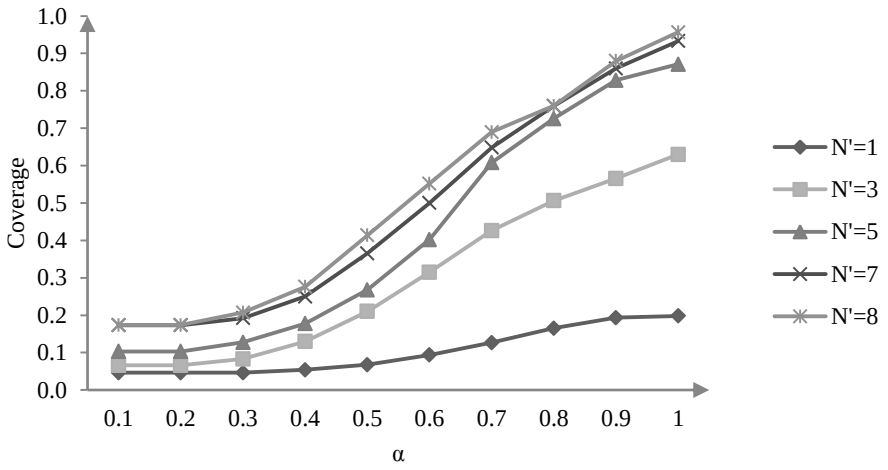
(b)

Figure 5-24: Pattern for Image1 dataset: (a) the input patterns and (b) the corresponding output patterns

Figure 5-25 displays the result of using protocol P_1 and protocol P_2 to construct the granular fuzzy rules. The graph shows that protocol P_2 outperform protocol P_1. Moreover, the coverage values are also increasing when we increase the number of rules used to construct the granular fuzzy rule. Next Figure 5-26 shows the results of using P_1, P_2, P_3, and P_4.



(a)



(b)

Figure 5-25: The plot of coverage κ (α) regarded as a function of alpha (α): (a) using Protocol P1 (b) using protocol P2

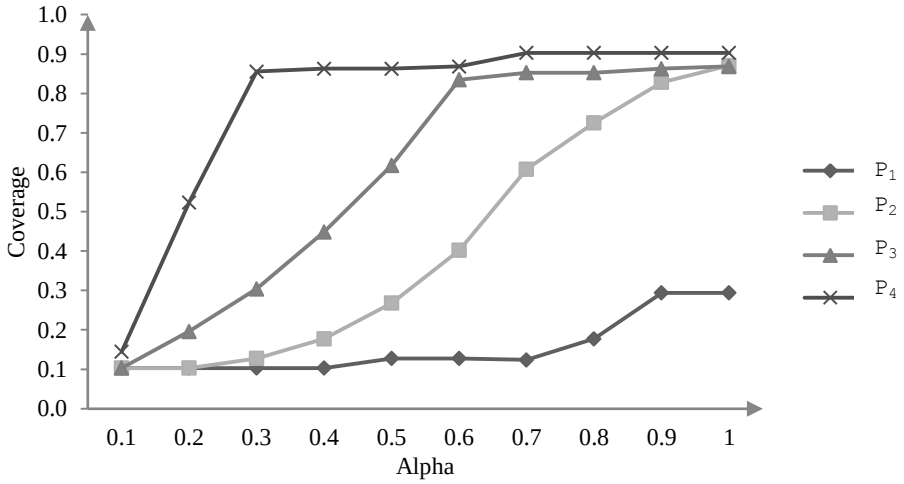
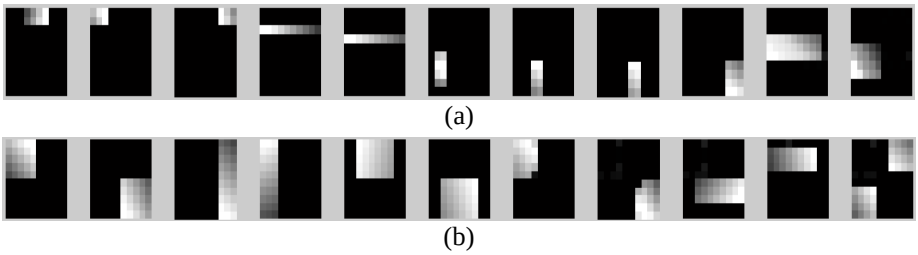


Figure 5-26: The coverage produced by the four protocols using N'=5

In Image 2 dataset, we consider the 11 patterns shown in Figure 5-27.

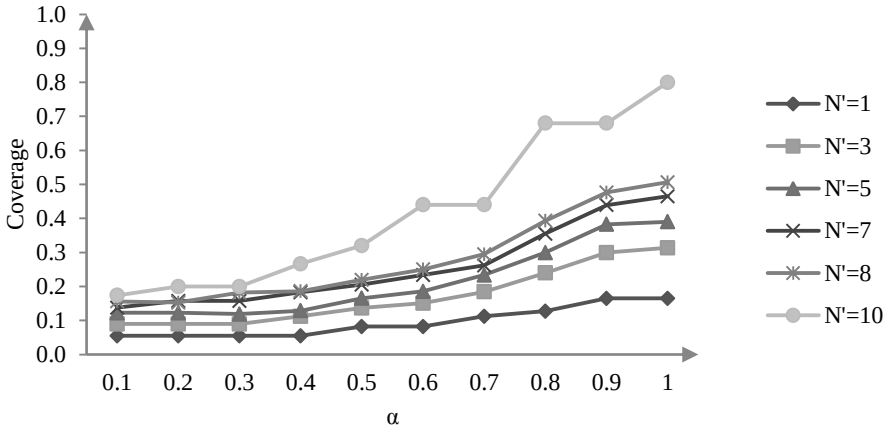


(a)

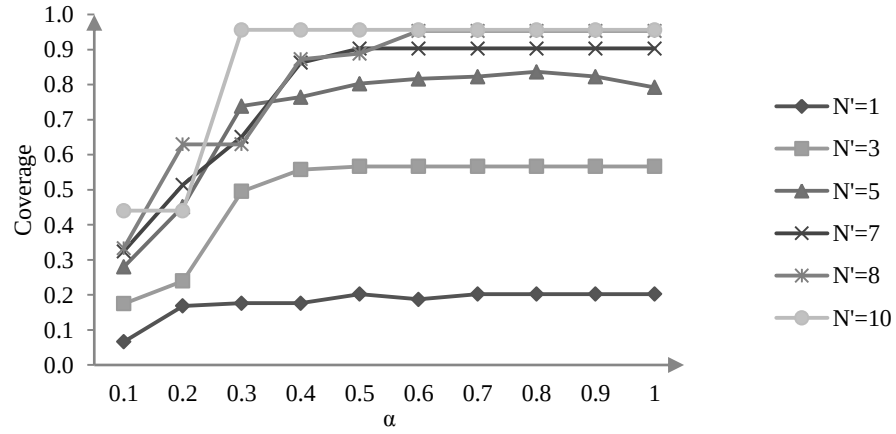
(b)

Figure 5-27: Pattern for Image2 dataset: (a) the input patterns and (b) the output patterns

The results are reported in Figure 5-28 and 5-29. The increased coverage is observed both when increasing the level of granularity and the number of selected rules.



(a)



(b)

Figure 5-28: The plot of coverage κ (α) regarded as a function of alpha (α): (a) using Protocol P1 (b) using Protocol P4

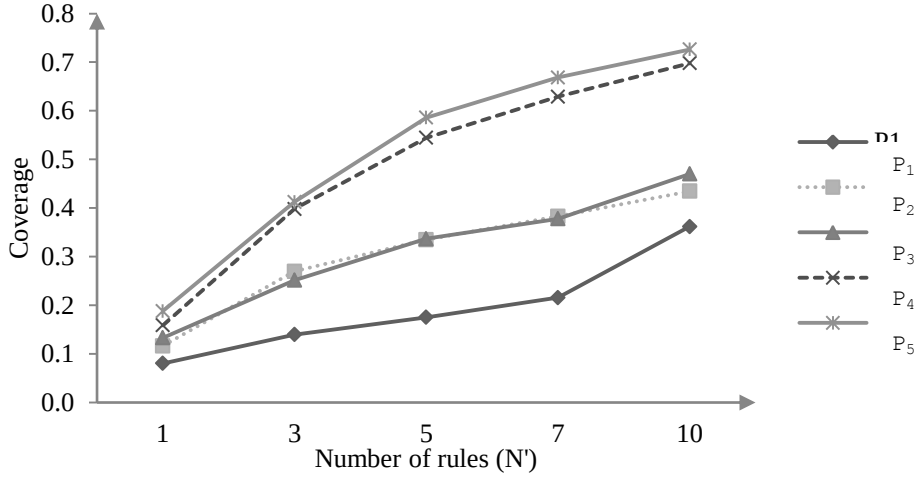


Figure 5-29: Area under curve AUC

Figure 5-30 shows a distribution of the allocation of granularity realized with the use of the protocol P_1 to P_5; apparently, the distribution becomes non-uniform over the input space.

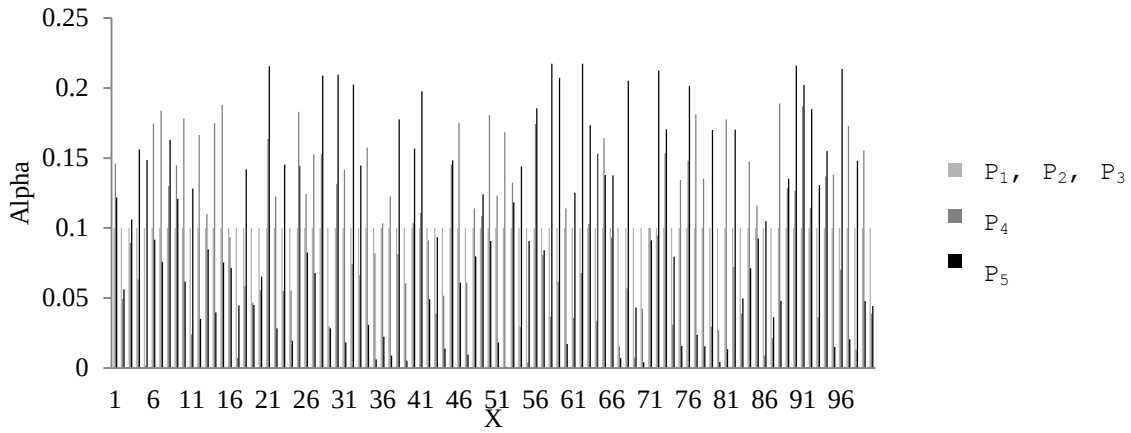


Figure 5-30: The allocation of information granularity for alpha, α=0.1

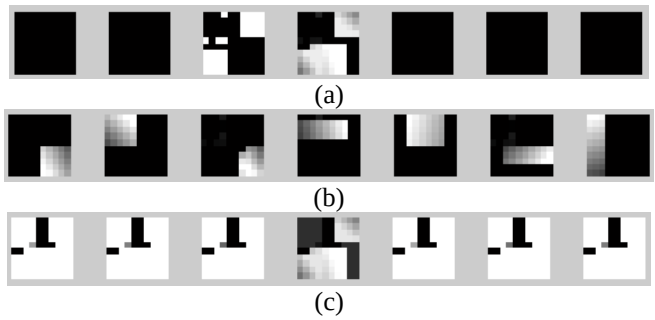


Figure 5-31: The output image (a) the lower output B-, (b) the original output, and (c) the upper output B+

Figure 5-31 shows the output when using a subset of 4 rules to form the granular fuzzy rule based.

5.6 Conclusions

In this chapter, we present a general issue of structural compression of rule-based systems as inherently associated with the emergence of granular constructs. Information granularity is reflective of the increased level of abstraction of the reduced set of rules. Information granularity is sought as an essential asset whose prudent allocation is behind the design of optimally reduced rule-based systems. The experimental part of the study shows essential linkages among the quality of the granular fuzzy rules and the number of retained rules and the admitted level of information granularity.

It has to be noted that the granular fuzzy sets form a general concept however in this study we focused on their interval realization. The entire development was presented in this way for clarity purposes (given our intent to concentrate on the concept). Nevertheless considerations of other realizations of the granular constructs follow the same general scheme and require some slight modifications.

6.The Development of Granular Takagi-Sugeno Fuzzy Model

In this chapter we develop a comprehensive framework of Granular Takagi-Sugeno fuzzy model. The construction of this model is based on the granular realization of the information granules (prototypes) with its corresponding granular firing strength (membership grades). In Section 6.1, we explain about the introduction of the research. In section 6.2, we discuss the underlying concept of Takagi-Sugeno fuzzy model. In the sequel, we discuss the granular fuzzy clusters and in Section 6.4 we briefly elaborate on the information granularity as a design asset and its optimal allocation. In Section 6.5, we explain about the performance index criterion quantifying the performance of the granular TS model. A suite of protocols of allocation of information granularity is presented. In Section 6.6, experimental studies are given. Finally, conclusions and some prospects of further studies are presented in Section 6.7.

6.1 Granular Takagi-Sugeno fuzzy model

The fuzzy models have proven to be remarkably successful in solving the nonlinear problem. The model given by fuzzy systems is more efficient compared to the traditional method especially when handling the uncertainty-based information in environments where the complexity is high and knowledge is low (Pedrycz 1993, Zadeh 1973). The *Takagi-Sugeno* (TS) is the most widely used fuzzy model. The TS model is a combination of the fuzzy logic and a mathematical function. The TS model represents a nonlinear system using the fuzzy rules in the form of a set of local affine models, which are connected by the fuzzy membership grades.

The construction of TS includes two steps. The first step is the determination of the fuzzy sets (membership grades) for the antecedent part. The second step is the estimation of the consequent parameter. When taking a close look at these constructs, there are some remarkable commonalities among all of them. Fuzzy models are numeric constructs, therefore for any input, formed in the corresponding numeric output, there are no ideal fuzzy models for which all data coincide with the results produced by the model.

To make the model more in rapport it becomes beneficial to construct a fuzzy model whose outputs are non-numeric that is more abstract. This motivated us to generalize the TS model based on the concept of information granularity.

The objective of this research is to introduce the concept of Granular TS fuzzy model. The idea arises from the concept of information granularity as the vehicle for constructing the structure of the fuzzy model. We emphasize this development by using the term granular TS fuzzy model. Here, the numeric antecedent part of the fuzzy model is reconstructed again based on the concept of information granularity, say intervals. More specifically, the membership grades that will be used as the firing strength in constructing the predicted output are described in the non-numeric representation.

Formally speaking, the original TS fuzzy model "If **x** is A_i then $y = f_i(\mathbf{x}, a_i)$" composed by N rules is represent by Granular TS fuzzy model "if **x** is $G(A_i)$ then $y = f_i(\mathbf{x}, a_i)$" where $G(A_i)$ denotes a granular generalization (abstraction) of the original fuzzy set of condition A_i. This granular abstraction can be realized in the form of an interval-valued fuzzy set (or type-2 fuzzy set, in general), rough set, shadowed set (as mentioned earlier), probabilistic set, etc. The conclusion part (f_i) is formed by a certain local function.

6.2 The development of *Takagi-Sugeno* fuzzy model

TS fuzzy models are composed of rules

$$\text{if } \mathbf{x} \text{ is } A_i \text{ then } y = f_i(\mathbf{x}, a_i) \tag{6-1}$$

where A_i is a multivariable information granule (fuzzy set) defined in the input space (R^n) and f_i is a local function $\mathbf{R}^n \rightarrow \mathbf{R}$ equipped with some parameters a_i. The processing carried out within the model consists of two steps: (a) determination of activation of the individual rules, which for any $\mathbf{x} \in R^n$ returns the values $A_1(x)$, $A_2(x)$,…, $A_c(x)$, which could be treated as activation levels of the corresponding rules, and (b) aggregation of outcomes of local models f_i weighted by the activation levels. The aggregation is typically realized as the following sum of $y = \sum_{i=1}^{c} A_i(\mathbf{x}) f(\mathbf{x})$.

The advantages of this category of the models are apparent. Modularity of the models and their local character are one of the visible modeling assets: even a complex phenomenon can be easily modeled by far less complex (e.g., linear) and local relationships and admitting that their relevance is confined to some quite limited regions of the input space. The paradigm of *local* rather than *global* modeling is behind the principle of rule-based modeling. It contributes to the success of this form of system modeling.

The essence of the design comprises two main phases (whose realization could exhibit some algorithmic diversity) that is reflective of the rule-based architecture of the model,

(a) construction of information granules

(b) construction of local modes forming the conclusion parts of the rules

With regard to the first phase, the common practice is to form information granules forming the condition part of the rules through fuzzy clustering. Fuzzy C-Means along with its numerous variants is one among well-established and commonly used techniques to determine fuzzy sets (whose description is based on the partition matrix and prototypes both being determined during the clustering process). Based on the knowledge of the prototypes v_1, v_2, …, v_c, the associated membership functions A_1, A_2, …, A_c are expressed through a well-known formula

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

where || . || denotes a distance function (for the generic version of the FCM we use a Euclidean distance or its weighted version (Eqn. 2-12)). The fuzzification coefficient (m) assumes values greater than 1 and impacts the geometry of the clusters (shape of membership functions).
If the local models (f_i) associated with the respective rules are linear with respect to its parameters then an estimation of their optimal values is realized by solving a standard LSE problem for which there is an analytical solution.

Let us stress that most of the design of these models is realized in a supervised mode meaning that for clustering and parameter estimation we use a collection of input-output data ($\mathbf{x}_k$, $target_k$), k=1, 2, …, N. The standard modeling practices of using these data for training and testing purposes are exercised as well.

6.3 Granular Fuzzy Clusters

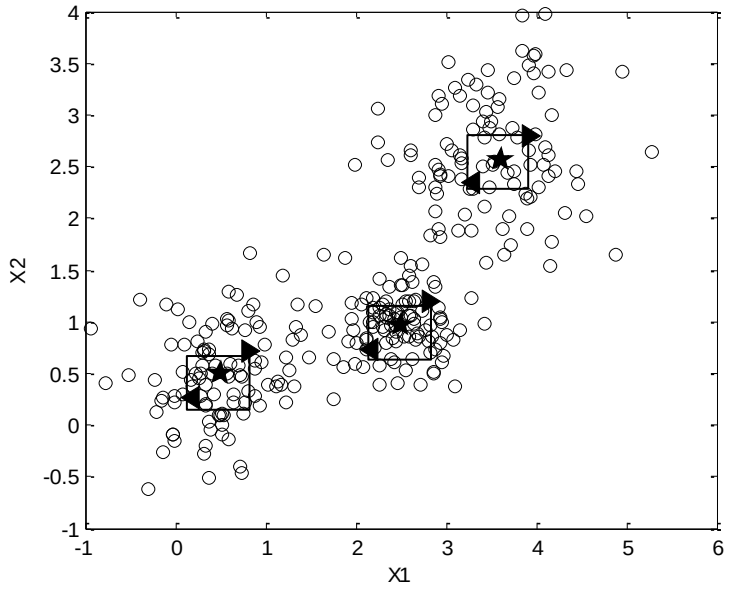
In the general architecture of granular TS models (Takagi and Sugeno 1985), the original numeric membership functions A_i are replaced by their granular counterparts built on the basis of the fuzzy sets occurring in the rules. In what follows we look at the formation of granular membership functions. The starting point of the entire construction are granular prototypes V_1, V_2,…, V_c formed around the numeric counterparts. More specifically, for the purpose of this study (and to focus on the essence of the construct), we assume that V_i are interval-valued prototypes, viz. some sets defined in the space of sets over R^n, that is $\mathbf{V}_i$ $P(R^n)$ where $P(.)$ stands for a family of sets (intervals).

The structure of the granular prototypes, $\mathbf{V}_i$ is built around the original prototype $\mathbf{v}_i$ by admitting some level of granularity ε assuming value in [0, 1]. In the simplest possible scenario,

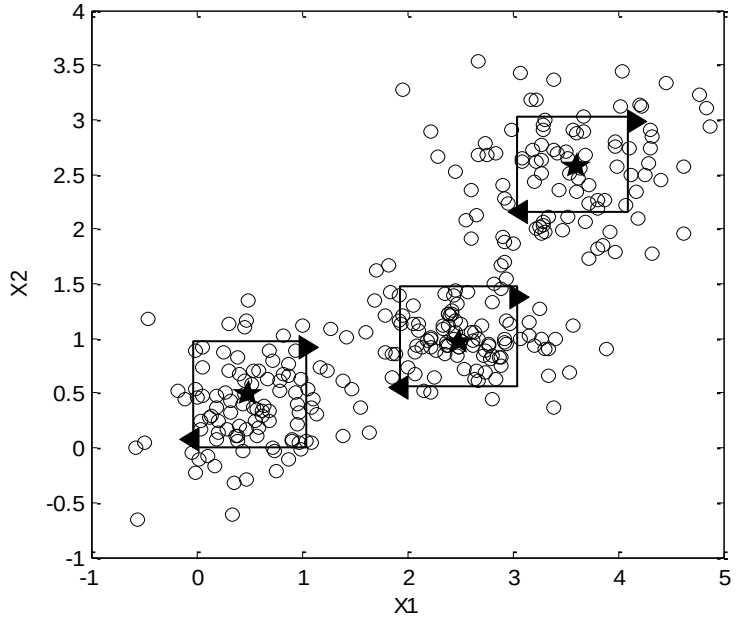
we can envision the following transformation using which an information granule of the prototype is being formed as the following expression (Pedrycz and Bargiela 2011),

$$V_{ij}=[\ v_{ij} - \varepsilon * range_j,\ v_{ij} + \varepsilon * range_j\ ] \quad (6\text{-}3)$$

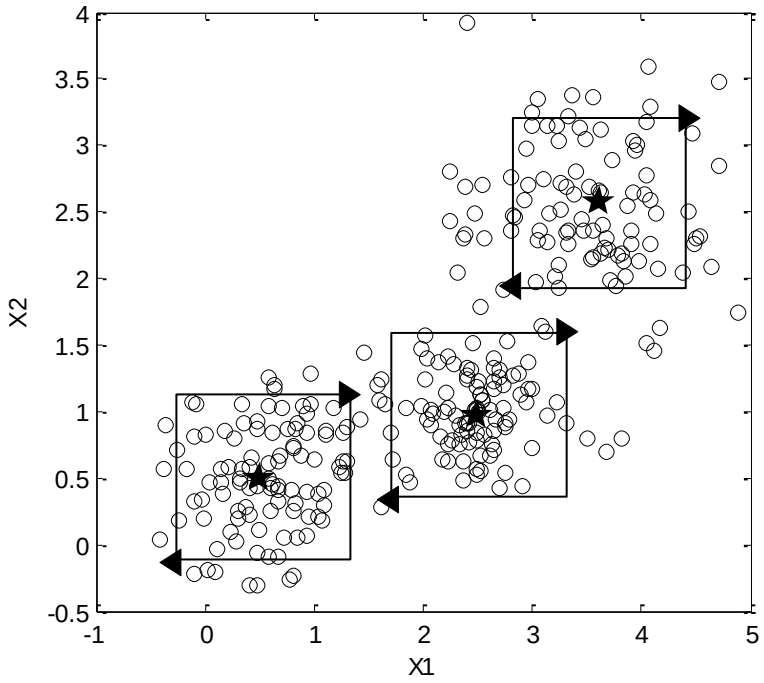
Where i=1,2,… c and j=1,2,…,n. Note that the prototype is made granular to the same extent with regard to all variables. All coordinate of prototype are transformed to the intervals that are symmetrically distributed around v_{ij} and equally affected by the imposed level of granularity. Figure 6-1 illustrates the representation of the granular prototypes.



(a)



(b)



(c)

Figure 6-1: The plot of the granular prototypes with its corresponding initial prototypes; (a) ε=0.1, (b) ε=0.2, and (c) ε=0.3

In this construction we have to consider a situation of overlap of v_j and v_i. The following is the analytical overlap condition for the granular prototypes where j≠i and i=1, 2,…, n:

$$\text{overlap}(v_j) = \begin{cases} 1 & v_j^+ < v_i^+ \,\&\, v_j^+ > v_i^- \text{ or } \quad v_j^+ > v_i^+ \,\&\, v_j^{-\cdot} < v_i^+ \\ 0 & \text{otherwise} \end{cases} \tag{6-4}$$

Next, because of their non-numeric nature, granular prototypes give rise to granular membership functions. Note that the original formulas for the membership functions developed in the generic FCM, the membership grades (functions) are determined on a basis of the distances between **x** and the (numeric) prototypes. Here the notion of distance has to be carefully revisited to properly account for the interval nature of the prototypes. While there are some well-known approaches to express the distance between granular constructs (say, a Hausdorff distance (Grzegorzewski 2004)), all of them return a single numeric value quantifying this distance. This view is somewhat limited as one could have expected a certain granular descriptor of the closeness. The simplest option here would be to establish some bounds of the values the distance could assume. We consider these extreme cases by looking at a single variable. Let us assume that for the j-th variable the bounds of the granular prototype $\mathbf{V}_i$ form the interval $[v_{ij}^-, v_{ij}^+]$. For the j-th coordinate of **x**, x_j, we consider two situations

i. $x_j \notin [\, v_{ij}^-, v_{ij}^+]$, the bounds of the distance are taken by considering the pessimists and optimistic scenario and computing the distances from the bounds of the interval by using the following formulas,

$$\min ((x_j - v_{ij}^-)^2, (x_j - v_{ij}^+)^2) \quad (6\text{-}5)$$

$$\max ((x_j - v_{ij}^-)^2, (x_j - v_{ij}^+)^2) \quad (6\text{-}6)$$

ii. $x_j \in [\, v_{ij}^-, v_{ij}^+]$, it is intuitive to accept that the distance is equal to zero (as x_j is included in this interval)

The distance computed on a basis of all variables $\| \mathbf{x} - \mathbf{V}_i \|^2$ is determined coordinate wise by involving the two situations outlined above. The minimal distance obtained in this way is denoted by $d_{min}(\mathbf{x}, \mathbf{V}_i)$, while the maximal one is denoted by $d_{max}(\mathbf{x}, \mathbf{V}_i)$. The following are the detail formulas for calculating the minimal and the maximal distances.

$$d_{min}(\mathbf{x}, \mathbf{V}_i) = \sum_{j \in K} \min ((x_j - v_{ij}^-)^2, (x_j - v_{ij}^+)^2) \quad (6\text{-}7)$$

$$d_{max}(\mathbf{x}, \mathbf{V}_i) = \sum_{j \in K} \max ((x_j - v_{ij}^-)^2, (x_j - v_{ij}^+)^2), \quad (6\text{-}8)$$

Where j=1, 2, …, n $x_j \notin [\, v_{ij}^-, v_{ij}^+]$. Having the distances obtained, we compute the two expressions,

$$w_1(\mathbf{x}) = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{min}(\mathbf{x}, \mathbf{V}_i)}{d_{min}(\mathbf{x}, \mathbf{V}_j)} \right)^{1/(m-1)}} \quad (6\text{-}9)$$

$$w_2(\mathbf{x}) = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{max}(\mathbf{x}, \mathbf{V}_i)}{d_{max}(\mathbf{x}, \mathbf{V}_j)} \right)^{1/(m-1)}}, \quad (6\text{-}10)$$

where m >1. These two expressions are used to calculate the lower and upper bounds of the interval-valued membership functions (induced by the granular prototypes). Again one has to proceed carefully with this construct. Let start with a situation when **x** is not included in any of the granular prototypes. In this case the granular membership grades are computed as follows:

$$u_i^-(\mathbf{x}) = \min (w_1(\mathbf{x}), w_2(\mathbf{x})) \quad (6\text{-}11)$$

$$u_i^+(\mathbf{x}) = \max (w_1(\mathbf{x}), w_2(\mathbf{x})) \quad (6\text{-}12)$$

If **x** is belong to $\mathbf{V}_i$ then apparently $u_i^-(\mathbf{x}) = u_i^+(\mathbf{x}) = 1$ (and this comes as a convincing assignment). Obviously, in this case $u_j^-(\mathbf{x})$ as well as $u_j^+(\mathbf{x})$ for all indexes "j" different from "i" are equal to zero.

As illustration, let us consider a one-dimensional case with three interval-valued prototypes V_1=[0.5, 1.0] V_2=[2.2, 2.7] and V_3=[4.0, 4.5]. The lower and upper bounds of membership function are displayed in Figure 6-2.

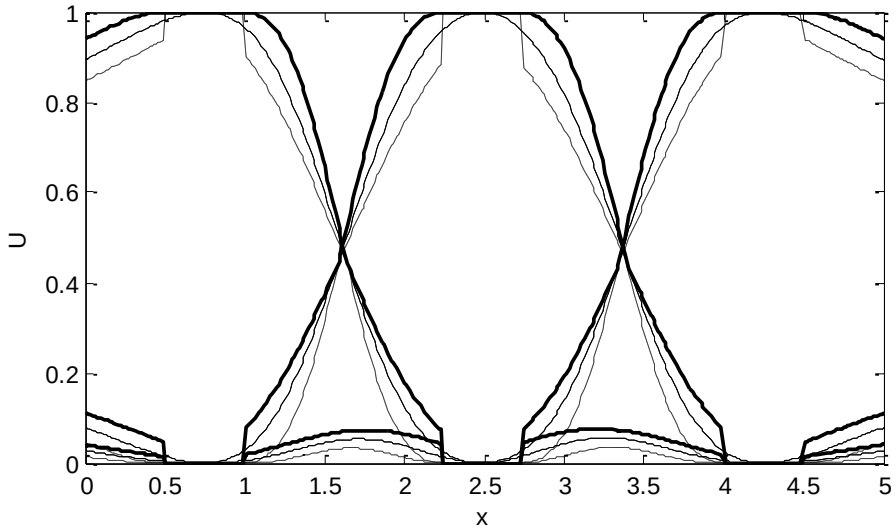


Figure 6-2: The plot of the membership grades and its corresponding granular membership grades. $ui^+(x)$ – thick line $ui^-(x)$ –dotted line and the original u_i – normal line

As expected, the bounds of the granular membership function differ over the universe of discourse. It is instructive to plot the differences, which visualize the regions where the highest differences in these membership grades are encountered. The corresponding plot is presented in Figure 6-3.

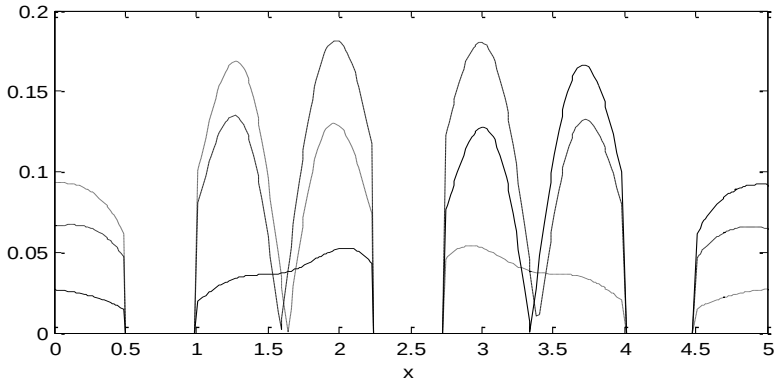


Figure 6-3: The plot of the difference between the upper and the lower membership grades

Interestingly, the most significant diffrence occur in-betwwen the prototypes –and this could have been expected as those are the regions where contributions coming from the prototypes are less definite (and this is quantified by the lower and upper bound of the membership values).

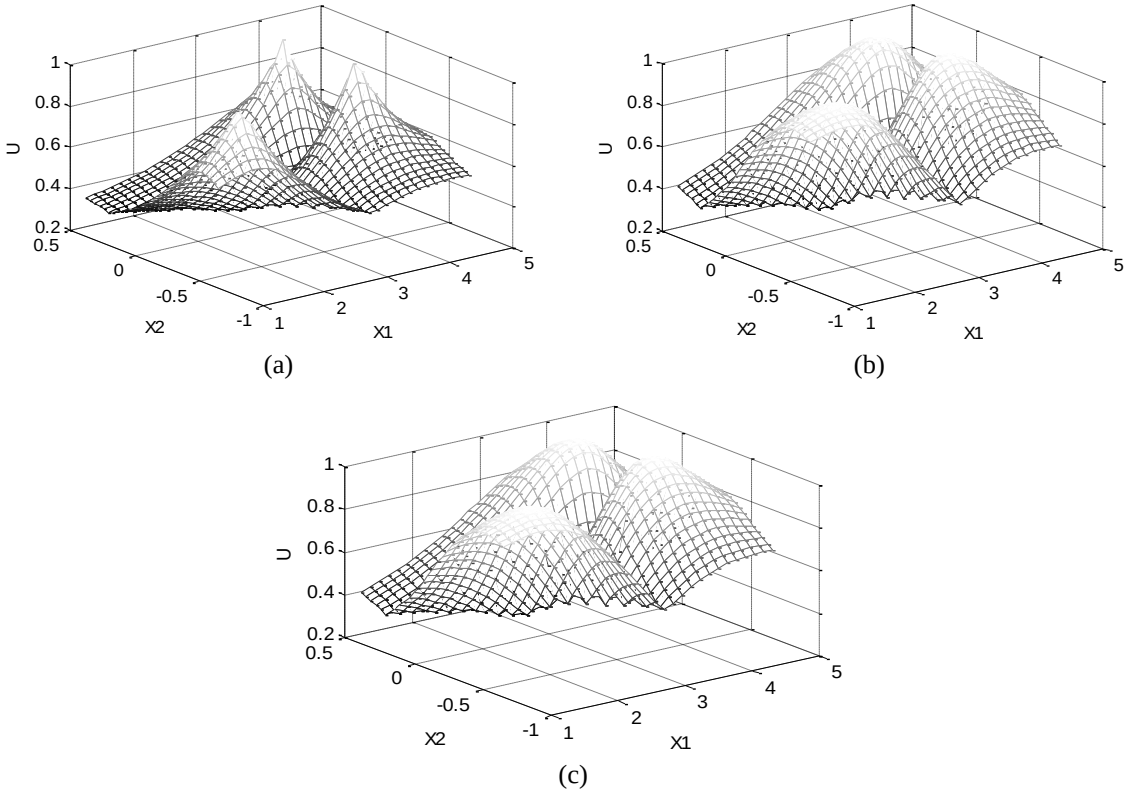


Figure 6-4: The plot of the membership grades; (a) standard membership grades, (b) granular membership grades (ui-(x)), and (c) (ui+(x))

6.4 Information granularity as a design asset and its optimal allocation

The original numeric prototypes $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_c$ are made granular by constructing intervals $\mathbf{V}_1$, $\mathbf{V}_2$, …, $\mathbf{V}_c$ around the prototypes. The formation of these multidimensional information granules is realized by implementing some protocols of allocation of information granularity. For a certain predetermined value of the level of information granularity $\varepsilon \in [0,1]$ the intervals are formed as follows;

Protocol 1(P_1*)*: The prototypes are interval-valued of the same length $\varepsilon/2 * \text{range}_j$. The corresponding interval granular prototypes are given as

$$[\, v_{ij} - \varepsilon/2 * \text{range}_j,\ v_{ij} + \varepsilon/2 * \text{range}_j \,] \tag{6-13}$$

i=1, 2,…, c; j=1, 2, …, n. The intervals are symmetrically spread around the original numeric values and all intervals are of the same length.

Protocol 2(P_2*)*: A uniform allocation of information granularity with asymmetric position of interval. It is similar to P_1 however it exhibit flexibility as we allow the asymmetric allocation information granules (intervals) meaning that the granular prototypes are given as

$$[\, v_{ij} - \gamma * range_j,\ v_{ij} + (1-\gamma) * range_j \,] \tag{6-14}$$

where $\gamma \in [0, 1]$ is used to control a level of asymmetry (asymmetry degree). If γ=1/2, this protocol reduces to P_1. The optimization concerned adjustment of the value of asymmetric (γ).

Protocol 3(P_3*)*: We admit asymmetric allocation of information granularity to values of the asymmetry degrees associated with the corresponding input variables. The granular prototype $v_{ij,}$ i=1,2,.. C and j=1,2,…,n are generalized and assume the form of the interval

$$[v_{ij} - \gamma_j * range_j,\ v_{ij} + (1-\gamma_j) * range_j \,] \tag{6-15}$$

where $\gamma_j \in [0, 1]$. In total, we have a vector of coefficients [γ_1, γ_2, …, γ_n].

In summary the search space explored by each of the protocols can be described as follows

Protocol	*Parameters*	*Dimensionality of search space*
P_1	ε	no optimization
P_2	γ	optimization of γ, $\gamma \in [0, 1]$, (1)
P_3	γ_j , j=1,2,…n	optimization of γ_1, γ_2, …, γ_n, (n)

In this study, we are using the Particle Swarm Optimization techniques to search for the optimal allocation of information granulation that maximum the performance index (coverage).

6.5 Performance index

The predicted output of the fuzzy model is in the interval-like form, $[\hat{y}_k^{-}, \hat{y}_k^{+}]$ inferred as

$$\hat{y}_k^{-} = \sum_{i=1}^{C} \tilde{u}_{ik}^{-} f_i(x_k, a_i) \tag{6-16}$$

$$\hat{y}_k^{+} = \sum_{i=1}^{C} \tilde{u}_{ik}^{+} f_i(x_k, a_i) \tag{6-17}$$

,

where the $u'_{ik}{}^{-}$ and $u'_{ik}{}^{+}$ the i-th rule's firing strength that represent by the following expression,

$$\tilde{u}_{ik}^{-} = \begin{cases} u_{ik}^{-} & \text{if} \quad f_i > 0 \\ u_{ik}^{+} & \text{if} \quad f_i \le 0 \end{cases} \tag{6-18}$$

$$\tilde{u}_{ik}^{+} = \begin{cases} u_{ik}^{-} & \text{if} \quad f_i \le 0 \\ u_{ik}^{+} & \text{if} \quad f_i > 0 \end{cases} \tag{6-19}$$

Let us assume that the Granular TS fuzzy model has been already constructed. The quality of this model depends on how well the information granules $\boldsymbol{G}(\hat{y}_k)$ of the predicted output "cover" the original output values. The fundamental with this regard is the notion of coverage of the information granule and its quantification. The following is the coverage index used for evaluating the granular TS model,

$$\kappa = \frac{\sum_{k=1}^{M} \text{incl}\left(y_k, \boldsymbol{G}(\hat{y}_k)\right)}{M} \tag{6-20}$$

where $\text{incl}\left(y_k, \boldsymbol{G}(\hat{y}_k)\right)$ is a measure of inclusion of target y_k in the granular counterpart produced by granular prototypes $\boldsymbol{G}(\hat{y}_k)$. The summation in the formula is done over all elements.

Performance index area under curve (AUC) computed in the following way

$$\text{AUC} = \frac{1}{\varepsilon_{\max}} \int_0^{\varepsilon_{\max}} \text{coverage}(\varepsilon)\, d\varepsilon \tag{6-21}$$

where $\varepsilon_{\max}$ is the maximal value of the level of information granularity at which the granular prototypes do not intersect.

6.6 Experimental studies

In the series of experiments we use some synthetic data and a number of publicly available datasets for which fuzzy models were constructed in the past. Table 6-1 provides a summary of these data. In each case the number of rules (local models) was selected on a basis of the modeling results (quantified in terms of the RMSE values) obtained for the successively increasing number of clusters (rules). A "standard" way of forming the TS model (as briefly outlined in Section 6.2) was used. The development of the fuzzy models is realized by randomly splitting the corresponding data into a training (60% of data) and testing (40%) subsets. In the realization of the protocol P_3, the optimization of the vectors of the levels of asymmetry γ was done with the use of the generic version of the particle swarm optimization algorithm; see (Eberhart and Shi 2001). The setup of the PSO was the following: the values of the inertia weight,

w, were linearly from 0 to 1 over the course of optimization. The value of cognitive factor, c_1 and social factor c_2 were both set to 1.5.

Table 6-1: A summary of data sets used in the experiments. Here shown is also the number of clusters (rules) used in the design of the RS fuzzy model.

No	*Data set*	*Abbreviation*	*Number of*	*Number of*	*Number of*
1	One Dimensional Sine Function	Data 1	1	401	8
2	Static Function Approximation	Data 2	2	150	10
3	Sine Function	Data 3	2	300	9
4	Air Pollution PM10	PM10	7	500	5
5	Boston Housing	Housing	13	506	10
6	Body Fat	Body Fat	14	252	10
7	Parkinson Tele-monitoring	Parkinson	17	5875	6
8	Voltage Estimation	Voltage	4	1056	9
9	Auto-MPG	MPG	7	252	7
10	Computer Activity	Computer	21	8192	6

One-dimensional synthetic data

For illustrative purposes, we start with the one-dimensional data set generated by a nonlinear function sin(5x)/x defined over [1,5], see Figure 6-5.

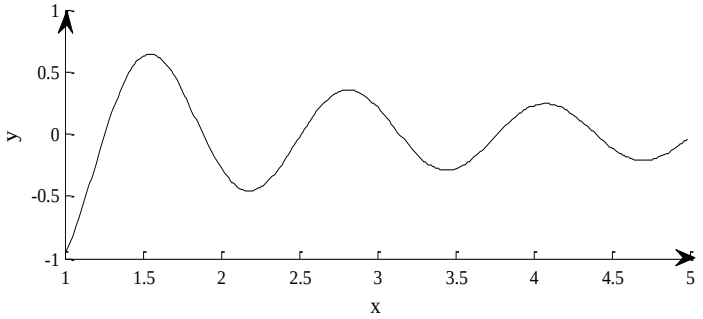
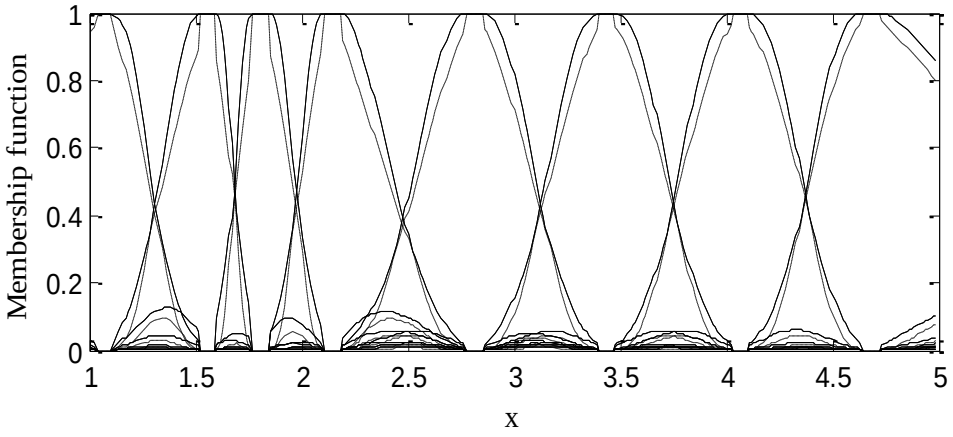


Figure 6-5: Plot of a one-dimensional function

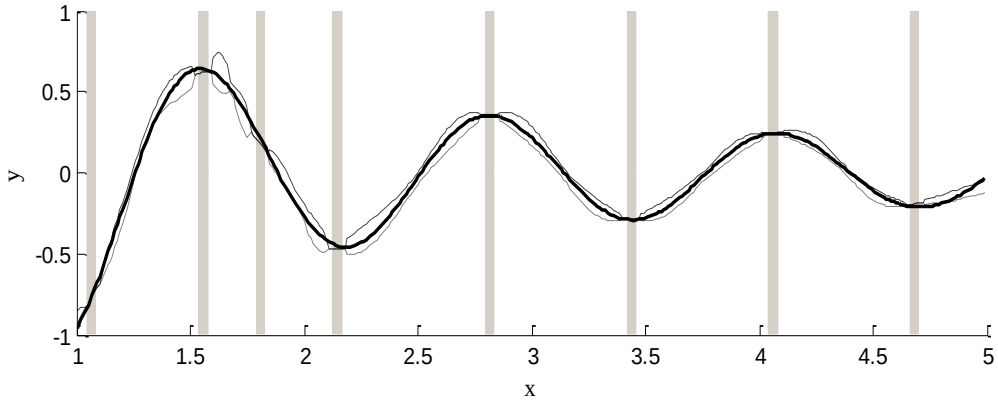
We randomly generate 200 input-output data and use them in the construction of the rule-based model. The number of rules was set to 8 and the rules are as follows

R_1: If X is A_1 then y= 0.03x+0.28
R_2: If X is A_2 then y= 2.53x-3.47
R_3: If X is A_3 then y= -1.99x+3.77
R_4: If X is A_4 then y= 0.14x-0.34
R_5: If X is A_5 then y=-0.11x-0.22
R_6: If X is A_6 then y= 0.52x-0.19
R_7: If X is A_7 then y= 0.11x-0.70
R_8: If X is A_8 then y= 0.03x-0.39

Proceeding with the first protocol, the obtained granular prototypes are shown in Figure 6-6 (a) while the granular outputs are included in Figure 6-6 (b). The results are reported for two levels of information granularity.

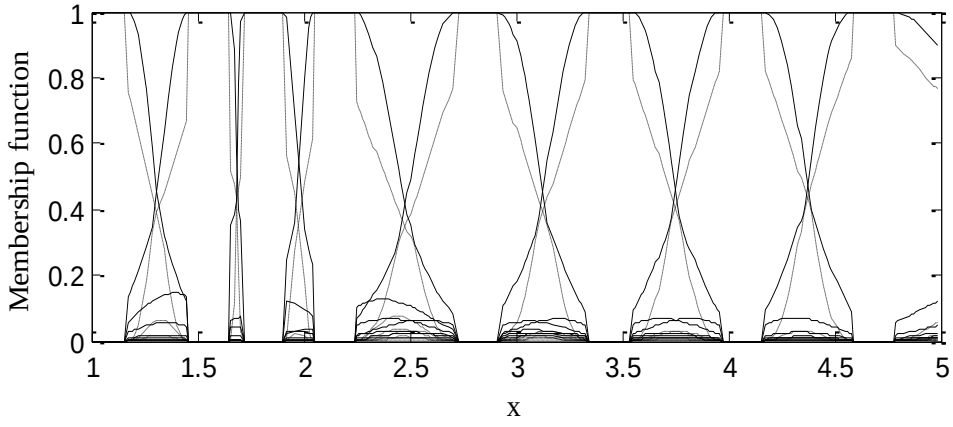


(a)

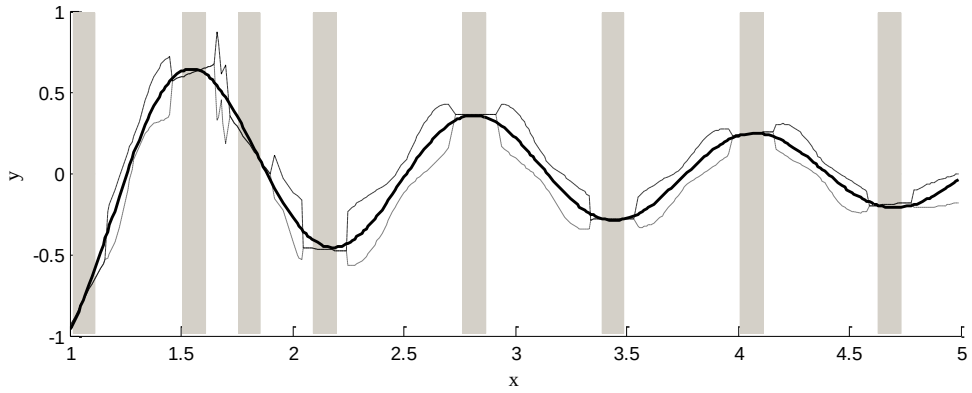


(b)

Figure 6-6: Granular membership functions (a) and granular outputs (b); ε =0.02. The granular prototypes are shown as shaded regions in the input variable



(a)



(b)

Figure 6-7: Granular membership functions (a) and granular outputs (b); ε =0.05. The granular prototypes are shown as shaded regions in the input variable

Figure 6-8 summarizes the results of optimization by showing the coverage values obtained for the testing and training data.

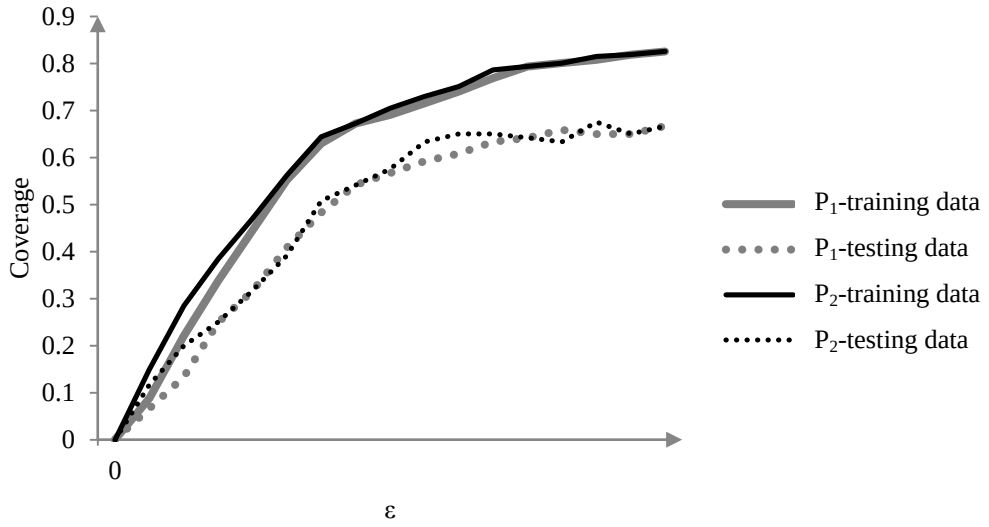


Figure 6-8: The plot of the coverage as a function of ε. The solid line – training data; dotted line – testing data

Figure 6-9 visualizes the optimal values of g when using protocol- P_2 and for selected values of ε. It is noticeable that most of these optimal values fluctuate around 0.5.

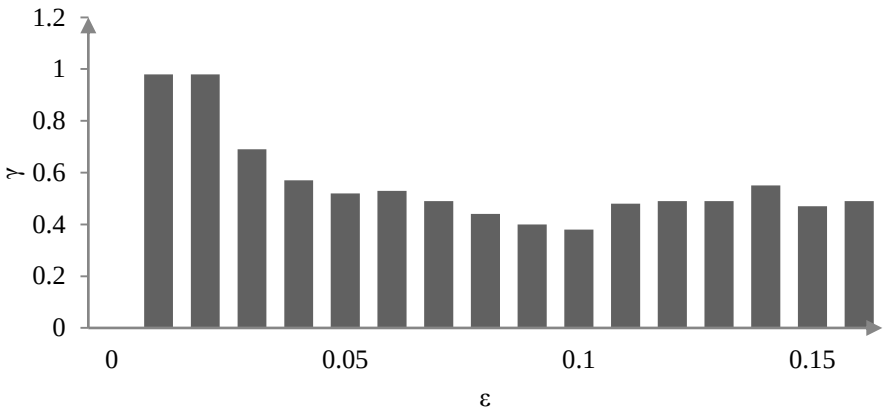


Figure 6-9: Optimal values of γ for the corresponding values of ε (protocol P_2)

Two-dimensional synthetic data

The two-dimensional data set generated by a nonlinear function $y= \sin(x_1)/x_1 + \sin(x_2)/x_2$ the data is composed of 200 data points, arranged in a regular grid within the [-10:10] x [-10:10] see Figure 6-10.

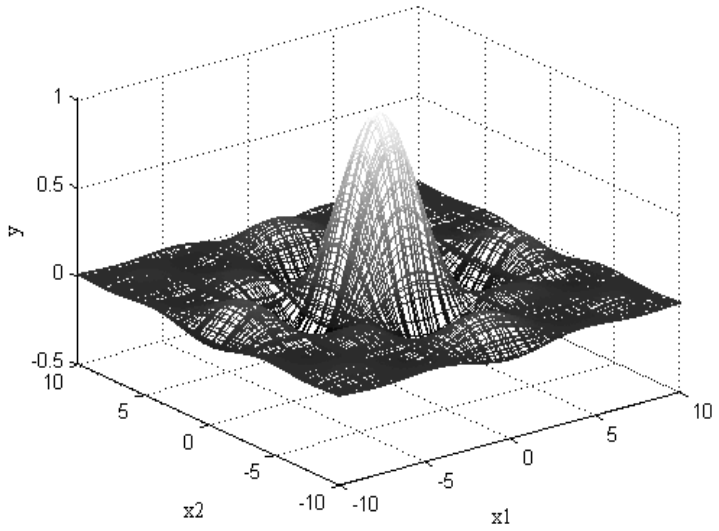


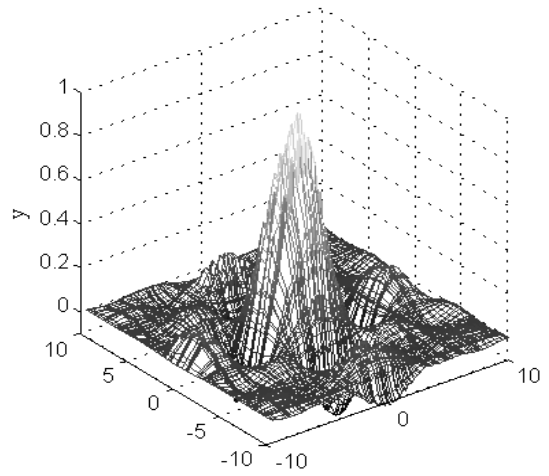
Figure 6-10: Plot of a two-dimensional function

The number of rules was set to 9 and the rules are as follows

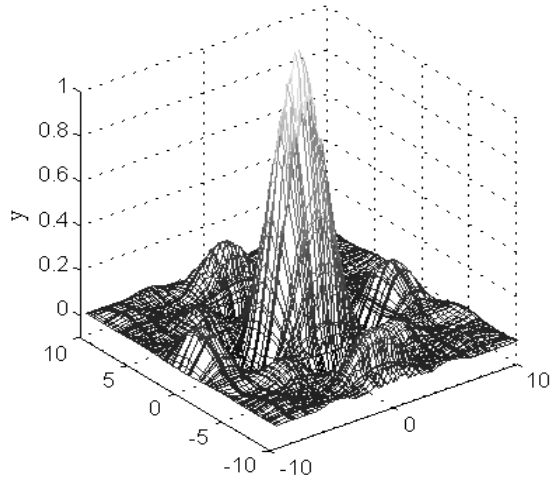
R_1: If X is A_1 then $y= 0.0065x_2+0.0202x_1-0.1811$
R_2: If X is A_2 then $y= -0.3168x_2-0.2116x_1+1.2330$
R_3: If X is A_3 then $y= 0.2127x_2+0.2294x_1+1.1541$
R_4: If X is A_4 then $y= -0.0094x_2+0.0168x_1+0.4576$
R_5: If X is A_5 then $y= 0.0492x_2+0.1086x_1+0.4576$
R_6: If X is A_6 then $y= -0.0889x_2+0.0370x_1-0.1241$

R_7: If X is A_7 then y= $-0.0212x_2+0.0178x_1-0.0089$
R_8: If X is A_8 then y= $-0.0039x_2+0.0510x_1-0.1441$
R_9: If X is A_9 then y= $-0.0301x_2+0.0365x_1-0.0652$

Proceeding with the first protocol, the obtained granular prototypes are shown in Figure 6-11. The results are reported for two levels of information granularity. Just looking at more detailed picture, we show the coverage treated as a function ε, Figure 6-12.



(a)



(b)

Figure 6-11: Granular outputs for using Protocol-P_1 for ε =0.01; (a) minimum output and (b) maximum output.

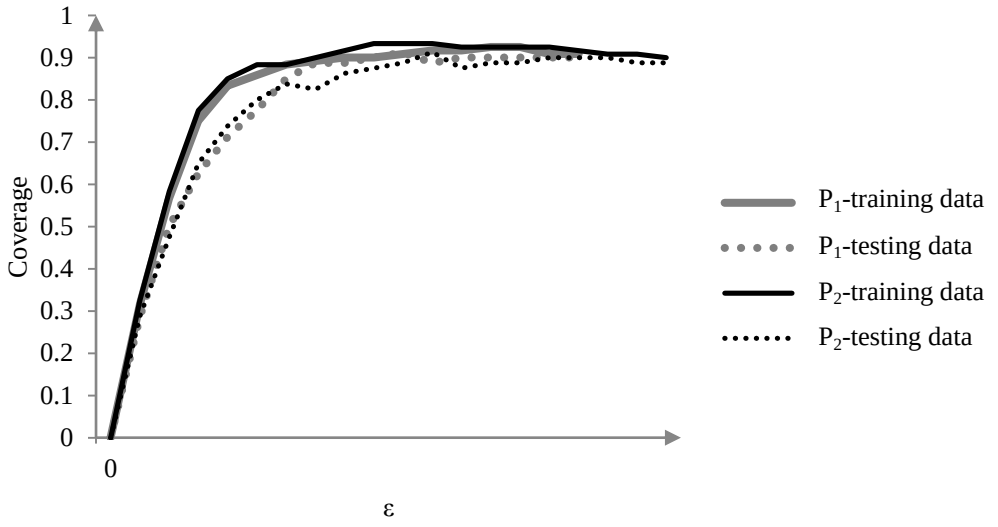


Figure 6-12: The plot of the coverage as a function of ε. The solid line – training data; dotted line – testing data

Figure 6-13 visualizes the optimal values of γ when using protocol- P_2 and for selected values of ε.

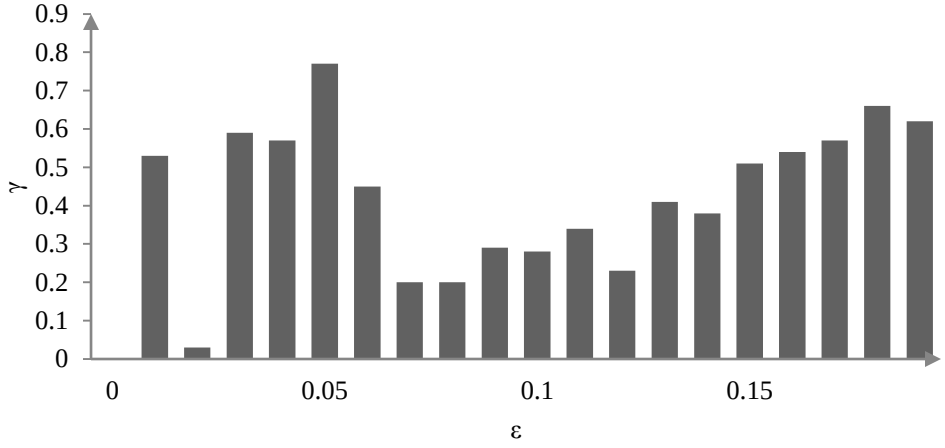
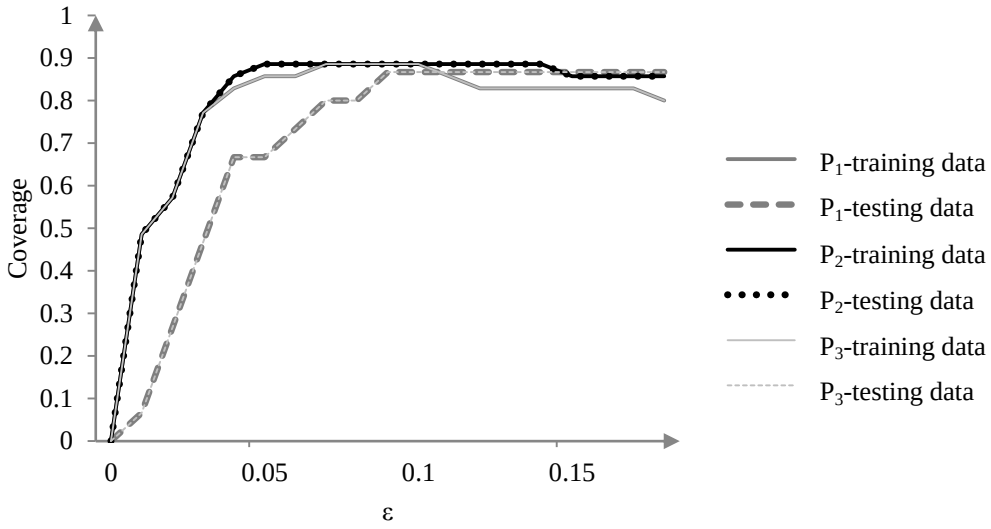


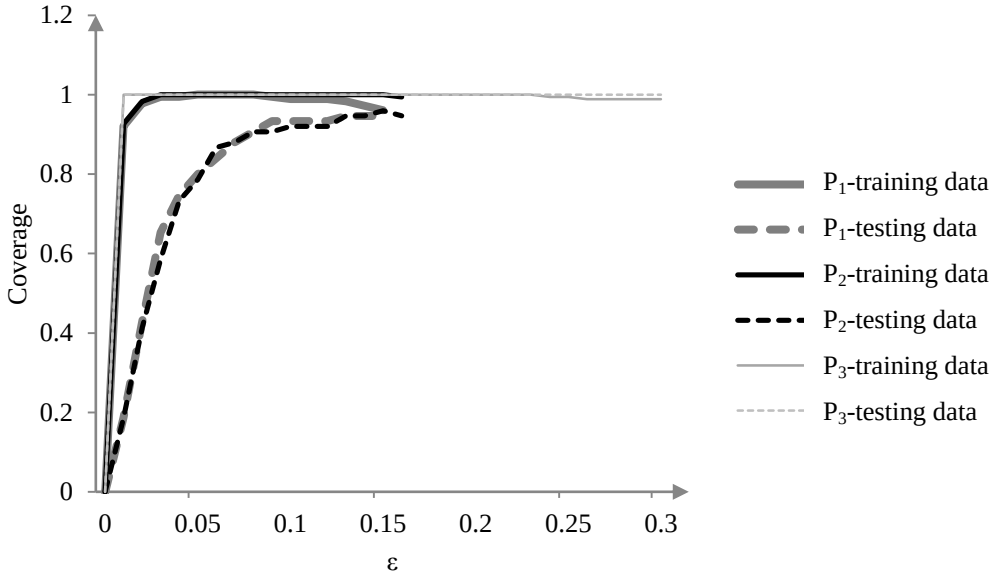
Figure 6-13: Optimal values of γ for the corresponding values of ε (protocol P_2)

Multivariable input data

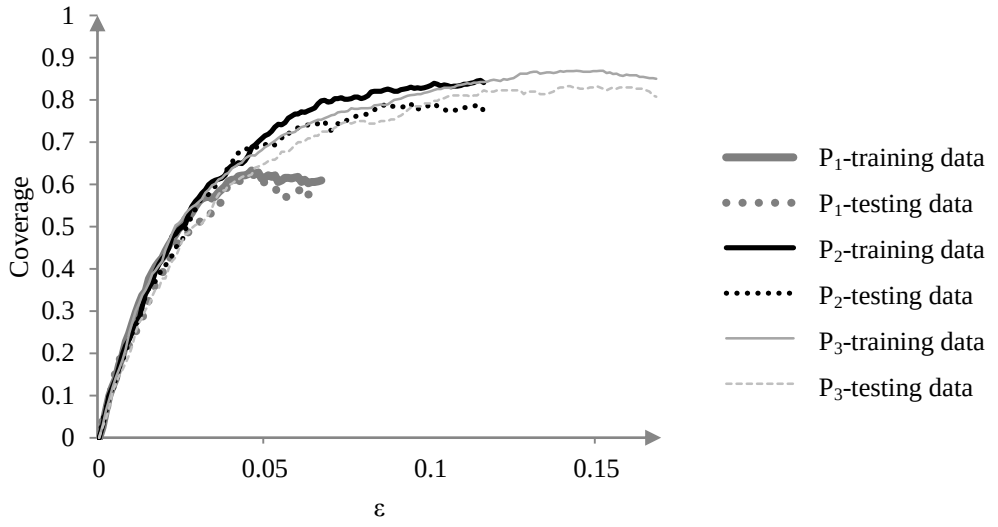
For the already constructed fuzzy models we developed its granular generalizations using protocols P_1-P_3. The results shown in Figure 6-14 visualize the coverage as a function of ε.


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Coverage
0
0.05
0.1
0.15
ε
P1-training data
P1-testing data
P2-training data
P2-testing data
P3-training data
P3-testing data

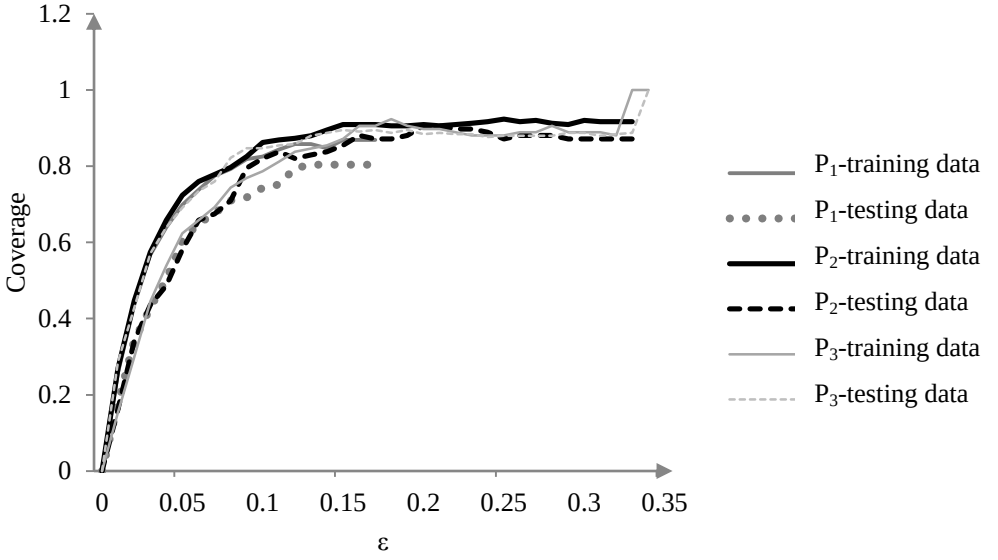
(a)


1.2
1
0.8
0.6
0.4
0.2
0
Coverage
0
0.05
0.1
0.15
0.2
0.25
0.3
ε
P1-training data
P1-testing data
P2-training data
P2-testing data
P3-training data
P3-testing data

(b)



(c)



(d)

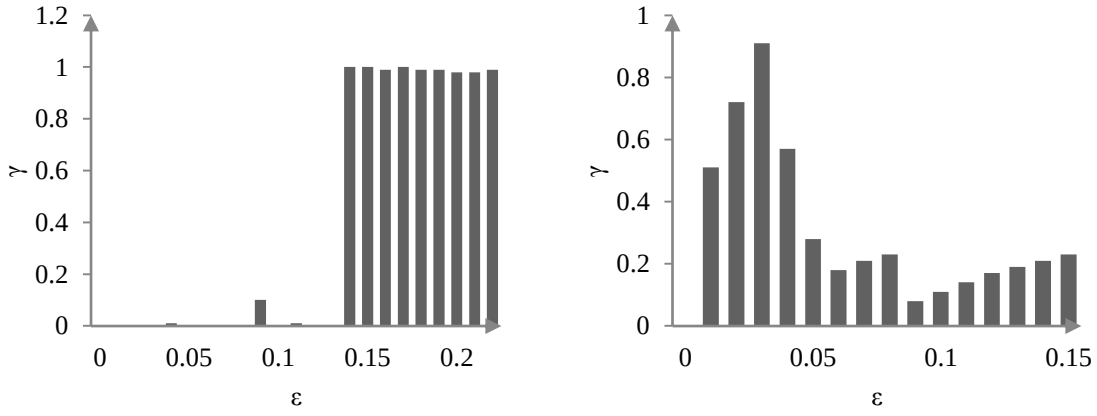
Figure 6-14: Plots of coverage versus ε for different data sets. The solid line – training data; dotted line – testing data ; (a) data 3, (b) Body Fat, (c) Voltage, and (d) Auto-MPG

We ran the three information granularity allocation. The main results are summarized in Table 6-2. There are significant improvement when using Protocol-P_2 and Protocol-P_3 when compared the obtained results to the results produced by Protocol-P_1. This is not surprising as these protocols offer a significant level of flexibility when allocating the information granularity. In addition, the maximum values of ε are also increased when using the other two protocols.

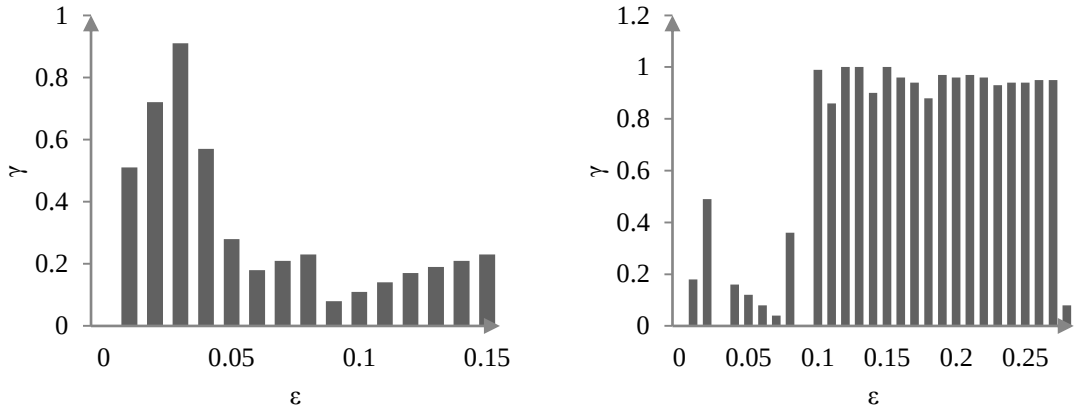
Table 6-2: Values of AUC obtained for protocols P_1-P_3

	AUC- protocol P_1			AUC- protocol P2			AUC- protocol P3		
Data set	Training	Testing	Max ε	Training	Testing	Max ε	Training	Testing	Max ε
Data 2	0.023	0.011	0.040	0.039	0.016	0.070	0.061	0.034	0.100
PM10	0.091	0.079	0.340	0.119	0.110	0.340	0.324	0.325	0.340
Housing	0.149	0.120	0.220	0.153	0.130	0.230	0.206	0.197	0.230
Body Fat	0.148	0.120	0.150	0.159	0.128	0.160	0.300	0.300	0.310
Parkinson	0.019	0.014	0.070	0.040	0.037	0.120	0.054	0.049	0.170
Voltage	0.033	0.031	0.067	0.076	0.073	0.117	0.120	0.110	0.160
Auto	0.123	0.107	0.170	0.276	0.257	0.340	0.279	0.271	0.340

While the use of the asymmetry index impact positively the performance of the granular model, it is helpful to see how the values of g are distributed, see Figure 6-15. Here, the value of optimal gamma is different for particular value of ε. In most cases, the value of optimal γ is different for each value of ε. Therefore, by using Protocol-P_2, we can find the optimal allocation of information granulation for the granular model that can improve the coverage value for the model.

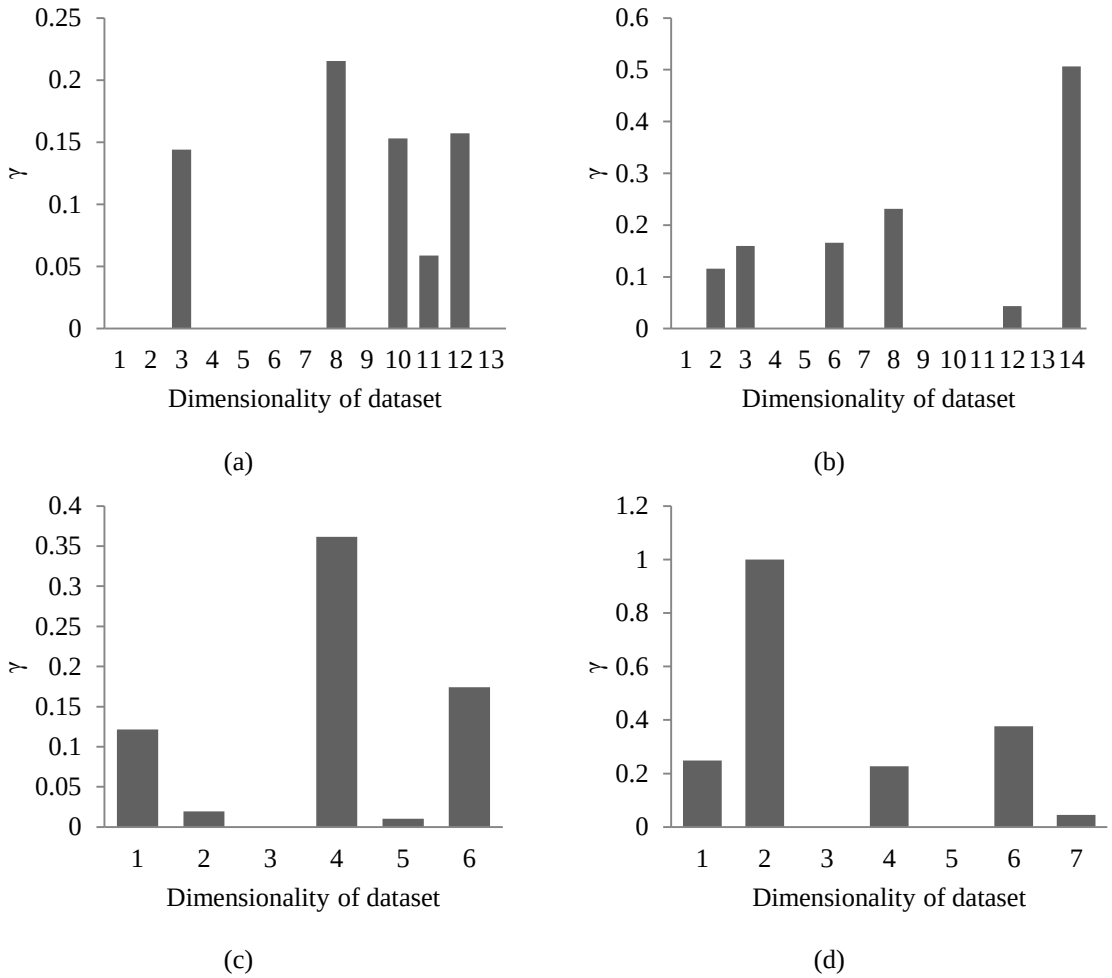


(a) (b)



(c) (d)

Figure 6-15: Plots of asymmetry values (gamma) versus ε using Protocol –P_2; (a) housing data set, (b) body fat data set (c) auto-MPG data set, and (d) PM10 data set.



(a) (b)

(c) (d)

Figure 6-16: Plots of asymmetry values (gamma) using Protocol- P_3; (a) housing dataset, (b) body fat data set (c) auto-MPG data set, and (d) PM10 dataset.

Yet another view at the results of the granular model can be obtained by plotting the results of the granular model vis-à-vis the data (both training and testing), see Figure 6-17 to 6-20. These figures help not only visualize the coverage the data by the intervals but also show the length of the individual intervals.

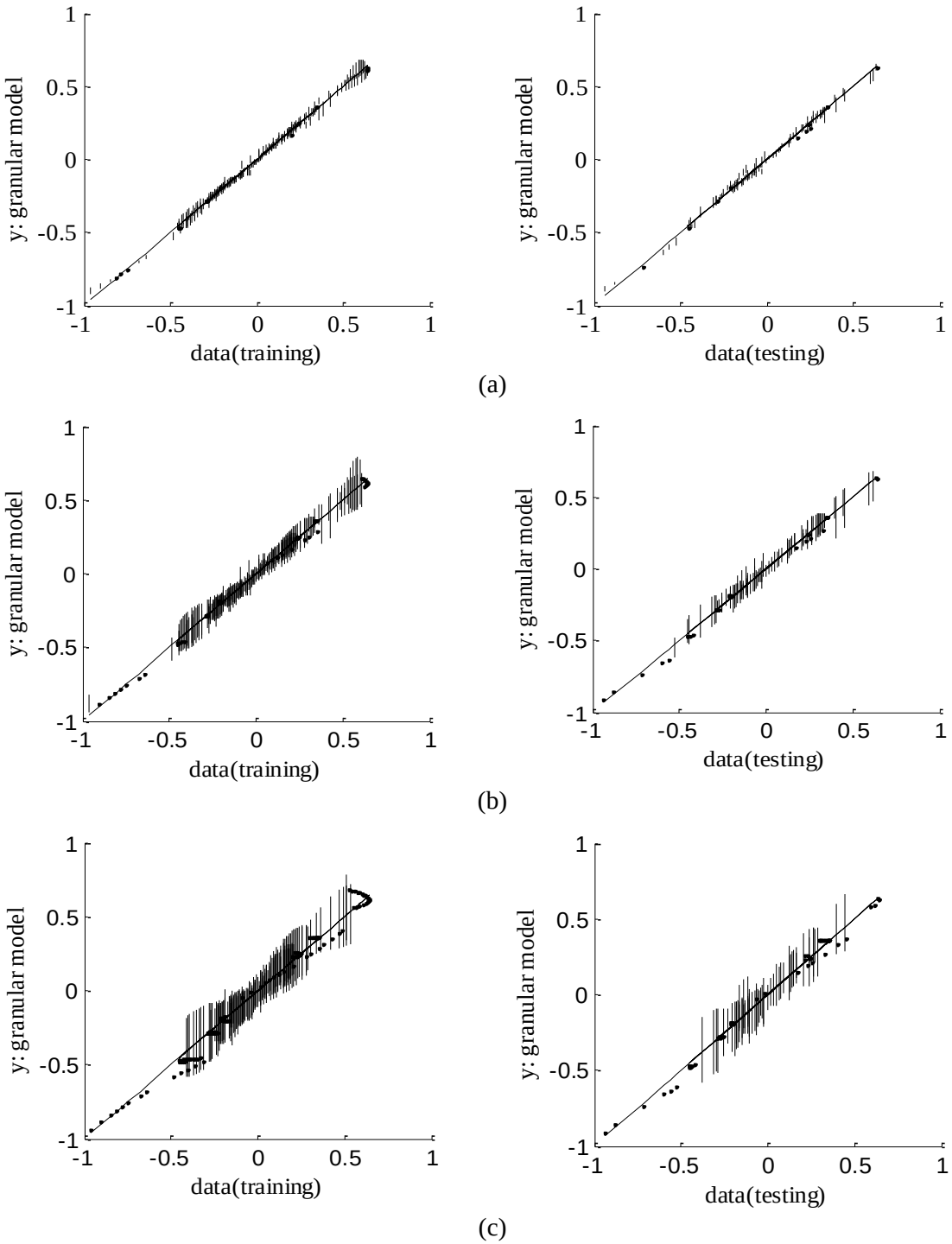
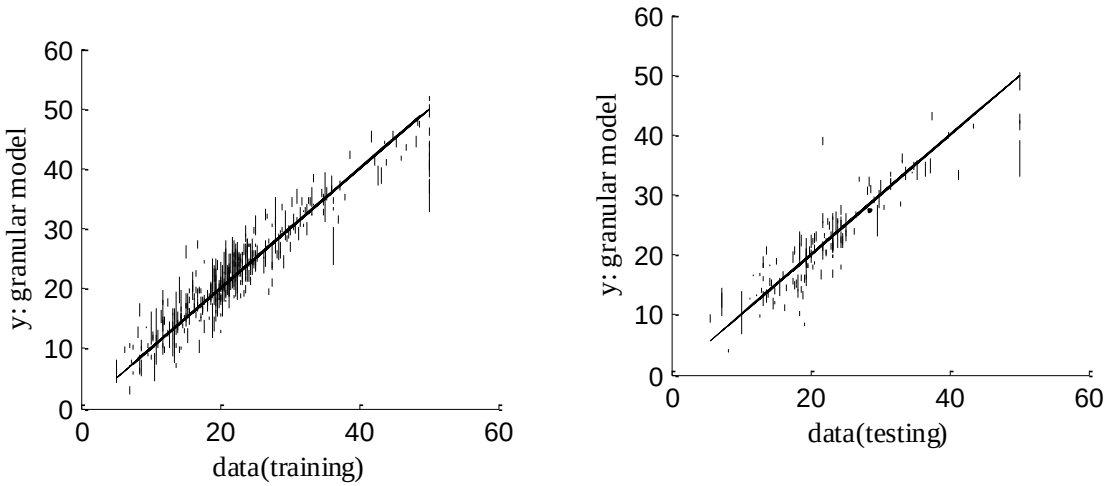
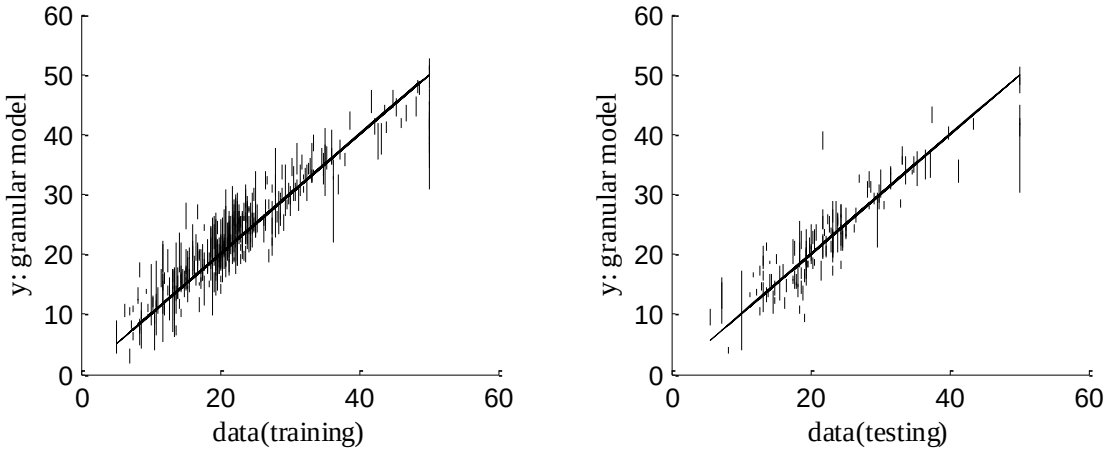


Figure 6-17: The plot of interval/granular output versus numeric output for Data 1 by using Protocol-P_1; (a) ε=0.01, (b) ε=0.03, and (c) ε=0.05.

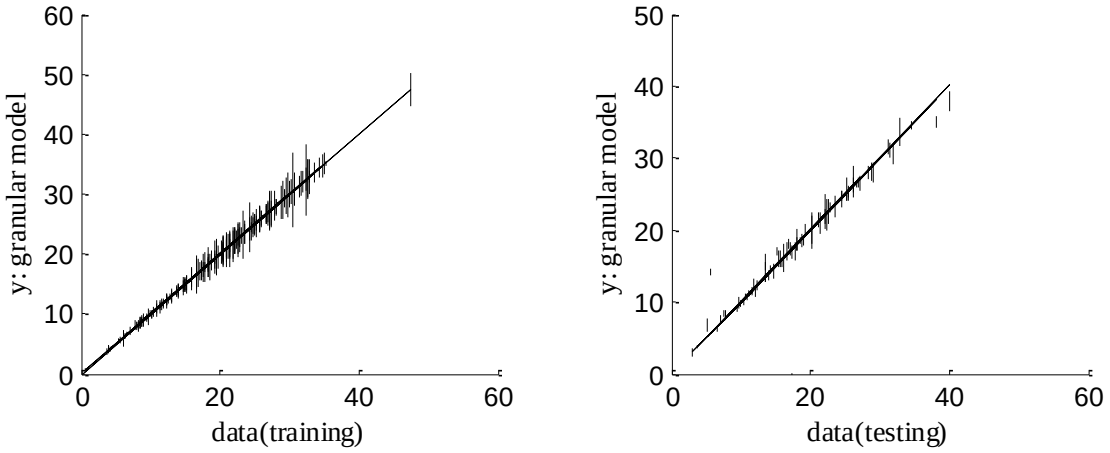


(a)

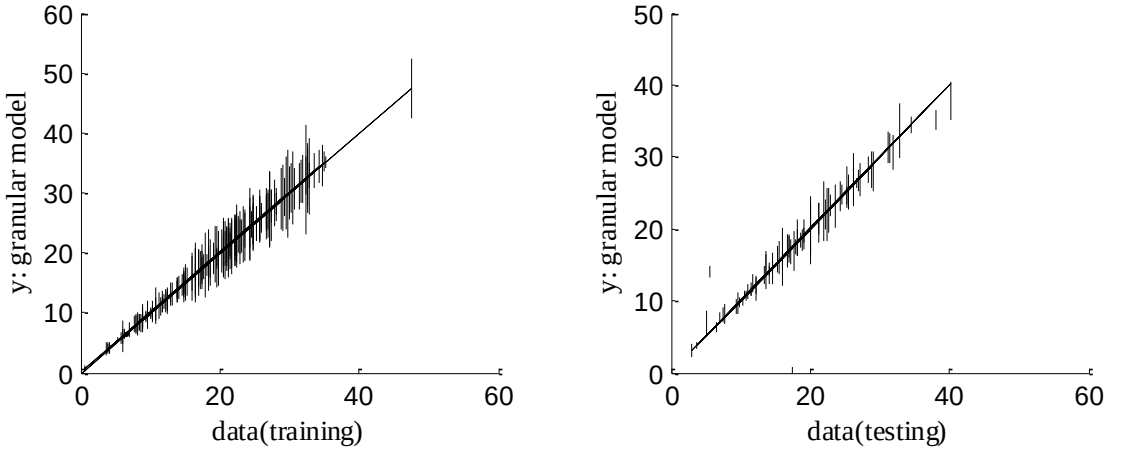


(b)

Figure 6-18: The plot of interval/granular output versus numeric output for the Housing data set by using Protocol-P_1; (a) ε=0.05 and (b) ε=0.1.

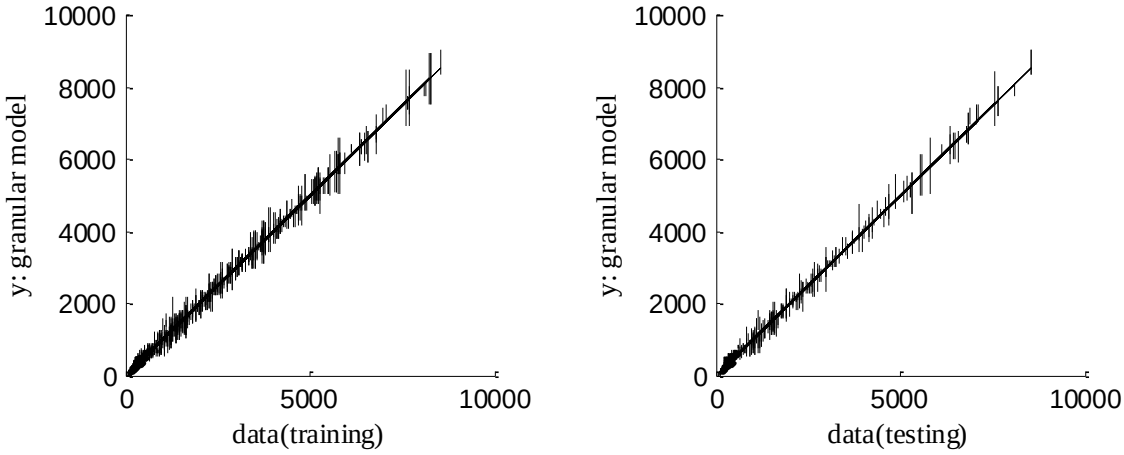


(a)

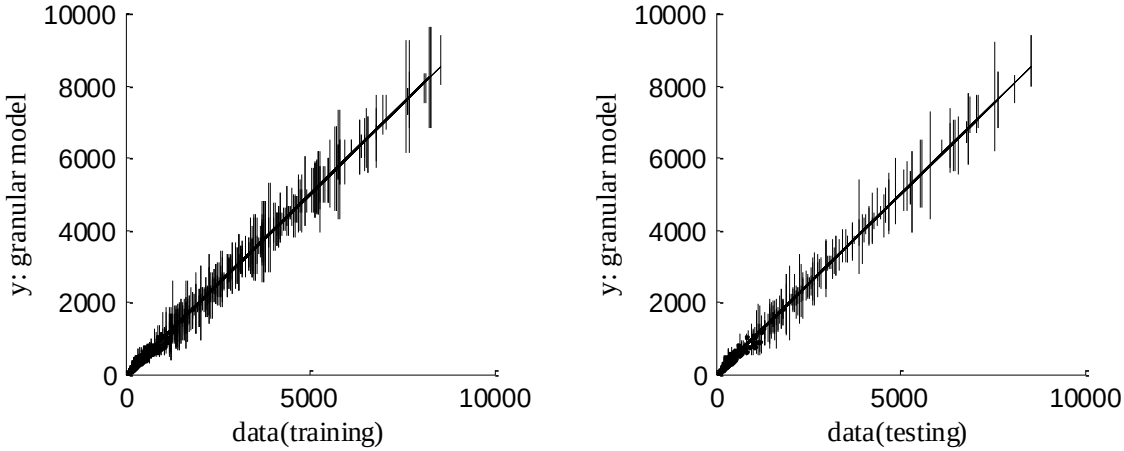


(b)

Figure 6-19: The plot of interval/granular output versus numeric output for Body Fat data set by using Protocol-P_1; (a) ε=0.05 and (b) ε=0.1



(a)



(b)

Figure 6-20: The plot of interval/granular output versus numeric output for Voltage data set by using Protocol-P_1; (a) ε=0.05 and (b) ε=0.1

To quantify the distribution of information granularity in the output of the granular model, we can display a relationship between the distance of a given **x** from the nearest prototype $\mathbf{v}_i$ and the length of the information granule produced by the model, namely $\rho = f(\min_{i=1,2,\ldots,c} \|\mathbf{x}_k - \mathbf{v}_i\|^2)$ where ρ is a length of the output of the granular model. The distance function used here is the same one as used in the clustering. The plots of this relationship for the experimental data are presented in Figure 6-21 and 6-22.

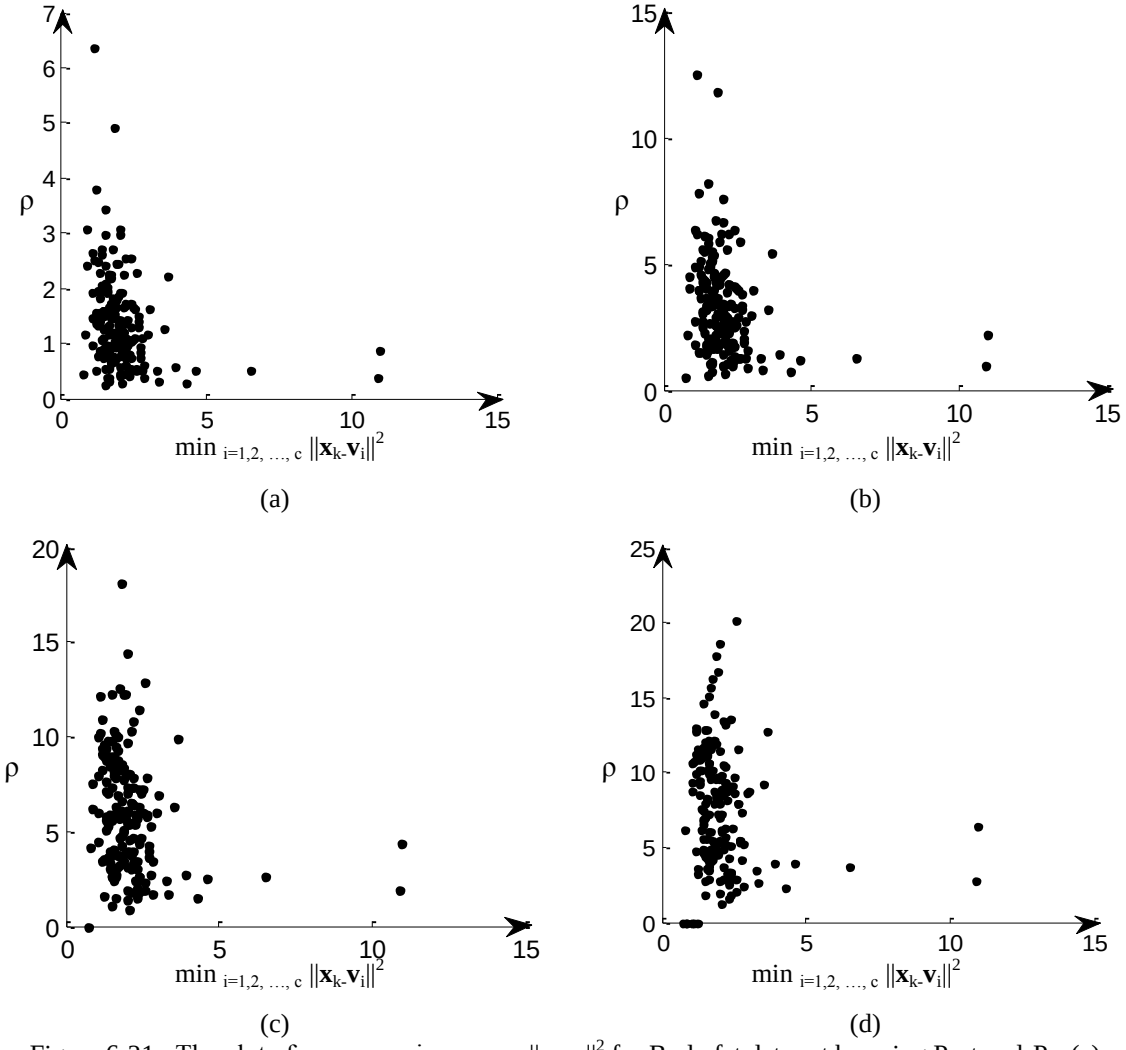


Figure 6-21: The plot of ρ versus $\min_{i=1,2,\ldots,c} \|\mathbf{x}_k - \mathbf{v}_i\|^2$ for Body fat data set by using Protocol-P_1; (a) ε=0.02, (b) ε=0.05, (c) ε=0.1, and (d) ε=0.15

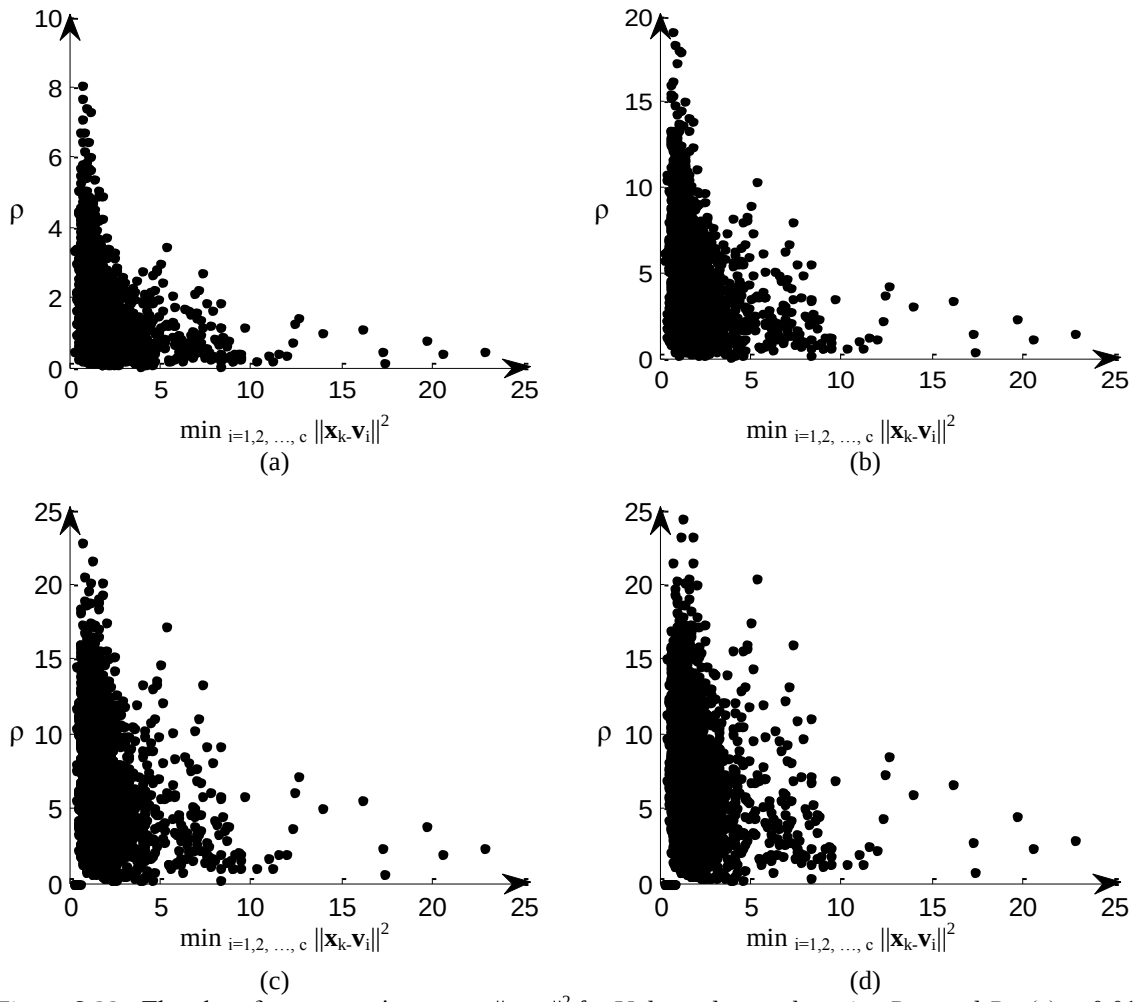


Figure 6-22: The plot of ρ versus min $_{i=1,2,\ldots,c}$ $\|\mathbf{x}_k-\mathbf{v}_i\|^2$ for Voltage data set by using Protocol-P_1; (a) ε=0.01, (b) ε=0.03, (c) ε=0.05, and (d) ε=0.06

6.7 Conclusions

In this chapter we present the development of the granular TS fuzzy model. In this approach we emphasized that information granularity plays an important role in the construction of granular TS model. The increased abstraction of construct is inherently associated with and quantified by granular fuzzy sets. It is shown that the protocols of allocation of information granulation form an effective design framework of rule-based system. The experimental validation was then carried out by implementing our proposed method to the standard TS fuzzy model. The experimental results have revealed that, granular TS model works effectively with all the data set.

7.Conclusion and Future Work

Data-driven fuzzy modeling has been used in various application domains. The advantages of fuzzy modeling include the capability to use the knowledge representation in the form of if-then rules. This procedure is similar to human reasoning in linguistic terms. In addition, the fuzzy model also can approximate complex non-liner problem by using a simple model. The construction of a fuzzy model has two principal aims: (1) to achieve a good approximation for the problem based on the accuracy of the resulting model and (2) to reduce the complexity of the fuzzy rules by reducing the total number of rules.

The ultimate challenge of data-driven fuzzy system modeling is to construct accurate and transparent models. The difficulty occurs because of the need to achieve these contradictory aims at the same time. In order to get a good approximation for a problem, we need to use more rules to represent the antecedent and the consequent part of the problem. In contrast, to obtain a comprehensible and interpretable model, we have to use a smaller number of rules to represent the problem. There is no easy way to achieve a balance between both aims. Most of the research proposes methods focusing only on the best accuracy and neglects the other component. Therefore, complexity reduction is becoming a pertinent research topic in the field of fuzzy rule-based systems.

In this thesis we proposed simple framework for constructing fuzzy modeling from high dimensional and large data. We focus on the complexity reduction by using the integration of feature and data reduction in the construction of the fuzzy models. We introduced a method for searching the subset of data based on Cooperative Particle Swarm Optimization. A cooperative PSO was developed in order to overcome the limitation of using standard PSO when dealing with a high dimensional search space. The best selected subset of data obtained with this framework is capable of representing the original large data set. The size of the selected features and data used to construct the fuzzy model can be adjusted based upon the feedback provided in terms of the performance of the model constructed.

Next, we introduced a granular fuzzy rule-based model, which results from a direct result compactification of the rule base, with the intent of arriving at a more compact, interpretable yet highly representative collection of rules. In this study, we emphasized that information granularity plays an important role in the reduction of rule-based systems. We showed that the protocols of the allocation of information granularity establish an effective design framework for the granular rule-based systems. Using the Cooperative PSO, we generated two optimization problem solutions: the rule selection and the optimal allocation of information granularity.

This thesis also presented a novel method for constructing the *Takagi-Sugeno* fuzzy model, based on the concept of information granulation. The motivation to construct this framework came from the granular fuzzy rules-based system, where the granular representation of the fuzzy rules improves the generality of the existing fuzzy rules-based system. In this study, we implemented an alternative concept to formulate the fuzzy model, called the Granular Takagi-Sugeno model. The granular model is more generalized compared to the existing TS model. The construction of the granular TS model began with formulating the granular prototype values with the corresponding granular membership grades. Several protocols were used to construct the representative of the antecedent part. Then predicted output was calculated based on the given antecedent part represented by the granular membership grades. Finally, the coverage of the predicted output was used as the performance index for evaluating the granular TS model.

By improving the fuzzy model, we achieved several significant research objectives:

1. Exploration of the use of population-based methods in solving several optimization problems.
2. Analysis of the use of simultaneous feature and data selection in constructing the fuzzy model.
3. Investigation of the concept of granular computing in the application to fuzzy models
4. Exploration of the optimal allocation of information granularity by using several protocols.
5. Investigation of the construction of the granular representation of the *Takagi-Sugeno* fuzzy model.
6. Evaluation of the performance of the suggested frameworks by using a real-world data set.

This research has contributed several improvements to the construction of fuzzy modeling. However, further issues are still worth investigating in the pursuit of an ideal granular fuzzy system framework. The further investigation and development of an efficient framework for the optimal allocation of information granularity are necessary in order to improve the quality of the granular representation. An alternative optimization algorithm could be used to optimize the parameters associated with the allocation of information granularity. In addition, researches could focus on the implementation of an extensible software application that includes all the proposed frameworks. The application tool has the possibility for further extensions that include new improvements to the frameworks for solving complex problems in the real world.

References

Abonyi, J. *Fuzzy Model Identification for Control.* Boston: Springer-Verlag, 2003.

Alcala, R., J. Alcala-Fdez, J. Casillas, O. Cordon, and F. Herrera. "Hybrid learning models to get the interpretability–accuracy trade-off in fuzzy modeling." *Soft Computing* 10, no. 9 (2006): 717-734.

Alcala, R., P. Ducange, F. Herrera, and B. Lazzerini. "A multiobjective evolutionary to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems." *IEEE Trans. on Fuzzy Systems* 17, no. 5 (2009): 1106-1122.

Alcala-Fdez, J., and F. Herrera R. Alcala. "A fuzzy associative classification system with genetic rule selection for high-dimensional problems." *4th Int. Workshop on Genetic and Evolutionary Fuzzy Systems.* Mieres, 2010. 33-38.

Arturo, J.,Olvera-Lopez, J., Ariel, Carrasco-Ochoa, Francisco,Martinez-Trinidad J, and J. Kittler. "A review of instances selection methods." *Artificial Intelligence Review* 34, no. 2 (2010): 133-143.

Baranyi, P., and Y. Yam. "Fuzzy rule case reduction." In *Fuzzy IF-THEN rules in computational intelligence: theory and applications*, by D. Ruan and E.E. Kerre (eds), 135-160. Kluwer Academic Publishers, 2000.

Bargiela, A., and W. Pedrycz. *Granular Computing: An Introduction.* Boston: Kluwer Academic Publishers, 2003.

Berlanga, F.J., A.J. Rivera, M.J. Del Jesus, and F. Herrera. "GP-COACH: Genetic programming-based learning of compact and accurate fuzzy rule-based classification systems for high-dimensional problems." *Information Sciences* 180, no. 8 (2010): 1183-1200.

Bezdek, J.C. *Pattern recognition with fuzzy objective function algorithms.* New York: Plenum Press, 1981.

Bezdek, J.C., R. Ehrlich, and W. Full. "FCM: The fuzzy c-means clustering algorithm." *Computers & Geosciences* 10, no. 2-3 (1984): 191-203.

Blum, A.L., and P. Langley. "Selection of relevent features and examples in machine learning." *Artificial Intelligence* 97, no. 1-2 (1997): 245-271.

Cano, J. R., F. Herrera, and M. Lozano. "Using evolutionary algorithms as instance selection for data reduction in kdd: an experimental study." *IEEE Trans. on Evolutionary Computation* 7, no. 6 (2003): 561-515.

Castellano, G., C. Castiello, A. M. Fanelli, and C. Mencar. "Knowledge discovery by a neuro-fuzzy modeling framework." *Fuzzy Sets and Systems* 149, no. 1 (2005): 187-207.

Chen, J.Q., Y.G. Xi, and Z.J. Zhang. "A clustering algorithm for fuzzy model identification." *Fuzzy Sets and Systems* 98, no. 3 (1998): 319-329.

Chen, M. Y., and D. A. Linkens. "Rule-base self-generation and simplification for data-driven fuzzy models." *Fuzzy Sets and Systems* 142, no. 2 (2004): 243-265.

Chen, Y., B. Yang, A. Abraham, and P. Lizhi. "Automatic design of hierarchical Takagi–Sugeno type fuzzy systems using evolutionary algorithms." *IEEE Trans. on Fuzzy Systems* 15, no. 3 (2007): 385-397.

Chui, S.L. "Fuzzy model identification based on cluster estimation." *Journal of Intelligent & Fuzzy Systems* 2 (1994): 267-278.

Chui, S.L. "Selecting input variables for fuzzy models." *Journal of Intelligent & Fuzzy Systems* 4, no. 4 (1996): 243-256.

Cover, T., and P. Hart. "Nearest neighbor pattern classification." *IEEE Trans. on Information Theory*, no. 1 (1967): 21-27.

Delgado, M.R., F. V. Zuben, and F. Gomide. "Coevolutionary genetic fuzzy systems: a hierarchical collaborative approach." *Fuzzy Sets and Systems* 141, no. 1 (2004): 89-106.

Derrac, J., S. Garcia, and F. Herrera. "IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule." *Pattern Recognition* 43, no. 6 (2009): 2082-2105.

Du, H., and N. Zhang. "Application of evolving Takagi–Sugeno fuzzy model to nonlinear system identification." *Applied Soft Computing* 8, no. 1 (2008): 676-686.

Dy, J.G., and C.E. Brodley. "Feature selection for unsupervised learning." *Journal of Machine Learning Research* 5 (2004): 845-889.

Eberhart, R. C., and Y. Shi. "Particle Swarm Optimization: Developments, Applications and Resources." *IEEE Congress on Evolutionary Computation.* Seoul, 2001. 81-86.

Eiben, A. E., and J. E. Smith. *Introduction to Evolutionary Computing.* New York: Springer-Verlag, 2003.

El-Abd, M. Kamel, M. "Cooperative particle swarm optimiers: a powerful and promising approach." *Studies in Computational Intelligence* 31 (2006): 239-259.

Emami, M.R., I.B. Turken, and A.A. Goldenberg. "Development of a systematic methodology of fuzzy modeling." *IEEE Trans. on Fuzzy Systems* 6, no. 3 (1998): 346-361.

Enzhe, Y., and C. Sungzoon. "Ensemble based on GA wrapper feature selection." *Computers & Industrial Engineering* 51, no. 1 (2006): 111-116.

Gaweda, A. E., J. M. Zurada, and R. Setiono. "Input selection in data-driven fuzzy modeling." *IEEE Int. Conf. on Fuzzy Systems.* Melbourne, 2001. 1251-1254.

Ghazavi, S. N., and T. W. Liao. "Medical data mining by fuzzy modeling with selected features." *Artificial Intelligence in Medicine* 43, no. 3 (2008): 195-206.

Gomez-Skarmeta, A.F., and M. Vila, M.A. Delgado. "About the use of fuzzy clustering techniques for fuzzy model identification." *Fuzzy Sets and Systems* 106, no. 2 (1999): 179-188.

Grzegorzewski, P. "Distances between intuitionistic fuzzysets and/or interval-valued fuzzysets based on the Hausdorff metric." *Fuzzy Sets and Systems* 148, no. 2 (2004): 319-328.

Guillaume, S. "Designing fuzzy inference systems form data: an interpretability-oriented review." *IEEE Trans. on Fuzzy Systems* 9, no. 3 (2001): 426-443.

Guyon, I., and A. Elisseeff. "An introduction to variable and feature selection." *Journal of Machine Learning Research* 3 (2003): 1157-1182.

Hadjili, M. L., and V. Wertz. "Takagi-Sugeno fuzzy modeling incorporating input variables selection." *IEEE Trans. on Fuzzy Systems* 10, no. 6 (2002): 728-742.

Hall, L. O., I. B. Ozyurt, and J.C.Bezdek. "Clustering with a genetically optimized approach." *IEEE Trans. on Evolutionary Comp.* 3, no. 2 (1999): 103-112.

Haupt, R.L., and S. E. Haupt. *Practical Genetic Algorithm.* Hoboken: John Wiley & Sons, 2004.

Herrera, F. Lozano, M. Verdegay, J.L. "Tackling real-coded genetic algorithms: operator and tool for behavioural analysis." *Artificial Intelligence Review*, 1998: 265-319.

Hertz, A., and D. Kobler. "A framework for the description of evolutionary algorithms." *European Journal of Operational Research* 126 (2000): 1-12.

Hu, X., Y. Shi, and R.C. Eberhart. "Recent advances in particle swarm optimization." *Congress on Evolutionary Computation.* 2004. 90-97.

Ishibuchi, H., and T. Yamamoto. "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining." *Fuzzy Sets and Systems* 141, no. 1 (2004): 59-88.

Ishibuchi, H., K. Nozaki, and H. Tanaka. "Distributed representation of fuzzy sets and its application to pattern classification." *Fuzzy Sets and Systems* 52, no. 1 (1992): 21-32.

Ishibuchi, H., T. Nakashima, and M. Nii. "Genetic-algorithm-based instance and feature selection." In *Instances selection and construction for data mining*, by H. Lui and H. Motoda (eds.), 95-112. Boston: Kluwer Academic Publishers, 2001.

Jain, A., and D Zongker. "Feature selection: evaluation, application, and small sample performance." *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19, no. 2 (1997): 153-158.

Jang, J-S,R., and C-T Sun. "Neuro-fuzzy modeling and control." *Proceeding of the IEEE*, 1995: 378-406.

Jiang, M., Y.P. Luo, and S.Y. Yang. "Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm." *Information Processing Letters* 102, no. 1 (2007): 8-16.

Jin, Y. "Fuzzy modeling of high-dimensional systems:Complexity reduction and interpretability improvement." *IEEE Trans. Fuzzy Systems* 8, no. 2 (2000): 212-221.

Karray, R.O., and C. DeSilva. *Soft computing and intelligent system design.* Harlow: Pearson Education Limited, 2004.

Kennedy, J., and R.C. Eberhart. "A discrete binary version of the particle swarm algorithm." *IEEE Int. Conf. on Systems, Man, and Cybernetics.* Washington, 1997. 4104-4108.

Kennedy, J., and R.C. Eberhart. "Particle swarm optimization." *IEEE Int. Conf. on Neural Networks.* Perth, 1995. 1942-1948.

Khanesar, M.A., and M. Shoorehdeli, M. A. Teshnehlab. "A novel binary particle swarm optimization." *Mediterranean Conf. on Control & Automation.* Athens, 2007. 1-6.

Kim, E., M. Park, S. Ji, and M. Park. "A new approach to fuzzy modeling." *IEEE Trans. on Fuzzy Systems* 5, no. 3 (1997): 328-337.

Kosko, B. *Neural Networks and Fuzzy Systems.* Englewood Cliffs, N.J.: Prentice Hall, 1992.

Krone, A., P. Krause, and T. Slawinski. "A new rule reduction method for finding interpretable and small rule bases in high dimensional search spaces." *IEEE International Conf. on Fuzzy Systems.* San Antonio, TX, 2000. 694-699.

Lee, H.M., C.M. Chen, J.M. Chen, and Y.L. Jou. "An efficient fuzzy classifier with feature selection based on fuzzy entropy." *IEEE Trans. on Syst., Man, and Cybernetics* 31, no. 3 (2001): 426-432.

Lui, H., and H. Motoda. *Computational Methods of Feature Selection.* Boca Raton: Chapman & Hall/ CRC, 2008.

Lui, H., and H. Motoda. *Instance Selection and Construction for Data Mining.* Boston: Kluwer Academic Publishers, 2001.

Lui, H., and L. Yu. "Toward integrating feature selection algorithms for classification and clustering." *IEEE Trans. on Knowledge and Data Engineering* 17, no. 14 (2005): 491-502.

Lui, H., and M. Horoshi. *Feature Selection for Knowledge Discovery and Data Mining.* Boston: Kluwer Academic Publishers, 1998.

Mamdani, H. E., and S. Assilian. "An experiment in linguistic synthetis with a fuzzy logic controller." *Int. Journal Man-Machine Studies* 7, no. 1 (1975): 1-13.

Marcelloni, F. "Feature selection based on a modified fuzzy c-means algorirthm with supervision." *Information Sciences* 151 (2003): 201-226.

Marwah, G, and L Boggess. "Artificial immune systems for classification ." *International Conf. on Artficial Immune Systems.* Canterbury, UK, 2002. 149-153.

Mikut, R., J. Jakel, and L. Groll. "Interpretability issues in data-based learning of fuzzy systems." *Fuzzy Sets and Systems* 150, no. 2 (2005): 179-197.

Nayak, P.C., and Sudheer. "Fuzzy model identification based on cluster estimation for reservoir inflow forecasting." *Hydrological Proceses* 22 (2008): 827-841.

Nouvo, A. G. D., and V. Catania M. Palesi. "Multi-objective evolutionary fuzzy clustering for high dimensional problems." *IEEE Int. Conf. on Fuzzy Systems.* London, 2007. 1-6.

Nuovo, A. G. D., and V. Catania. "An evolutionary fuzzy c-means approach for clustering of bio-informatics databases." *IEEE Int. Conf. on Fuzzy Systems.* Hong Kong, 2008. 2077-2082.

Oliveira, J.V de, and W. Pedrycz. (eds.) *Advances in Fuzzy Clustering and its Application.* West Sussex: John Wiley & Sons, 2007.

Olvera-Lopez, J. A., J. A. Carrasco-Ochoa, J. F. Martinez-Trinidad, and J. Kittler. "A review of instance selection methods." *Artificial Intelligence Review* 34, no. 2 (2010): 133-143.

Paterlini, S., and T. Krink. "Differential evolution and particle swarm optimization in partitional clustering." *Comp. Stats. & Data Analysis* 50, no. 5 (2006): 1220-1247.

Pavlenko, T. "On feature selection, curse-of-dimensionality and error probability in discriminant analysis." *Journal of statistical planning and inference* 115, no. 2 (2003): 565-584.

Pawlak, Z. "Rough set." *International Journal of Computer and Information Science* 11, no. 5 (1982): 341-356.

Pedrycz, W. "An identification algorithm in fuzzy relation systems." *Fuzzy Sets and System* 13, no. 2 (1984): 153-167.

Pedrycz, W. "Data and dimensionality reduction in data analysis." In *Encyclopedia of Complexity and System Science*, by R. A. Meyers(ed.), 1775-1789. Berlin: Springer, 2009.

Pedrycz, W. *Fuzzy Control and Fuzzy Systems.* Research Studies Press, 1993.

Pedrycz, W. *Granular Computing with shadowed sets.* Vol. 3641, in *Rough sets, fuzzy sets, data mining, and granular computing*, by Dominik Slezak, Guoyin Wang, Marcin Szczuka, Ivo Duntsch and Yiyu Yao, 23-32. Springer Berlin / Heidelberg, 2005.

Pedrycz, W. *Knowledge-based Clustering: From Data to Information Granules.* Hoboken: John Wiley & Sons, Inc, 2005.

Pedrycz, W., and A. Bargiela. "An optimization of allocation of information granularity in the interpretanble of data structures: toward granular fuzzy clustering." *IEEE Trans. Syst., Man, and Cybernetics, Part B*, 2011: 1-9.

Pedrycz, W., and A. Bargiela. "Fuzzy clustering with semantically distinct families of variables: descriptive and predictive aspects." *Pattern Recognition Letters* 31, no. 13 (2010): 1952-1958.

Pedrycz, W., and F. Gomide. *An Introduction to Fuzzy Sets: Analysis and Design.* Cambridge: MIT Press, 1998.

Pedrycz, W., and F. Gomide. *Fuzzy System Engineering: Toward Human-Centric Computing.* Hoboken: John Wiley & Sons, Inc, 2007.

Pedrycz, W., and G. Vokovich. "Abstraction and specialization of information granules." *IEEE Trans. on Syst. Man, and Cybernetics* 31, no. 1 (2001): 106-111.

Pedrycz, W., and J. Valenta de Oliveira. "A development of fuzzy encoding and decoding through fuzzy clustering." *IEEE Trans. On Instrument and Measurement* 57, no. 4 (2008): 829-837.

Raymer, M.L., W. F. Punch, E.D. Goodman, L.A. Kuhn, and A.K. Jain. "Dimensionality reduction using genetic algorithms." *IEEE Trans. on Evolutionary Computation* 4, no. 2 (2000): 164-171.

Rezaee, B., and M.H. F. Zarandi. "Data-driven fuzzy modeling for Takagi-Sugeno-Kang fuzzy system." *Information Sciences* 180, no. 2 (2010): 241-255.

Ritter, G.L., Woodruff, and S.R, Isenhour Lett, T.L Lowry. "An algorithm for a selective nearest neighbor decision rule." *IEEE Trans. on Information Theory* 21, no. 6 (1975): 665-669.

Roubos, H., and M. Setnes. "Compact and transparent fuzzy models and classifiers through iterative complexity reduction." *IEEE Trans. on Fuzzy Systems* 9, no. 4 (2001): 516-524.

Setnes, M., and R. Verbruggen, H.B. Babuska. "Complexity reduction in fuzzy modeling." *Mathematics and Computer in Simulation* 46, no. 5-6 (1998): 507-516.

Setnes, M., R. Babuska, U. kaymak, and H. R. van Nauta Lemke. "Similarity measures in fuzzy rule base simplification." *IEEE Trans. On Syst., Man, and Cybernatics-Part B* 28, no. 3 (1998): 376-386.

Shi, Y., and R. C. Eberhart. "Empirical study of particle swarm optimization." *Congress of Evolutionary Computation.* Washington, 1999. 1945-1950.

Sindelar, R., and R. Babuska. "Input selection for nonlinear regression models." *IEEE Trans. on Fuzzy Systems* 12, no. 5 (2004): 688-696.

Sugeno, M, and G. T. Kang. "Structure identification of fuzzy model." *Fuzzy Sets and Systems* 28, no. 1 (1988): 15-33.

Takagi, T, and M. Sugeno. "Fuzzy identification of systems and its applications to modeling and control." *IEEE Trans. On Systems, Man, and Cybernetics* SMC-15, no. 1 (1985): 116-132.

Tanaka, K, T Taniguchi, and H.O. Wang. "Generalized Takagi-Sugeno fuzzy systems: rule reduction and robust control." *IEEE International Conf. on Fuzzy Systems.* San Antonio, TX, 2000. 688-693.

Taniguchi, T., K. Tanaka, H. Ohtake, and H.O. Wang. "Model construction, rule reduction, and robust compensation for generalized form of Takagi-Sugeno fuzzy systems." *IEEE Trans. on Fuzzy Systems* 9, no. 4 (2001): 525-538.

Tsekouras, G. Sarimveis, H., and G. Bafas. "A simple algorithm for training fuzzy systems using input-output data." *Advances in Engineering Software* 34, no. 5 (2003): 247-259.

Tsekouras, G., H. Sarimveis, E. Kavakli, and G. Bafas. "A hierarchical fuzzy-clustering approach to fuzzy modeling." *Fuzzy Sets and Systems* 150, no. 2 (2005): 245-266.

Tsekourus, G.E. "On the use of the weighted fuzzy c-means in fuzzy modeling." *Advances in Engineering Software* 36, no. 5 (2005): 287-300.

Tsukamoto. "An approach to fuzzy reasoning method." In *Advances in Fuzzy Set Theory and Application*, by M.M. Gupta, R. K. Ragade and R.R. Yager (eds.), 137-149. North-Holland, Amsterdam, 1979.

Turksen, I.B., Tian, Y., and M. Berg. "A fuzzy expert system for a service center of spare parts." *Expert Systems with Applications* 5, no. 3-4 (1992): 447-464.

Uncu, O., and I.B. Turksen. "A novel feature selection approach: Combining feature wrappers and filters." *Information Sciences* 177, no. 2 (2007): 449-466.

van den Bergh, F. *An analyis of particle swarm optimizers.* PhD thesis, Department of Computer Science, University of Pretoria, South Africa, 2002.

van den Bergh, F., and A. P. Engelbrecht. "A cooperative approach to particle swarm optimization." *IEEE Trans on Evolutionary Computation* 8, no. 3 (2004): 225-239.

Wan, F., H. Shang, L. Wang, and Y. Sun. "How to determine the minimum number of fuzzy rules to achieve given accuracy: A computational geometric approach to SISO case." *Fuzzy Sets and Systems* 150, no. 2 (2005): 199-209.

Wang, H., S. Kwong, Y. Jin, W. Wei, and K.F. Man. "Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction." *Fuzzy Sets and Systems* 149, no. 1 (2005): 149-186.

Wang, L.X., and J.M Mendel. "Generating fuzzy rules by learning from examples." *IEEE Trans. on Syst., Man, and Cybernetics* 22, no. 6 (1992): 1414-1424.

Wang, X., J. Yang, X. Teng, W. Xia, and R. Jensen. "Feature selection based on rough sets and particle swarm optimization." *Pattern Recognition Letters* 28, no. 4 (2007): 459-471.

Whitley, D. "A genetic algorithm tutorial." *Statistics and Computing* 4, no. 2 (1994): 65-85.

Wilson, D.R., and T.R Martines. "Reduction techniques for instance-based learning algorithms." *Machine Learning* 38, no. 3 (2000): 257-286.

Xiong, N., and L. Litz. "Reduction of fuzzy control rules by means of premise learning – method and case study." *Fuzzy Sets and Systems* 132, no. 2 (2002): 217-231.

Yager, R.R. "On hierarchical structure for fuzzy modeling and control." *IEEE Trans. on Sys., Man, and Cybernetics* 23, no. 4 (1993): 1189-1197.

Yang, J., and V. Honavar. "Feature subset selection using genetic algorithm." *IEEE Intelligent Systems and their Applications* 13, no. 2 (1998): 44-49.

Yao, Y. "Perspective of granular computing." *IEEE International Conf. on Granular Computing.* 2005. 85-90.

Yi, S.Y., and M.J. Chung. "Identification of fuzzy relational model and its application to control." *Fuzzy Sets and Systems* 59 (1993): 25-33.

Yoshinari, Y., W. Pedrycz, and K. Hirota. "Construction of fuzzy models through clustering techniques." *Fuzzy Sets and Systems* 54, no. 2 (1993): 157-165.

Yu, F., and W. Pedrycz. "The design of fuzzy granules: Tradeoffs between specificity and experimental evidence." *Applied Soft Computing* 9, no. 1 (2009): 264-273.

Zadeh, L.A. "Fuzzy sets." *Information and Control*, 1965: 338-353.

Zadeh, L.A. "Key roles of information granulation and fuzzy logic in human reasoning, concept formation and computing with words." *IEEE 5th International Fuzzy System.* 1996. 8-11.

Zadeh, L.A. "Outline of a new approach to the analysis of complex systems." *IEEE Trans. Systems, Man and Cybernetics* SMC-3, no. 1 (1973): 28-44.

Zadeh, L.A. "Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic." *Fuzzy Sets and Systems* 2, no. 90 (1997): 111-127.

Zarandi, M. H. F., I. B. Turken, and B. Rezaee. "A systematic approach to fuzzy modeling for rule generation from numerical data." *IEEE Annual Meeting of the Fuzzy Information.* 2004. 768-773.

Zhang, Q., and M. Mahfouf. "A hierarchical mamdani-type fuzzy modeling approach with new training data selection and muti-objective optimisation mechanisms: A special application for the prediction of mechanical properties of alloy steels." *Applied Soft Computing* 11, no. 2 (2011): 2419-2443.

Zhang, Y., X. Wu, Z. Xing, and W. Hu. "On generating interpretable and precise fuzzy systems based on pareto multi-objective cooperative co-evolutionary algorithm." *Applied Soft Computing* 11, no. 1 (2011): 1284-1294.

Zhou, S. M., and J. Q. Gan. "Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modeling." *Fuzzy Sets and Systems* 159, no. 23 (2008): 3091-3131.