DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

INVESTIGATION OF FUZZY FUNCTIONS APPROACH AND ITS POSSIBLE APPLICATIONS IN INDUSTRIAL ENGINEERING PROBLEMS

by
Sultan MARAL

March, 2013
İZMİR

INVESTIGATION OF FUZZY FUNCTIONS APPROACH AND ITS POSSIBLE APPLICATIONS IN INDUSTRIAL ENGINEERING PROBLEMS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Industrial Engineering, Industrial Engineering Program

by
Sultan MARAL

March, 2013
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"INVESTIGATION OF FUZZY FUNCTIONS APPROACH AND ITS POSSIBLE APPLICATIONS IN INDUSTRIAL ENGINEERING PROBLEMS"** completed by **SULTAN MARAL** under supervision of **PROF. DR. ADİL BAYKASOĞLU** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof.Dr. Adil BAYKASOĞLU

Supervisor

Yrd. Doç. Dr. Serdar TAŞAN	Doç. Dr. M. Evren TOYGAR
(Jury Member)	(Jury Member)

Prof.Dr. Ayşe OKUR
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

First and foremost I would like to express my gratitude and thanks to my graduate thesis advisor, Prof. Dr. Adil BAYKASOĞLU, whose support I always felt for leading and encouraging me to complete my master degree within the period of my master studies. Thankful to him for his great help me to overcome the problems I encountered during my studies.

I am also grateful to my dear colleagues and would like to thank them for being helpful and understanding towards me throughout my studies.

Finally I would like to pay my respect to my beloved family; my mother Menekşe MARAL, my father Ali MARAL, my sister İlknur MARAL and my twin brother Hıdır MARAL for their continuous support, patient and encouragement through all my life.

Sultan MARAL

INVESTIGATION OF FUZZY FUNCTIONS APPROACH AND ITS POSSIBLE APPLICATIONS IN INDUSTRIAL ENGINEERING PROBLEMS

ABSTRACT

Fuzzy set theory was introduced by Zadeh in 1965 as an extension to classical set theory. It has been a very important research subject for many researchers and has led to new developments for many fields since it enables to handle uncertainties successfully. One of these important developments is the fuzzy functions concept which was introduced by Professor I. Burhan Türkşen and combines fuzzy sets and fuzzy clustering concepts to provide an alternative solution approach to solve problems in diverse domains**.** The novelty of fuzzy functions is based on the fuzzy clustering concept and therefore based on fuzzy membership values. Fuzzy clustering is one of the corner stone of the fuzzy functions since finding the best partition constitutes the main problem in this approach. There are several fuzzy clustering algorithms in the literature which can be used in generating fuzzy functions. In this thesis Fuzzy c-Means (FCM) clustering algorithm is used in order to find out the membership values.

One of the main motivations behind the development of the fuzzy functions approach was to overcome some of the drawbacks of the fuzzy rule bases which are one of the most frequently used fuzzy inference methods with many successful applications.

As a contribution to the existing studies about fuzzy functions, first time in the present thesis we proposed to use genetic programming (GP) along with fuzzy clustering as a new approach in generating fuzzy functions. We used many data sets from the literature in order to present the application and the performance of our approach. We also performed comparisons with the existing fuzzy function generation methods like Least Square Estimation (LSE) in order to prove the validity of our approach. Based on the computational results we illustrated that fuzzy functions which are generated through genetic programming are very competitive and effective in many problem settings.

Keywords: Fuzzy set theory, fuzzy rule bases (FRB), fuzzy clustering, fuzzy functions (FF), least square estimation (LSE), support vector machines (SVM), genetic programming (GP).

BULANIK FONKSİYON YAKLAŞIMININ ARAŞTIRILMASI VE ENDÜSTRİ MÜHENDİSLİĞİ PROBLEMLERİNDE OLASI UYGULAMALARI

ÖZ

Bulanık küme teorisi, Zadeh tarafından 1965'de klasik küme teorisinin genişletilmiş bir şekli olarak ortaya atılmıştır. Bulanık küme teorisi birçok araştırmacı için çok önemli bir araştırma konusu olmuş ve belirsizliklerle başarılı bir şekilde baş etme olanağı sağladığı için birçok alanda yeni gelişmelere yol açmıştır. Bu önemli gelişmelerden biri de, Profesör I. Burhan Türkşen tarafından ortaya atılan ve çeşitli alanlardaki problemlerin çözümünde alternatif çözüm yaklaşımı sağlamak için bulanık küme ve bulanık kümeleme kavramlarını kombine eden bulanık fonksiyonlardır. En iyi bölümlemeyi bulmak bulanık fonksiyonlar yaklaşımın temel problemini oluşturduğundan dolayı, bulanık kümeleme, bulanık fonksiyonların temel taşlarından biridir. Literatürde, bulanık fonksiyonları üretmede kullanılabilen çeşitli bulanık kümeleme algoritmaları vardır. Bu çalışmada, üyelik değerlerini bulmak için Fuzzy c-Means (FCM) kümeleme algoritması kullanılmaktadır.

Bulanık fonksiyon yaklaşımının gelişiminin arkasındaki ana etkenlerden biri, pek çok başarılı uygulaması olan ve en sık kullanılan bulanık çıkarsama yöntemlerinden biri olan bulanık kural tabanlarının bazı dezavantajlarının üstesinden gelmektir.

Bulanık fonksiyonlar ilgili var olan çalışmalara katkı olarak, mevcut tezde ilk defa yeni bir yöntem olarak bulanık fonksiyonların oluşturulmasında, bulanık kümelemeyle birlikte genetik programlamanın (GP) kullanmasını önerdik. Yaklaşımımızın uygulanışını ve performansını göstermek için literatürden birçok veri setini kullandık. Ayrıca yaklaşımımızın geçerliliğini kanıtlamak için En Küçük Kareler Yöntemi (EKKY)gibi mevcut yöntemler ile oluşturulan bulanık fonksiyonları kullanarak karşılaştırmalar yaptık. Sayısal sonuçlara dayanarak, genetik programlamayla oluşturulan bulanık fonksiyonların birçok problem kümelerinde rekabetçi ve etkili olduklarını örneklendirdik.

Anahtar sözcükler: Bulanık küme teorisi, bulanık kural tabanları (BKT), bulanık kümeleme, bulanık fonksiyonlar (BF), en küçük kareler yöntemi (EKKY), destek vektör makineleri (DVM), genetik programlama (GP).

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

CHAPTER ONE
INTRODUCTION

1.1 Background

Uncertainty is an important part of the systems and almost all of the problems encountered in real life stem from containing uncertainty. Therefore defining and modeling the systems appropriately constitutes the basis of problems. This uncertainty leads to subjectivity of the expressions which could be changed from different points of view and limits measuring the performance of the systems. The classical set theory ignores this uncertainty and defines the systems with sharp boundaries such as true or false expressions. According to classical set theory an element either is a member of a set or not. When it is thought the element belongs to a set, it is represented with “1”, when it is thought the element does not belong to a set, it is represented with “0” which could be liken to seeing the glass either empty or full ignoring the water inside the glass. There is a sharp distinction between the element and the set. But in real life elements are not classified with sharp boundaries and the classical set theory of 0-1 cannot reflect the systems adequately. Because of that classical set theory is not capable of explaining such vague systems precisely. In order to eliminate such an important insufficiency, Prof. Dr. Lotfi Zadeh proposed fuzzy set theory in 1965 and since than it has become a very important subject. In his article (1965) he described this new concept as follows; “a fuzzy set is as a class of objects with a continuum of grades of membership” (p. 338) and claimed that, an element of a set can take values between 0 and 1 which represents the degree of belongings of the element to a fuzzy set. Therefore it could be said that fuzzy set theory describes the systems more accurately and gives better results when classical set theory is not successful and sufficient.

Since Prof. Dr. Lotfi Zadeh introduced fuzzy set theory, thanks to enabling to cope with data more sufficiently, it has been a very important way of analyzing and modeling the systems. As it was expressed by Çelikyılmaz (2005), “fuzzy logic (FL) provides a means for modeling linguistic terms (i.e., fair, good, excellent) by utilizing membership functions; and in turn provides a framework for Fuzzy System Modeling” (p. 2).

After fuzzy logic theory has become widely known and itsimportance has been understood, it has formed the basis of many well-known and efficient researches. One of them is fuzzy rule bases concept which is originally proposed by Zadeh (1973) and then studied and developed by many of researchers. Many researchers such as Mamdani (1974) and Takagi & Sugeno (1985) have made important contributions depending on the encountered problems in the course of application.

Fuzzy rule bases concept is one of the most known fuzzy inference methods and could be defined as a system thatis composed of a set of rules which describe the relationships between inputs and outputs with linguistic variables. The ability of fuzzy rule bases to model complex systems and developing rules that make intuitive sense are some of the important advantages of fuzzy rule bases. But despite the widespread use of fuzzy rule bases, enabling to model complex systems easily and successful applications, fuzzy rule bases still have some important drawbacks that obstruct to define systems easily and correctly when the systems are being larger besides fuzzy rule bases require expert knowledge. All these aforementioned subjects and more detailed information concerning the fuzzy rule bases could be found out in chapter 2.

Fuzzy functions concept, which wasproposed by Professor I. Burhan Türkşen in order to overcome all aforementioned deficiencies of fuzzy rule bases such as dependence on expert knowledge and complexity of required operators during the modeling and analyzing phase, forms the basis of this study. Fuzzy functions concept could be defined as a combination of functions and fuzzy sets that offers a more objective way of analyzing the systems. In the literature “fuzzy functions” term has been used in order to describe many different concepts. Among them, the most widely used is the one which represents the membership functions. One of the examples of other definitions is mathematical definition of fuzzy functions that is proposed by Professor Mustafa Demirci (1999, 2000 and 2001). The implied meaning of fuzzy functions suggested by Demirci is different from fuzzy functions concept that is proposed by Professor I. Burhan Türkşen. However it would not be wrong to say that fuzzy functions term used by Demirci underlines the mathematical basis of Türkşen's fuzzy functions concept.

In their studies, Çelikyılmaz and Türkşen (2007a, 2007b, 2008a and 2008b) have applied fuzzy functions to many dataset from the literature and have shown that this proposed approach gives more efficient results in comparison to fuzzy rule bases.

> "Fuzzy Functions" are multi-variable crisp valued functions. The prominent feature of these functions $f(X,\mu)$ are that they use the degree of membership μ, of each object to the specified fuzzy set as an additional attribute just as the rest of the input variables, X. In a sense, the gradations (membership values) become the predictors. This type of "Fuzzy Functions" emerged from the idea of representing each unique fuzzy rule in terms of functions (Çelikyılmaz and Türkşen, 2009b,p. 35).

According to Türkşen's approach membership values and some of their transformations such as exponential and logarithmic transformations are added as new variables to the original datasets. As it could be understood from here, membership values are the keystones of fuzzy functions. In the literature many different methods have been proposed for the purpose of finding membership values and for thepresent study fuzzy c-means (FCM) clustering algorithm is taken as a basis in order to obtain membership values. As Rezaee, Lelieveldt and Reiber (1998) defined, "The objective of most clustering methods is to provide useful information by grouping (unlabeled) data in clusters; within each cluster the data exhibits similarity" (p. 237). As stated by Rezaee et al. (1998) similarity is very important and constitutes the basis of fuzzy clustering. Therefore many methods have been proposed in order to measure the validity of fuzzy clustering algorithms.

In chapter 6, for the implementation phase of fuzzy functions, three different ways are followed. After membership values of datasets which are taken from UCI learning machine repository have been found, first of all, only these membership values are added to the original input variables as new predictors. Then respectively four and two different transformations of these membership values are added as new variables to original input variables. But before fuzzy functions with LSE is applied to these datasets, an artificial dataset is generated and Türkşen's proposed algorithm is explained via this artificial dataset step by step in chapter 4. Then in the next chapter genetic programming concept which is the main focus of this study and forms the basis of fuzzy functions with genetic programming is introduced and the algorithm is explained with the generated artificial dataset.

1.2 The Main Scope of the Study

Based on Türkşen's fuzzy functions approach, the proposed model of fuzzy functions with genetic programming (GP) forms the basis of this study. The purpose in using genetic programming is to search whether using the proposed model is increasing the performance of fuzzy functions or not.

> Langdon, Poli, McPhee and Koza (2008) defined genetic programming (GP) as an evolutionary computation (EC) technique that automatically solves problems without having to tell the computer explicitly how to do it. At the most abstract level GP is a systematic, domain-independent method for getting computers to automatically solve problems starting from a high-level statement of what needs to be done (p. 927).

Genetic programming is an efficient technique on its own, and gives competitive results compared to other techniques. In the literature, there are many studies that combine the genetic programming with other techniques. From this point of view, assuming that using genetic programming with fuzzy functions may improve the performance of fuzzy functions, just as in the case of the application of fuzzy functions with LSE, the same three methods are followed for fuzzy functions with GP and the same datasets and transformations are used for all methods. Moreover the same artificial dataset is used in order to explain the algorithm of the proposed model of fuzzy functions with GP.

After the algorithm of the proposed model is explained step by step with an artificial dataset, the proposed model is applied to all datasets and then the results of fuzzy functions with GP and the results of fuzzy functions with LSE are compared. With the intention of being able to compare in itself R-squarevalues of training, validation and testing data are calculated for fuzzy functions with LSE. However in order to be able to compare fuzzy functions with LSE and fuzzy functions with GP, R-square values are calculated for also whole datasets without separating into training or testing data. Afterwards based on these R-square values, the validity of the proposed model is discussed.

1.3 The Structure of The Thesis

The present thesis consists of seven chapters and organized as follows. In chapter 1, a brief introduction is made on the course of the study. In chapter 2, the fundamental theory of fuzzy rule bases; mostly used types of fuzzy rule bases and their main drawbacks are explained in detail. Fuzzy clustering concept which constitutes the basis of the fuzzy functions; type of fuzzy clustering algorithms and most widely used clustering validity indexes that provide to determine best possible fuzzy partition are presented in chapter 3. In chapter 4, outlines of fuzzy functions concept and fuzzy functions with Least Square Estimation (LSE) is explained step by step with an artificial dataset. After fuzzy functions concept is overviewed, the proposed method of fuzzy functions with genetic programming approach is discussed and the algorithm is explained with the same artificial dataset in chapter 5. In chapter 6, the datasets taken from UCI Machine Learning Repository are evaluated with “fuzzy functions with LSE” and “fuzzy functions with genetic programming”. Finally the study is ended with chapter 7 in which a brief summary of the study is provided, conclusions are reviewed and potential future researches are stated.

CHAPTER TWO
A BRIEF OVERVIEW OF FUZZY RULE BASES

2.1 Introduction

A system can be described as a collection of elements which have relationships with each other and aiming at a common purpose. As much as modeling the systems always has been an important subject for researches, defining these systems appropriately has also become an important part of the problems and constitutes prerequisite step to able to modeling the systems. However systems often contain linguistic expressions and are stated with linguistic variables which in other words mean subjectivity. Therefore modeling the systems that composed of linguistic variables is quite difficult and the classical inference systems are not sufficient for these systems and do not reflect the accurate results. The notion of fuzzy system deals with such these problems.

Palit and Popovic (2005) stated that "Fuzzy systems are unique in the sense that they can simultaneously process numerical data and linguistic knowledge" (p. 146). As it was expressed by Palit and Popovic, thanks to that fuzzy systems allows both processing numerical and linguistic variables, modeling the systems realistically become easier. This advantage has provided fuzzy systems to be widespread in a short time and to be used successfully for various purposes such as for prediction, modeling and classification.

After Zadeh introduced fuzzy set theory in 1965 and then its advantages were discovered, many researches on fuzzy sets have been made. In the literature many studies have been proposed on fuzzy sets. Between them the most commonly known and applied fuzzy inference system is fuzzy rule bases system which is also originally introduced by Zadeh in 1973 and then developed by many researchers.

In his study, Zadeh (1973) described the difference of his proposed approach from the conventional quantitative techniques of system analysis. As it was expressed by Zadeh (1973), the proposed approach has three main distinguishing features: "1) use of so-called "linguistic" variables in place of or in addition to numerical variables; 2) characterization of simple relations between variables by fuzzy conditional statements; and 3) characterization of complex relations by fuzzy algorithms"(p. 28). More information could be found in his study which is called "Outline of a new approach to the analysis of complex systems and decision processes".

In the following section fuzzy rule bases concept is reviewed and then detail information on most commonly known and used types of fuzzy rule bases is given.

2.2 Fuzzy Rule Bases

In their study Cordon, Herrera, Hoffmann and Magdalena (2001) described fuzzy rule bases as follows; "FRBS is a rule-based system where fuzzy logic (FL) is used as a tool for representing different forms of knowledge about the problem at hand, as well as for modeling the interactions and relationships that exist between its variables" (p.1).

Fuzzy rule bases concept is one of the most known fuzzy inference method and could be defined as a system thatis composed of a set of rules which describe the relationships between inputs and outputs with linguistic variables. Due to consisting of a set of if-then rules fuzzy rule bases are generally known as IF-THEN rules and in a general structure of fuzzy rule base, IF part represents the antecedent part and THEN part represents the consequent part of a system. Explaining mathematically, the general form of a fuzzy rule base is, IF *antecedent propositions* THEN *consequent proposition.* The general representation is shown as follows;

$$If\ X1\ is\ A1\ and;\ X2\ is\ A2, then\ y\ is\ B, \tag{2.1}$$

Due to fuzzy rule bases composed of linguistic variables such as IF, THEN rules and do not contain any mathematical values, while fuzzy rule bases are handled researchers could be confronted with some important problems which are explained in details in the following parts.

As it can be seen in the Figure 2.1, a typical fuzzy interface system is composed of a few elements. Rule bases block represents the IF-THEN rules and the database block defines the membership functions of fuzzy sets. Fuzzification interface is the process where the crisp values are transformed into fuzzy values. In order to get a crisp solution, contrary to fuzzification interface, in defuzzification interface obtained fuzzy values are transferred into crisp values. And the decision making unit block represents that all these processes are done by the decision making unit.

As it is mentioned above, fuzzy rule bases are composed of a set of operators that provide to convert crisp variables into fuzzy variables and also fuzzy variables into crisp variables. Therefore the identification of right operators and variables and their proper use are very important for modeling systems ideally. Because of that in order to improve the efficiency of the systems, many studies have been made and still many researchers study for the correct identification of systems.

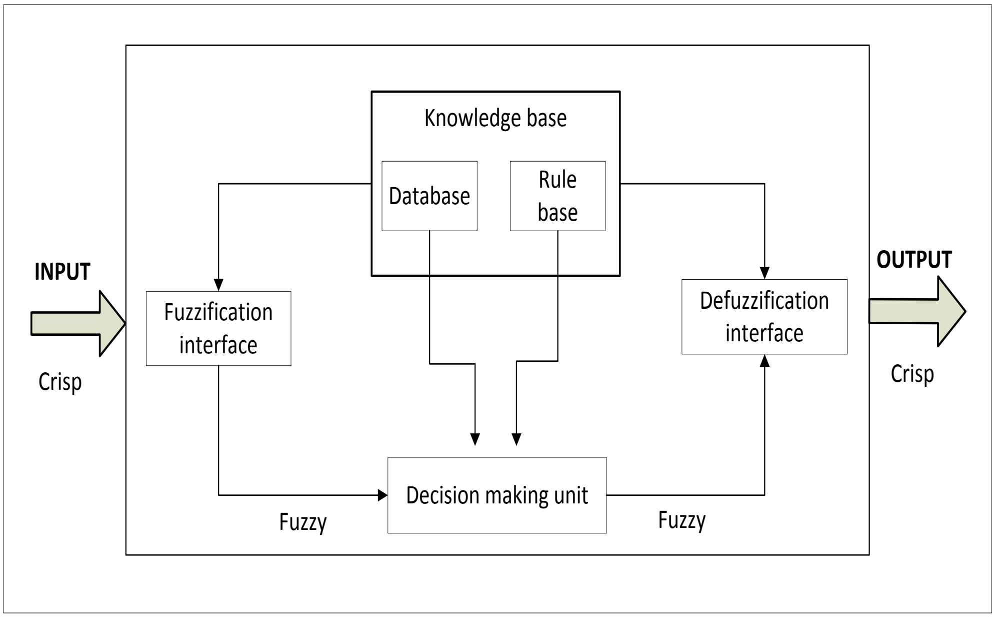


Figure 2.1 A typical fuzzy inference system (Moallem, Mousavi and Monadjemi, 2011)

Fuzzy rule base system was firstly applied by Mamdani (1974). With his studyMamdani applied fuzzy rule bases to a simple dynamic plant - a model steam engine and Mamdani’s study showed that fuzzy rule base inference systems could be applied to such these areas easily and successfully.

Tsoi and Gao (1999) used fuzzy rule bases system to control injection velocity for thermoplastics injection molding and based on the results of the experiments, in their study they indicated that “the fuzzy logic-based controller works well with different molds, materials, barrel temperatures, and injection velocity profiles, indicating that the fuzzy logic controller has superior performance over the conventional PID controller in response speed, set-point tracking ability, noise rejection, and robustness” (p. 3).

As it was mentioned by Leondes (1998) in his study, fuzzy rule bases have an extensive range of application areas. Some example studies on fuzzy rule bases are as follows:

- Tsoi and Gao (1999) used fuzzy rule bases in order to control injection velocity for thermoplastics injection molding which is widely using and important in plastic processing.

- Traffic signal control is one of the oldest applications of fuzzy logic theory and in the study of “general fuzzy rule base for isolated traffic signal control-rule formulation” Niittymaki (2001) used fuzzy rule bases for traffic signal control.

- Surmann and Selenschtschikow (2002) appliedgenetic fuzzy rule base learning algorithm to some datasets taken from machine learning repository in order to compare the results with other approaches.

- Chang and Chen (2009) used fuzzy rule bases and fuzzy clustering techniques in order to predict the temperature based on the data set of the daily average temperature and the data set of the daily average cloud density.

- Based on Mamdani fuzzy rule base system, Sivarao, Brevern, El-Tayeb and Vengkatesh (2009) developed a Matlab GUI in order to predict surface roughness in laser machining.

- Kaur and Kaur (2012) both applied Mamdani and Takagi-Sugeno fuzzy rule base for air conditioning system and compared the results.

- Moallem et al. (2011) proposed a novel fuzzy rule base system and applied this proposed fuzzy rule based system for pose, size, and position independent face detection in color images.

- Kamyab and Bahrololoum (2012) used TSK fuzzy rule based system with bacterial foraging optimization algorithm (BFOA) in order to simulate the foraging behavior.

- In their study which was named as "a genetic fuzzy-rule-based classifier for land cover classification from hyperspectral imagery" Stavrakoudis, Galidaki, Gitas, and Theocharis (2012) used fuzzy rule bases for land cover classification by combining genetic programming.

From this point of view the wide range of application areas of fuzzy rule bases can be seen clearly. In the following section most commonly known types of fuzzy rule bases are introduced. Some of the most commonly used fuzzy rule bases are Zadeh' fuzzy rule base, Takagi-Sugeno (TSK) fuzzy rule base, Mamdani's rule base and Mizumoto's fuzzy rule base system. Detailed information on the fundamental theory of these fuzzy rule bases and the difference between them are explained briefly in the next section.

2.2.1 Zadeh's Fuzzy Rule Base Structure

"Zadeh first introduced the Fuzzy Modus Ponens known as Generalized Modus Ponens (GMP) and defined a methodology known as Compositional Rule of Inference (CRI), which is used to infer fuzzy consequents. Generally, GMP is shown as follows"(Çelikyılmaz, 2005, p. 21);

$$Premise1: A \rightarrow B$$
$$Premise2: A^{'} \qquad (2.2)$$
$$Deduction: B^{*}$$

Where A and A' are fuzzy sets corresponding to linguistic values of linguistic variables defined on the universe of discourse of antecedent variable x with membership functions $\mu_A(x): x \in X \rightarrow [0,1]$ and B and B^* are linguistic values of linguistic variable defined on the universe of discourse of the consequent variable y with membership functions, $\mu_B(y): y \in Y \rightarrow [0,1]$. → denotes the implication relation operator and each premise is a relation and denoted as $R_i: A \rightarrow B, i: 1, ...,$ number of relations (Çelikyılmaz, 2005, p. 21).

The above mentioned equations could be also indicated as in equation (2.3) where "o" represents the composition operator and "→" represents the implication operator.

$$B^* = A^{'} o (A \rightarrow B) \qquad (2.3)$$

Another and common representation of Zadeh's (1965) fuzzy rule base structure is formulated as follows (Çelikyılmaz and Türkşen, 2009b, p. 36):

$$\mathcal{R}: \underset{i=1}{\overset{c}{ALSO}} \left[IF \underset{j=1}{\overset{nv}{AND}} \left(x_j \in X_j \, isr A_{ij} \right) THEN y \in Y isr B_i \right] \qquad (2.4)$$

- c is the number of rules,
- x_j represents the jth input variable and $j = 1, \dots, nv$, nv represents the number of input variables, X_j is the domain of x_j
- A_{ij} is the linguistic label associated with input variable x_j in rule i with membership function $\mu_{Aij}(x_i): X_j \rightarrow [0, 1]$
- y is the output variable of each rule, Y is the domain of y,
- B_i is the linguistic label associated with the output variable in the ith rule with the membership function $\mu_{Bi}(y): Y \rightarrow [0, 1]$

- AND is the logical connective that aggregate the membership values of input variables for a given observation,
- THEN (→) is the logical implication connective,
- ALSO is the logical connective used to aggregate model outputs of fuzzy rules,
- 'isr' is introduced by Zadeh and it represents the definition or assignment is not crisp, it is fuzzy.

Zadeh's fuzzy rule base has become fundamental for further works and led to development of new methods, depending on the encountered problems and shortcomings. Thereinafter, some basic and well known fuzzy inference methods are going to be introduced briefly.

2.2.2 Mamdani's Fuzzy Rule Base Structure

Mamdani's fuzzy inference method is one of the most widely used fuzzy inference method. By taking Zadeh's study as a base, Mamdani introduced the concept of fuzzy logic control. In his study Mamdani (1974) used fuzzy rule bases in order to control a steam engine and boiler combination by using a set of linguistic rules supplied from experienced human operators.

The format of his fuzzy rules is as follows; "If; x_1 is $A_1$and $x_2$is $A_2$and... and $x_n$is A_n then y is B, where $A_1, A_2, \dots, A_n$and B are fuzzy sets. The consequence of implication is a fuzzy set"(Leondes, 1998, p. 63). The mathematical notation and the general structure of Mamdani's fuzzy rule base are respectively given in equation 2.5 and in Figure 2.2.

$$\mathcal{R} : \underset{i=1}{\overset{c}{ALSO}} \left[IF \underset{j=1}{\overset{nv}{AND}} \left(x_j \in X_j \, is A_{ij} \right) THEN y_i \text{ } is \text{ } b_i \right] \tag{2.5}$$

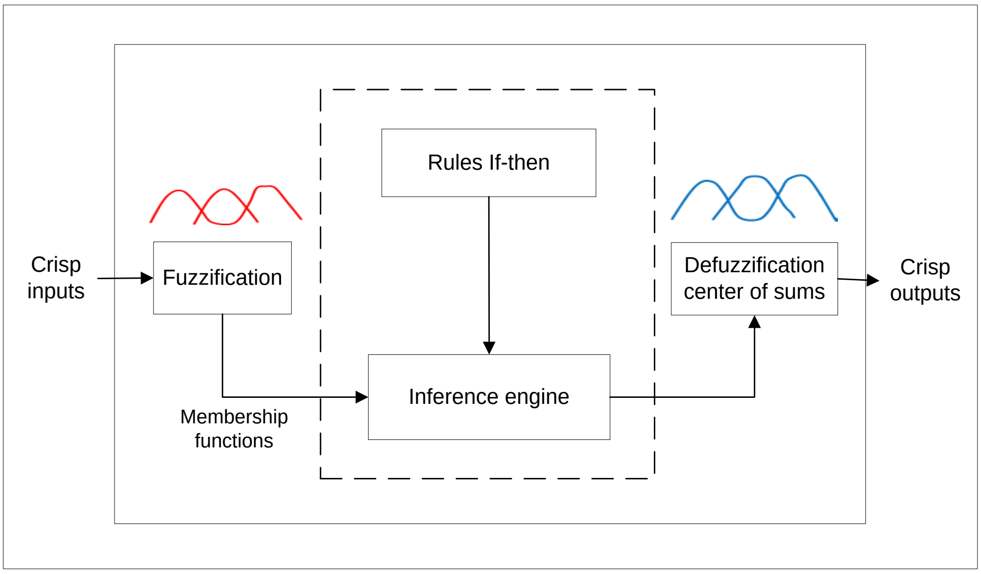


Figure 2.2 Takagi-Sugeno fuzzy rule base (Ponce-Cruz and Ramirez-Figueroa, 2010)

Mamdani type fuzzy rule based systems provide a highly flexible means to formulate knowledge, but although Mamdani fuzzy rule based systems possess several advantages, still they have some drawbacks. As it mentioned in the study of Cordon (2011) one of the main pitfalls of Mamdani's fuzzy rule base is the lack of accuracy when complex and high-dimensional systems are modeled and this is stemmed from the inflexibility of the linguistic variables, which imposes hard restrictions to the fuzzy rule structure.

Cordon, Herrera and Zwir (2001) also stated the deficiency of Mamdani fuzzy rule base as follows: "The lack of accuracy of Mamdani type models is due to some problems related to the linguistic rule structure considered, which is a consequence of the inflexibility of the concept of linguistic variables" (p. 63).

2.2.3 Mizumoto Fuzzy Rule Base Structure

Mizumoto fuzzy rule base differs from Zadeh's fuzzy rule base, with its consequence part, it could be said that, it is a simplified version of Zadeh rule base. In Mizumoto rule base, instead of a fuzzy set scalarB_i, each consequence of rules represented with a scalarb_i. Mizumoto fuzzy rule base is represented as follows;

$$\mathcal{R}: \underset{i=1}{\overset{c}{ALSO}} \left[IF \underset{j=1}{\overset{nv}{AND}} \left(x_j \in X_j isr A_{ij} \right) THEN y_i = \; b_i \right] \tag{2.6}$$

In the equation AND, THEN, ALSO are connectives, *c* represents the number of rules.

2.2.4 Takagi-Sugeno-Kang (TSK) Fuzzy Rule Base Structure

Takagi and Sugeno modified the consequence of Mamdani rule base structure and applied their proposed rule base to parking control of a model car. The format of their fuzzy rules is; If ; $\mathrm{x_1}$ is $\mathrm{A_1}$ and $\mathrm{x_2}$ is $\mathrm{A_2}$ and... and $\mathrm{x_n}$ is $\mathrm{A_n}$ then $\mathrm{y} = (\mathrm{a_0} + \mathrm{a_1 x_1} + \cdots + \mathrm{a_n x_n})$.

As stated by Kaur and Kaur (2012) in their study, contrary to Mamdani fuzzy rule bases TSK fuzzy rule base is computationally more efficient and gives better results with optimization and adaptive techniques which enables to model the data more appropriately.

Kaur and Kaur (2012) explain the difference between Mamdani and TSK fuzzy rule base as follows; “Mamdani-type FIS and Sugeno-type FIS is the way the crisp output is generated from the fuzzy inputs. While Mamdani-type FIS uses the technique of defuzzification of a fuzzy output, Sugeno-type FIS uses weighted average to compute the crisp output” (p. 323).

As it could be seen from the Figure 2.3 the difference between Takagi-Sugeno and Mamdani fuzzy rule bases is that, the outputs of the rule bases are not defined by membership functions; they are defined with non-fuzzy analytical functions.

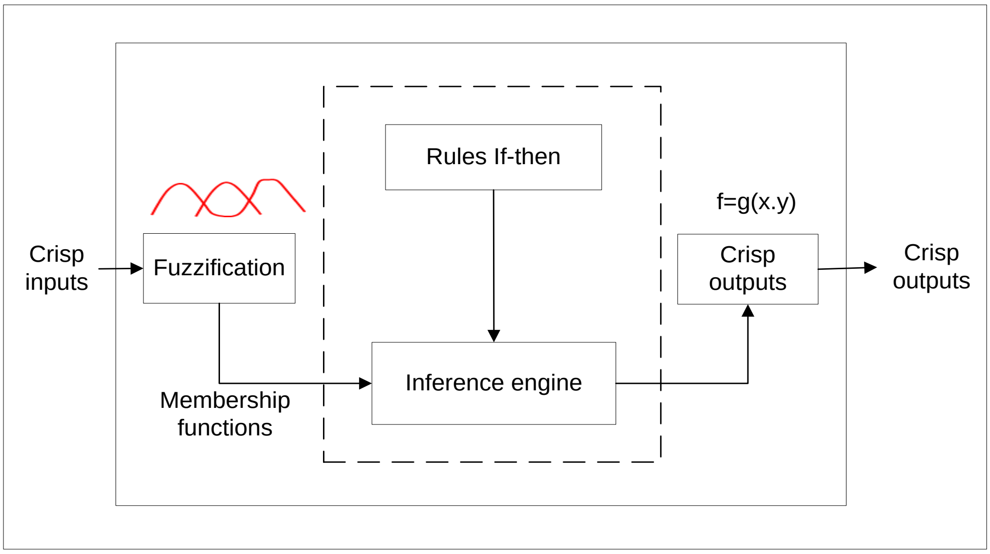


Figure 2.3 Takagi-Sugeno fuzzy rule base (Ponce-Cruz and Ramirez-Figueroa, 2010)

As Mizumoto rule base structure, TSK is differ from Zadeh's rule bases with its consequent part. Consequent part of TSK fuzzy rule base structure is expressed with a function of input variables. Fuzzy rule base structure ofTSK can be given as follows;

$$\mathcal{R} : \underset{i\,=\,1}{\overset{c}{ALSO}} \left[IF \underset{j\,=\,1}{\overset{nv}{AND}} \left(x_j \in X_j\, isr A_{ij} \right) THEN y_i = \; a_i x^T + b_i \right] \qquad (2.7)$$

- a_i and b_i are regression line coefficients associated with *ith* rule,
- y_i is the model output of ith rule,
- THEN is the connective, which weights y_i for each rule by using corresponding degree of firing of a given observation in order to find the model output from each rule,
- ALSO is the connective, which takes the weighted average of the model output of each rule in order aggregate the model outputs of fuzzy rules (Çelikyılmaz and Türkşen, 2009b, p. 39).

2.3 Advantages and Disadvantages of Fuzzy Rule Bases

Despite the wide range of application areas, fuzzy rule bases still have some disadvantages. Constructing a rule base is generally difficult and time consuming besides the need of expert knowledge, due to containing linguistic variables and need to know the system very well. Another substantial disadvantage of fuzzy rule bases is the increasing number of parameters and therefore the increasing complexity of fuzzy rule bases while the systems are being larger. If the system that is going to be studied has a large number of parameters, it will be so hard to build up an inference system and decide which parameters are going to be used such as t-norms, co-norms.In their study Siary and Guely (1998) also mentioned some basic disadvantages of fuzzy rule bases when the knowledge does not exist and parameters take time and no consistent methodology exist.

> In order to increase the efficiency of fuzzy rule-based systems with multiple variables, it is necessary to reduce bigger fuzzy rule bases into smaller fuzzy rule bases while keeping the essential fuzzy rules in the rule bases. However, reducing fuzzy rule bases will cause sparse fuzzy rule bases which contain blank areas uncovered by fuzzy rules in the universe of discourse while conventional fuzzy inference methods only can handle complete fuzzy rule bases (Chang and Chen, 2009, p. 3444).

In order to eliminate these deficiencies, by integrating fuzzy rule bases with other techniques such as genetic algorithms, neural networks and etc. many different approaches are proposed. Based on the fuzzy rule base systems and its disadvantages, one of these proposed approaches is fuzzy functions approach which is suggested by Türkşen and combines Least Square Estimation (LSE) with fuzzy membership values.

2.4 Conclusion

As it could be understood from all aforementioned expressions, fuzzy rule bases have a great importance and have provided great convenience after they have been proposed by Zadeh (1973) and then have become widely known. Fuzzy rule base system applied to a variety of fields successfully and provided to be able to obtain very good results. But despite their all benefits, they have many substantial limitations. Türkşen and Çelikyılmaz have proposed fuzzy functions concept in order to eliminate these insufficiencies.

The fundamental theory of Türkşen's fuzzy functions concept is explained in chapter 4, after the theory of fuzzy clustering, which forms the cornerstone of fuzzy functions, and the basic types of clustering algorithms are reviewed in the next chapter.

CHAPTER THREE
A BRIEF OVERVIEW OF FUZZY CLUSTERING AND CLUSTER VALIDITY MEASURES

3.1 Introduction

Clustering could be defined as dividing predefined data elements into a number of subgroups according to their similarities or dissimilarities. In other words a data set is split into different groups where each element of a group shows a degree of closeness and similarity. For grouping into classes, different measures are used according to the data and the aim of clustering. Palit and Popovic (2005) expressed that “clusters are usually defined as groups of objects mutually more similar within the same groups than with the members of other clusters, whereby the term ‘similarity’ should be understood as mathematical similarity, measured in some well-defined sense” (p. 174).

> The objective of most clustering methods is to provide useful information by grouping (unlabeled) data in clusters; within each cluster the data exhibits similarity. Similarity is defined by a distance measure, and global objective functional or regional graph-theoretic criteria are optimized to find the optimal partitions of data. The partitions generated by a clustering approach define for all data elements to which class (cluster) they belong (Rezaee et al., 1998, p. 237).

Clustering has been a very important way of data analysis and has been subjected to many researches. In order to improve the efficiency of existing clustering algorithms, researchers are studying on new approaches which integrate clustering algorithms with different methodologies.

In the following sections, some well-known clustering methods and their basic properties are going to be introduced and compared with each other.

3.2 Basic Types of Clustering Algorithms

Clustering methods have been widely applied in various areas such as taxonomy, geology, business, engineering systems, medicine and image processing etc. The objective of clustering is to find the data structure and also partition the data set into groups with similar individuals. These clustering methods may be heuristic, hierarchical and objective-function-based etc. (Yang, Hwang and Chen, 2004, p. 301).

To classify clustering algorithms, in a general manner, clustering could be divided c-partitions of data as hard (or crisp) and soft (or fuzzy) clustering as Ross (2004) classified in his study. In the next sections, hard clustering, fuzzy c-means clustering and Gustafson-Kessel clustering algorithms are introduced briefly.

3.2.1 Hard Clustering

In classical set theory, when elements are grouped, they are split into clusters according to whether they belong to a cluster or not. If an element belongs to a cluster it is represented with “1” if it doesnot belong to a cluster it is represented with “0”. Furthermore an element can be a member of only one cluster, cannot be a member of a different cluster at the same time. In the literature this is called as hard clustering.

A hard partition can be considered as a group of subsets formulated in terms of classical sets. The objective of hard clustering is to partition the given data set; $X = \{x_1,\ x_2, \dots, x_n\}$ into c clusters.

Let we define a family of $\{A_i,\ i = 1, .., c\}$ as a hard partition of X, the following forms apply to these partitions:

$$\bigcup_{i=1}^{c} A_i = X \qquad 2 \le c < n \tag{3.1}$$

$$A_i \cap A_j = \emptyset \qquad all\ i \neq j \tag{3.2}$$

$$\emptyset \subset A_i \cap X \qquad all \tag{3.3}$$

$$U = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \mu_{c1} & \mu_{c2} & \cdots & \mu_{cn} \end{bmatrix} \tag{3.4}$$

The above equations that the elements of the U partition matrix must satisfy the following conditions:

$$\mu_{ik} \in \{0,1\}, \qquad 1 \leq i \leq c;\ 1 \leq k \leq n \tag{3.5}$$

$$\sum_{i=1}^{c} \mu_{ik} = 1, \quad 1 \leq k \leq n \tag{3.6}$$

$$0 \leq \sum_{k=1}^{n} \mu_{ik} < n, \quad 1 \leq i \leq c \tag{3.7}$$

The discrete nature of hard partitioning causes difficulties with algorithms based on analytic functionals, since these functional are not differentiable. Clustering algorithms may use an objective function to measure the desirability of partitions. Nonlinear optimization algorithms are used to search for local optima of the objective function. The concept of fuzzy partition is essential for cluster analysis, and consequently also for the identification techniques based on fuzzy clustering (Palit and Popovic, 2005, p. 175).

3.2.2 Fuzzy C- Means Clustering Algorithm

Contrary to hard clustering, in fuzzy clustering data elements do not have to belong only one cluster. Each element can belong to a cluster with different membership degrees and these membership degrees indicate the strength of relationship between the element and cluster.

> Bezdek, Ehrlich and Full (1984) explained the fuzz clustering as follows; the key to Zadeh's idea is to represent the similarity a point shares with each cluster with a function (termed the membership function) whose values (called memberships) are between zero and one. Each sample will have a membership in every cluster; memberships close to unity signify a high degree of similarity between the sample and a cluster while memberships close to zero imply little similarity between the sample and that cluster (p. 191).

Fuzzy c-means clustering algorithm has proposed by Bezdek (1981) and this algorithm gives a c-partition of a dataset. According to this algorithm, each sample in the dataset represented with membership function which ranges between zero and one and the sum of the memberships for each sample must be unity. After Bezdek has proposed fuzzy c-means clustering algorithm, it has been one of the most popular clustering algorithm and paved the way for the developments of new methods. In the literature there are many different variations of fuzzy c-means algorithm.

The FCM algorithm tries to divide the elements of a dataset $X = \{x_1, \ldots, x_n\}$ into fuzzy clusters according to the some given criterions. Given a finite set of data, the algorithm returns a list of c cluster centers$C = \{c_1, \ldots, c_c\}$ and a partition matrix $U = u_{i,j} \in [0,1], \ i = 1, \ldots, n, \ j = 1, \ \ldots, c$ where each element $u_{i,j}$ tells the degree to which element x_i belongs to clusterc_j. Same as hard clustering FCM algorithm aims to minimize an objective function.

In fuzzy clustering the membership value of the kth data in the ithcluster represented as in the following notation:

$$\mu_{ik} = \mu_{A_i}(x_k) \in [0,1] \qquad (3.8)$$

In fuzzy c-means (FCM) algorithm the equation below must be satisfied;

$$\sum_{i=1}^{c} \mu_{ik} = 1 \qquad for\ all\ \ k = 1,2,\dots,n \qquad (3.9)$$

As in crisp classification, there can be no empty classes and there can be no class that contains all the data points. This qualification is manifested in the following expression:

$$0 < \sum_{i=1}^{c} \mu_{ik} < n \qquad (3.10)$$

Fuzzy c-means is based on minimization of the objective function, which is shown below (Dulyakarn and Rangsansei, 2001);

$$J_m(U,V) = \sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}^{m}\ \|X_i - V_i\|^2, \qquad 1 \le m \le \infty(3.11)$$

The *"m"* value is the degree of fuzziness and is greater than 1, u_{ij} is the membership values which represents the degree of belongingness of $X_i$to cluster i, V_i represents the cluster center and $\|*\|$ is any norm expressed the similarity between any measured data and the center.

For FCM algorithm, fuzzy partition is carried out through an iterative optimization of with the update of membership u_{ij} and the cluster centers $V_i$by;

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{ik}}\right)^{\frac{2}{m-1}}} \qquad (3.12)$$

$$V_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} X_j}{\sum_{k=1}^{c} u_{ij}^{m}} \tag{3.13}$$

FCM algorithm is iterated until the equation below is supplied. In the equation ε is a termination criterion between 0 and 1.

$$\max_{ij} \left| u_{ij}^{m} - \hat{u}_{ij}^{m} \right| < \varepsilon \tag{3.14}$$

As it mentioned before, fuzzy c-means clustering algorithm is one of the most know and used soft clustering algorithm. It has a diverse of application areas and many researchers have applied fuzzy c-means clustering algorithm successfully (Chaira, 2012; Kim, Kim, Ho and Chu, 2011; Kuo, Shih and Lee, 2004). Kuo, et al. (2004) used fuzzy c-means clustering algorithm for the automatic recognition of fabric weave patterns. Also in another study Kim et al. (2011) applied fuzzy c-means clustering method to cluster tropical cyclone tracks.

In the literature there are many different kinds of clustering methods. Some example studies on fuzzy c-means clustering and its improved versions are as follows:

- Çelikyılmaz and Türkşen (2008a) proposed a new clustering algorithm which combines the standard fuzzy clustering and regression methods.

- One of the improved versions of FCM algorithm “DifFUZZY: A fuzzy clustering algorithm for complex data sets” clustering method proposed by Cominetti et al. (2010). Cominetti et al. indicated that their clustering method is applicable to a larger class of clustering problems and can handle complex, nonlinear geometric structures in comparison to FCM clustering algorithm.

- Chaira (2012) also proposed a new approach based on fuzzy c-means to cluster pathological cell images by using different color models.

- Parker, Hall and Bezdek (2012) proposed new clustering algorithms which are some different variations of fuzz c-means clustering algorithm and proposed for the purpose of being able to cope with large datasets.

- Dagher (2012) proposed the complex fuzzy c-means algorithm (CFCM) and concluded that CFCM algorithm gave better cluster partitions.

Other new methods also have been also proposed based on fuzzy c-means clustering algorithm (Cannon, Dave and Bezdek, 1986; Hathaway and Bezdek, 2006).

FCM clustering algorithm has two important information; “c” the number of clusters and m-the order of fuzziness. It is difficult to select suitable (c*, m*) pairs because of the unsupervised behavior of FCM. There are many different validity indexes for choosing the number of clusters and the order of fuzziness for fuzzy clustering algorithms (Başkır and Türkşen, 2013, p. 930).

In section 3.3, some of the commonly used validity indexes are introduced briefly.

3.2.3 Gustafson-Kessel Clustering Algorithm

Gustafson-Kessel clustering algorithm differs from the FCM clustering algorithm. The FCM clustering algorithm is a cluster prototype with one center of gravity location, while the Gustafson-Kessel clustering algorithm is a cluster prototype of volume, each of which contains the relevant covariance matrix and center of gravity location. Hence, each data set has a sub-clustering center of gravity location and data set distribution information (Kuo , Jian, Wu and Peng, 2012, p. 580).

Hamed, Keshavarz, Dehghani and Pourghassem (2012) in their study indicated that, ”the Gustafson-Kessel algorithm (GK) extended the standard fuzzy c-means algorithm by employing an adaptive distance norm, in order to detect clusters with different geometrical shapes in one data set. Each cluster has its own norm-inducing matrix" (p. 223).

In comparison to fuzzy c-means algorithm, GK clustering algorithm needs more computation. In order to reduce calculations, the GK clustering can be performed after obtaining results from fuzzy c-means algorithm.

The GK clustering is based on iterative optimization of an objective function of the c-means type:

$$J(P; U, V, \{M_i\}) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m D_{ikA_i}^2 \qquad (3.15)$$

Given the data set P, choose the number of clusters $1 < c < N$, degree of fuzziness > 1 , the termination tolerance $\varepsilon > 0$ and the cluster volumes ρ_i. Initialize the partition matrix randomly, such that $U^{(0)} \in M_{fc}. U = [\mu_{ik}] \in [0,1]^{\propto N}$ is fuzzy partition matrix of the data. The algorithm of GK clustering algorithm is repeated for $l = 1, 2, \ldots$ as below (Hamed et al., 2012, p. 224).

Firstly cluster centers are calculated:

$$v_i{}^l = \frac{\sum_{k=1}^{N} u_{ik}^{(l-1)}\, p_k}{\sum_{k=1}^{N} \left(u_{ij}^{(l-1)}\right)^m} \,, \quad 1 \leq i \leq c \qquad (3.16)$$

Then cluster covariance matrix is calculated:

$$F_i = \frac{\sum_{k=1}^{N} \left(u_{ik}^{(l-1)}\right)^m (p_k - v_i{}^l)(p_k - v_i{}^l)^T}{\sum_{k=1}^{N} \left(u_{ij}^{(l-1)}\right)^m} \qquad (3.17)$$

Selected identity matrix is added:

$$F_i = (1-\gamma), F_i + \gamma \det(F_0)^{\left(\frac{1}{n}\right)} I \qquad (3.18)$$

Extract eigenvalues λ_{ij} and eigenvectors φ_{ij} from F_i. Find $\lambda_{imax} = max_j \lambda_{ij}$ and set: $\lambda_{ij} = \lambda_{imax} / \beta \; \forall j$ for which $\frac{\lambda_{imax}}{\lambda_{ij}} > \beta$. Reconstruct F_i by;

$$F_i = [\phi_{i1} \dots \phi_{in}] diag(\lambda_{i1}, \dots, \lambda_{in}) [\phi_{i1} \dots \phi_{in}]^{-1} \qquad (3.19)$$

Then the distance is calculated:

$$D_{ikA_i}^2 = (p_k - {v_i}^l)^T \left\lfloor \rho_i \det(F_i)^{\frac{1}{n}} F_i^{-1} \right\rfloor (p_k - {v_i}^l), \qquad 1 \le i \le c, 1 \le k \le N \quad (3.20)$$

The partition matrix is updated:

$$u_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} \left(D_{ikA_i} / D_{jkA_i} \right)^{2/(m-1)}} \qquad (3.21)$$

The production of the cluster centers and partition matrix is continued until $\left\| U^{(l)} - U^{(l-1)} \right\| \ge \varepsilon$. Otherwise GK algorithm is stopped.

3.3 Cluster Validity Measures

Validity measures are scalar indices that assess the goodness of the partition obtained. Clustering algorithms generally aim at locating well-separated and compact clusters. When the number of clusters is chosen equal to the number of groups that are actually present in the data, it is expected that the clustering algorithm will identify them correctly. When this is not the case, misclassifications appear, and the clusters are not likely to be well-separated and compact. Hence, most cluster validity measures are open to interpretation and can be formulated in different ways (Palit and Popovic, 2005, p. 181).

For fuzzy clustering, cluster validity is based on finding a fuzzy partition that fits the all data appropriately. Therefore clustering validity always tries to find the best fixes number of clusters. In the literature there are many different cluster validity measures. But as Balasko, Abonyi and Feil (2005) indicated in their study, no validation index is reliable only by itself. The optimal number of cluster should be determined by synthesizing all available measures. Also in their study they stated that less clusters are better for the optimal number of clusters.

Commonly used cluster validity indexes are represented below. Before representing validity indexes, general parameters which are used in validity indexes are introduced below.

- "c" is the number of cluster,
- "n" is the number of data vectors,
- "μ" represents the membership values,
- "v_{i}" is center points of ith cluster
- "m" is degree of fuzziness,
- "n_i" is the number of element in ith dimension,
- "c_i" ith cluster
- $||c_i||$ number of element in ith cluster
- $d(x,y)$ distance between two data element

- **Partition coefficient (PC):** It is defined by Bezdek, and measures the amount of overlapping clusters. For partition index, the maximum value means the optimum value.

$$PC(c) = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n}(\mu_{ij})^2 (3.22)$$

- **Classification entropy (CE):**It measures the fuzziness of the cluster partition. For classification entropy the minimum value is the optimum value.

$$CE(c) = -\frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}\log(\mu_{ij}) \quad (3.23)$$

- **Partition index (SC):** is the ratio of the sum of compactness and separation of the clusters. The lower value of partition index represents a better partition.

$$SC(c) = \sum_{i=1}^{c}\frac{\sum_{j=1}^{n}(\mu_{ij})^m\left\|x_j - v_i\right\|^2}{n_i\sum_{k=1}^{c}\|v_k - v_i\|^2} \qquad (3.24)$$

- **Separation index (S):** The separation index uses the minimum-distance separation for partition validity. The minimum value gives the best partition.

$$S(c) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n}(\mu_{ij})^2\left\|x_j - v_i\right\|^2}{nmin_{i,k}\left\|x_j - v_i\right\|^2} \qquad (3.25)$$

- **Xie and Beni's index (XB)**: XB index quantifies the ratio of the total variation within clusters and the separation of clusters. The minimum value gives the optimum number of clusters**.**

$$S(c) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n}(\mu_{ij})^m\left\|x_j - v_i\right\|^2}{nmin_{i,j}\left\|x_j - v_i\right\|^2} \qquad (3.26)$$

- **Dunn's index (DI):** The maximum value of Dunn index gives the optimum number of clusters.

$$DI(c) = min_{i\in c}\left\{min_{j\in c, i\neq j}\left\{\frac{min_{x\in C_i,\ y\in C_j} d(x,y)}{max_{k\in c}\{max_{x,y\in C} d(x,y)\}}\right\}\right\} \tag{3.27}$$

- **Davies–Bouldin index (DB): "**This is probably one of the most used indices in CVI comparison studies. It estimates the cohesion based on the distance from the points in a cluster to its centroid and the separation based on the distance between centroids" (Arbelaitz, Gurrutxaga, Muguerza, Perez and Perona, 2013, p. 245).

The Davies-Bouldin Validation Indice (DB) represents the ratio of the total within-cluster scatter to between-cluster separation. The scatter,S_i, within the i th cluster, is computed as (Sato, Suzuki and Mabuchi, 2007);

$$S_i = \frac{1}{\|c_i\|}\sum_{x\in c_i} d(x, v_i) \tag{3.28}$$

Where c_i is the set of data points in the ith cluster, $\|c_i\|$ is the number of data points in ith cluster and v_i is the cluster center point of ith cluster. The centroid distance, d_{ij} is;

$$d_{ij} = \|v_i - v_j\| \tag{3.29}$$

Thus Davies-Bouldin index is defined as where $i, j = 1, \dots, c$;

$$DB(c) = \frac{1}{c}\sum_{i=1}^{c} \max_{i,j\neq i} \frac{S_i + S_j}{d_{ij}} \tag{3.30}$$

The minimum value of Davies-Bouldin index gives the optimum number of clusters.

- **Kim Index (KI):** In cluster validity index, the relative degree of sharing of two fuzzy clusters is defined as the weighted sum of the relative degrees of sharing for all data (Zhang and Qian, 2012).

$$Kim(c) = \frac{2}{c-(-1)} \sum_{p \neq q}^{c} \sum_{j}^{n} \left[c \times \min\left(u\mathrm{F}_p(x_j), u\mathrm{F}_q(x_j) \right) \times h(x_j) \right] \qquad (3.31)$$

Where $h(x_j) = -\sum_{i=1}^{c} u\mathrm{F}_i(x_j) \log_a u\mathrm{F}_i(x_j)$, F_p and F_q are be two fuzzy clusters belonging to a fuzzy partition (U, V) and c is the number of clusters. The minimum value of Kim index, gives the best optimum number of clusters.

Arbelaitz et al. (2013) compared 30 cluster validity indexes in an experimental setting. More information on cluster validity indexes could be found out in their study.

3.4 Conclusion

As it was emphasized in previous sections, clustering concept is one of the cornerstones of Türkşen's fuzzy functions approach. Clustering is also crucial in the proposed "fuzzy functions with genetic programming" approach as it is based on the Tükşen's fuzzy functions concept.

As it was emphasized before the novelty of Türkşen's fuzzy functions approach is that membership values and some of their user predefined transformations are added as new variables to original input variables of the dataset. Therefore fuzzy clustering forms an important part of fuzzy functions. For this reason in this chapter, the concept of clustering and basic types of clustering algorithms areintroduced.

In order to find out membership values of the datasets, fuzzy c-means (FCM) clustering algorithm ischosen to be used in the present study. To find out the optimal number of clusters for the application phase of fuzzy functions with LSE and fuzzy functions with GP, partition coefficient (PC), classification entropy (CE), partition index (SC), separation index (S), Xie and Beni’s index (XB),Dunn’s index (DI) and alternative Dunn index (ADI) are used. These validity indexes are realized via fuzzy clustering toolbox which is prepared by Balasko, Abonyi and Feil(2005) in Matlab.

In the next chapter, firstly fuzzy functions concept and its algorithm is introduced, afterwards a small artificial data is generated and the algorithm is explained with this dataset step by step for enabling a better understanding of the concepts.

CHAPTER FOUR
FUZZY SYSTEM MODELING BY TURKSEN'S FUZZY FUNCTIONS APPROACH

4.1Introduction

In the literature, there have been many different definitions of "fuzzy functions" concept. Probably the most known definition of "fuzzy functions" is the one which represents the membership functions. Another implied meaning of fuzzy functions is the mathematical definition which is coined by Demirci (1999). The fuzzy functions term which used in this study was introduced by Türkşen in 2004 and it is not same with fuzzy function term used by Demirci (1999). However as also stated by Çelikyılmaz and Türkşen (2009a, 2009b) the fuzzy functions term used by Demirci (1999) underlines the mathematical basis of Türkşen's fuzzy functions concept.

Fuzzy rule bases which are overviewed in the previous chapter were used successfully for modeling many problems. Although its success in many problems, fuzzy rule bases still have some difficulties. In fuzzy rule bases there are several parameters to be identified such as "number of fuzzy rules", "type of fuzzy operators" that affect the performance of the fuzzy rule bases. In a sense this means that fuzzy inference system which is based on fuzzy rule bases involves subjectivity and requires expert knowledge. Many researchers have pointed out the difficulty of fuzzy rule bases, when it is not easy to access the knowledge and the dimensions of the system changes (Siary and Guely, 1998). It is clear that systems are generally complex and this poses an obstacle for correct identification of the systems and therefore modeling them properly. In this respect, applying fuzzy rule bases to real problems can become more difficult. For this reason, Türkşen in 2004 has proposed fuzzy functions as an alternative to fuzzy rule bases. Fuzzy functions approach does not require "expert knowledge" and fuzzy set operators such as "fuzzification", "difuzzification", "t-norms", "co-norms" etc. Therefore, these properties provide fuzzy functions to be implemented more easily for several problem types.

After Türkşen’s introduction of fuzzy functions approach, Çelikyılmaz and Türkşen (Çelikyılmaz and Türkşen, 2007a and 2007b; Türkşen and Çelikyılmaz, 2006;) have also made improvements by combining fuzzy functions with several other soft computing techniques like metaheuristics.

4.2The Concept of Fuzzy Functions

> Türkşen (2012) described fuzzy functions concept as an approach where a classical regression is enhanced by the introduction of membership values and their transformations to improve the regression constant, and hence the introduction of fuzzy functions in place of fuzzy rule bases where a fuzzy clustering algorithm such as FCM or IFC is used to determine the number of such fuzzy regressions required for an affective solution (p. 348).

As it was stated before in fuzzy functions approach, instead of representing a system with IF-THEN rules or similar linguistic expressions, a system is represented with fuzzy functions. In their excellent study Çelikyılmaz and Türkşen (2009b) indicated that in fuzzy functions approach depending on the complexity of the system, each vector could be represented with different methods such as least square estimation (LSE) or support vector machines (SVM).

> One can build models for various system structures as with the other fuzzy system modeling tools by making use of fuzzy functions approach. The goal of the general system modeling depends on the type of the system under study. If the aim is to assign class labels to objects, such as in classification problems, the goal of the system modeling is to reduce the number of misclassified cases. On the other hand, if the problem involves estimation of a relationship between given independent variables and the dependent variable by using functions, then the goal of a system modeling is to find a representation function that can minimize the prediction error (Çelikyılmaz and Türkşen 2009b, p. 106).

The novel feature of fuzzy functions is that the membership values and some of their proper transformations obtained from fuzzy clustering algorithms (i.e. fuzzy c-means (FCM) clustering algorithm or Gustafson-Kessel clustering algorithm) can also be added to the original data matrix in order to explain the relationship between input and output values better. Türkşen propound that, using membership values and their transformations as additional variables will enable to identify the structure of the given data more easily.

Çelikyılmaz (2005)in her thesis, applied fuzzy functions with LSE and fuzzy functions with SVM to two datasets and compared the results of both model. Türkşen and Çelikyılmaz also used different methodologies with fuzzy functions and other fuzzy inference methods and compared them in order to evaluate the performance of fuzzy functions (Çelikyılmaz and Türkşen, 2008a, 2008b; Türkşen and Çelikyılmaz, 2006). Also in his study, Türkşen (2011) studied Type-1 Fuzzy Functions (FF) and Improved Fuzzy Functions (IFF) in which improved fuzzy clustering algorithm was used and results were compared.

Generally, modeling a system is composed of three phases; “training”, “validation” and “testing” phases. Structure identification of the model constitutes the training phase. General structure of the system and the parameters which represent the system ideally are tried to be found out with training dataset. Training dataset comprise a large part of the system. The modeling performance of the system which is modeled according to parameters found out during the training algorithm is trying to be measured with testing dataset. These processes are repeated several times in order to calculate general performance of the system.

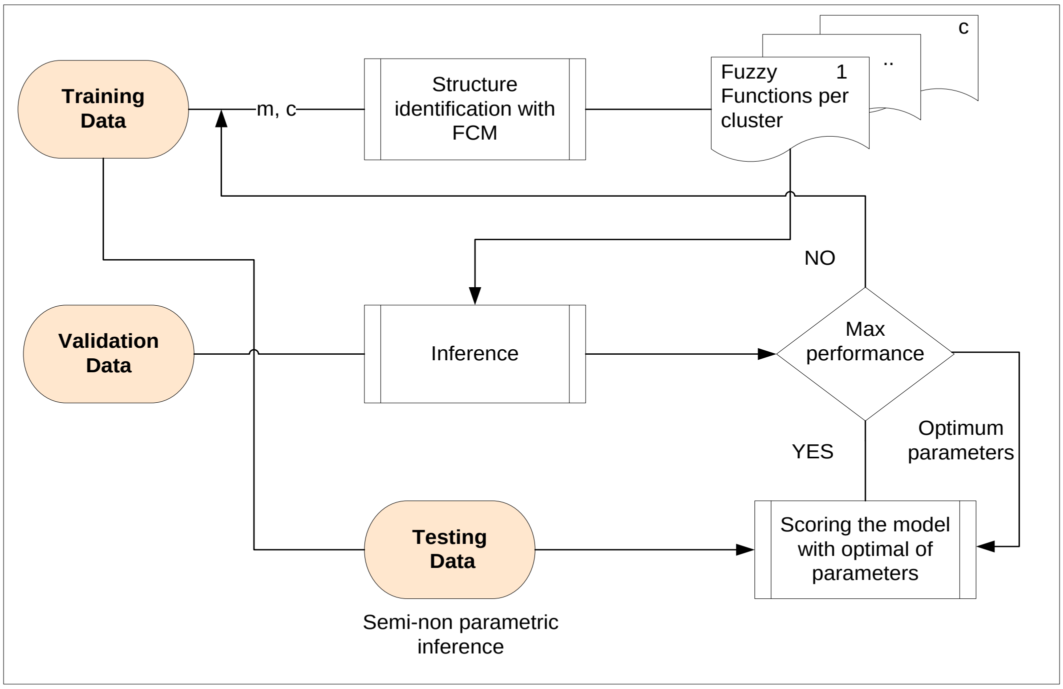

Training Data
m, c
Structure identification with FCM
Fuzzy 1 Functions per cluster
..
c
NO
Validation Data
Inference
Max performance
YES
Optimum parameters
Testing Data
Scoring the model with optimal of parameters
Semi-non parametric inference

Figure 4.1 General structure of fuzzy functions (Çelikyılmaz and Türkşen, 2009b)

4.2.1Type-1 Fuzzy Function Approach with Least Square Estimation (T1FF)

In the first step of the fuzzy functions approach, the data which is going to be searched is firstly clustered into overlapping clusters. FCM clustering algorithm is one of the most commonly used clustering technique and the degree of overlapping clusters is represented with “m”. In order to obtain membership values that represent the degree of belongingness to each cluster, fuzzy c-means (FCM) clustering algorithm is decided to be used also in this study. Then the membership degrees of the observations for each cluster have to be found out. As it can be understood clearly, the membership values play a key role for fuzzy functions approach. Finding the best descriptive membership degrees directly related to finding the most appropriate number of clusters. In the next section more detailed information on the structure identification of fuzzy functions and the inference mechanism is given.

4.2.1.1 *Structure Identification of Fuzzy Functions with LSE*

Let $Z(x,y) = \{(x_1,y_1),\ (x_2,y_2), \dots, (x_n,y_n)\}$, represents the input-output space, where $z(x_k,y_k) \subset \Re^{nv+1}$ denotes any data vector from training set and every data point is composed of $(nv+1)$ dimensions of input vectors, $x_k = \left(x_{1,k}, \dots, x_{nv,k}\right) \in \Re^{nv},\ k = 1, \dots, n,$ a total of n vectors, and an output $y_k \in \Re^{nv}$(Çelikyılmaz and Türkşen, 2009b, p. 114). Here Z represents the input-output matrix. "nv" is the number of variables.

Before applying the fuzzy functions approach, some parameters are defined and FCM algorithm is applied. In the FCM clustering algorithm, "i" is used to symbolize "c" which represents the total number of clusters. "n" represents the number of data vectors and "m" represents the degree of fuzziness which means "degree of overlapping clusters" and it is greater than 1. To indicate the related matrixes, let assume that there is a multi-input single output (MISO) dataset and X represents the input matrix; the mathematical notation of input matrix is shown below;

$$X = \begin{bmatrix} x_{1,1}\, x_{1,2} & & x_{1,nv} \\ \vdots \quad \vdots & \dots & \vdots \\ x_{n,1} x_{n,2} & & x_{n,nv} \end{bmatrix} (4.1)$$

Let Y represents the output matrix; the output matrix is shown as follows;

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ \vdots \\ \vdots \\ y_n \end{bmatrix} (4.2)$$

$\mu_{ki} \in [0,1]$ represents the membership degrees of the kth data in cluster "i". The matrix of the membership degrees of all data for each cluster is shown as below;

$$U = \begin{bmatrix} \mu_{1,1}\, \mu_{1,2} \cdots \mu_{1,c} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ \mu_{n,1} \mu_{n,2} \cdots \mu_{n,c} \end{bmatrix} (4.3)$$

“nm” is the dimension of augmented matrix (membership values and their transformations) that is added to the original data matrix. To give an example, we assume that there is a dataset composed of multi input single output and only membership values are selected to be added. So nm is equal to 1 (nm=1) and the new matrix is shown in equation 4.4. The abnormalities generated by the clustering algorithms could be eliminated with an α-cut.

$$\Phi_i(x,\mu_i) \in \Re^{nv+1} = \begin{bmatrix} \mu_{k,1} x_{1\times1} & \cdots & x_{1\times nv} \\ \vdots \quad \vdots & \ddots & \vdots \\ \mu_{k,i} x_{k\times1} & \cdots & x_{k\times nv} \end{bmatrix} \quad \begin{array}{l} \mu_{k,i} > \alpha - cut \\ 0 < k \leq n \\ i = 1,\ldots,c \end{array} \qquad (4.4)$$

As it was mentioned before the novelty of the fuzzy functions is that membership values and their transformations are added to the original data matrix as additional dimensions. The final matrix which is composed of the original data, membership values and some of their transformations is shown by equation 4.5;

$$\Phi(x,\mu_1) = \begin{bmatrix} \mu_{1,i} & exp(\mu_{1,i}) & (\mu_{1,i})^p & x_{1\times2} & \cdots & x_{1\times nv} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{n,i} & exp(\mu_{n,i}) & (\mu_{n,i})^p & x_{n\times2} & \cdots & x_{n\times nv} \end{bmatrix} (4.5)$$

In fuzzy functions approach in order to explain the relationship between variables, some kind of statistical methods such as least square estimation (LSE) or support vector machines (SVM) can be used according to the complexity of the datasets. In this study, fuzzy functions approach with least square estimation (LSE) is going to be introduced and it is used for all example case studies.

Çelikyılmaz and Türkşen (2009b) have described the training algorithm for fuzzy functions as follows;

Step 1: Firstly the parameters of the FCM clustering algorithm are decided;

- m≥1.1 (degree of fuzziness),
- c>1 (the number of clusters),
- ε (a termination threshold).

Step 2: Execute FCM clustering algorithm to find cluster centers $v_i(xy)$of the dataset$Z(x, y)$.

$$\underset{\substack{1\leq i\leq c\\ 1\leq k\leq n}}{\forall} \mu_{ki}(xy) = \left(\sum_{j=1}^{c}\left(\frac{d_{ki}(xy)}{d_{kj}(xy)}\right)^{\frac{2}{m-1}}\right)^{-1} \quad d_{ki}^{xy} = \|(x_k, y_k) - v_i(x, y)\| \quad (4.6)$$

Step 3: Membership values are found out according to equation in (4.7);

$$\underset{\substack{1\leq i\leq c\\ 1\leq k\leq n}}{\forall} \mu_{ki}(x) = \left(\sum_{j=1}^{c}\left(\frac{d_{ki}(x)}{d_{kj}(x)}\right)^{2/(m-1)}\right)^{-1} \quad , \; where d_{ki}(x) = \|x_k - v_i(x)\| \quad (4.7)$$

Step 4: Membership values of each input data sample, μ_{ki}, their transformations and identity matrix are augmented to the original input matrix as shown by equation (4.8) for each cluster “i”.

$$\Phi_i = \begin{bmatrix} 1 & \mu_{1,i} & exp(\mu_{1,i}) & (\mu_{1,i})^p & x_{1\times 1} & \cdots & x_{1\times nv} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \mu_{n,i} & exp(\mu_{n,i}) & (\mu_{n,i})^p & x_{n\times 1} & \cdots & x_{n\times nv} \end{bmatrix} (4.8)$$

Step 5: Regression coefficient parameters are calculated for each cluster “i” by executing the equation (4.9).

$$\beta_i = ({\Phi_i}^T \Phi_i)^{-1} ({\Phi_i}^T Y_i) \qquad (4.9)$$

As it was stated in the algorithm above, firstly FCM clustering parameters; m, c, and ε are chosen. Then applying FCM clustering algorithm, membership values are obtained. In step 4, membership values (μ_{ki}) and their transformations are augmented into the original data matrix as new dimensions of the original dataset.

In the algorithm as it wasdepicted above, the last step means that one regression function $f(\Phi_i, \beta_i)$ is identified for each cluster. Original input matrix could be mapped onto higher dimensions by using transformations of membership values. In order to get more appropriate or accurate results, Çelikyılmaz and Türkşen (2009b) proposed to use mathematical transformation of membership values such as $(\mu_{ki})^{2}$, $\mu_{ki}{}^{m}$, $\exp(\mu_{ki})$, $\ln(1-(\mu_{ki})/(\mu_{ki})$.

4.2.1.2 *Inference Mechanism of Fuzzy Functions with LSE (T1FF)*

Let the validation data be represented with $X^{v} = \{x_1^{v},\ x_2^{v}, \dots, x_{ndv}^{v}\}$ every kth data vector contain input vectors of dimension of nv, $X_k^{v} = (x_{1,k}^{v}, \dots, x_{nv,k}^{v})$ and an output $y_v^{k} \in \Re$. Here X^{v} represents the input matrix of $(ndv \times nv)$, ndv is the number of validation vectors, c is the number of clusters, m is the degree of fuzziness and i $(i = 1, \dots, c)$ is the cluster identifier. Same as in the validation data, testing data is represented with $X^{test} = \{x_1^{test}, x_2^{test}, \dots, x_{nte}^{test}\}$. In every vector (observation) contains a nv dimensional vector of $X_k^{test} = (x_{1,k}^{test}, \dots, x_{nv,k}^{test}) \in \Re^{nv}$ and an output variable$y_v^{test} \in \Re$. X^{test} is the input matrix, nte is the number of testing vectors.

The algorithm for the inference mechanism of fuzzy functions is described as follows (Çelikyılmaz and Türkşen, 2009b);

Step 1: Membership values of each validation sample, x_k^{v}, $k = 1, \dots, ndv$ are found out by using the equation (4.10);

$$\underset{\substack{1 \le i \le c \\ 1 \le k \le ndv}}{\forall} \mu_{ki}^{v} = \left(\left(\sum_{j=1}^{c} \frac{d_{ki}^{v}}{d_{kj}^{v}} \right)^{\frac{2}{m-1}} \right)^{-1} \quad i = 1, \dots, c, where d_{ki}^{v} = \| x_k^{v} - v_i(x) \| \qquad (4.10)$$

Step 2: Membership values of validation data, μ_{ki}^{v}, their transformations and identity matrix are added to original validation data $x^{v} \to \Phi_{\mathrm{i}}(x^{v} | \mu_i^{v})$, in $\Re^{nv+nm}$ space.

$$\Phi_i{}^v = \begin{bmatrix} 1\mu_{1,i}^v & exp(\mu_{1,i}^v) & (\mu_{1,i}^v)^p & x^v{}_{1\times 1} & \cdots & x^v{}_{1\times nv} \\ \vdots\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1\mu_{n,i}^v & exp(\mu_{n,i}^v) & (\mu_{1,i}^v)^p & x^v{}_{n\times 1} & \cdots & x^v{}_{n\times nv} \end{bmatrix} (4.11)$$

Step 3: Then by using equation (4.12)output values are calculated.

$$\hat{y}_{k,i} = \Phi_{k,i}\beta_i \qquad (4.12)$$

Step 4: Finally single output value for validation data samples are calculated by weighting inferred fuzzy output values from each cluster with their corresponding membership values.

$$\hat{y}_k = \frac{\sum_i^c \hat{y}_{ki}\mu_{ki}}{\sum_i^c \mu_{ki}} \quad i = 1, \dots, c, \qquad k = 1, \dots, ndv \qquad (4.13)$$

In the algorithm as stated above, firstly membership values according to fuzzy-c means clustering (FCM) algorithm are calculated. Then these membership values and their transformations (same as in training algorithm) are added to the original validation data matrix as additional dimensions. Then (same as the training algorithm with this new matrix) fuzzy functions are defined for each observation. Afterwards, the predictedoutput values of the data vectors are found by multiplying coefficients matrix which is found in the training algorithm and this new matrix. When this stage is finished "c" numbers of output matrixes are found for each observation. Finally for each data sample a single output value is found by using the equation (4.13),by multiplying the output values with their corresponding membership values.

In order to enable much better understanding of the fuzzy functions approach a hypothetical example with all necessary computational steps is shown in the next sub-section.

4.3An Illustrative Example for Fuzzy Functions with LSE

In order to ensure that the concept of Türkşen’s fuzzy functions approach is understood more easily, the algorithm is explained with a numerical example. For this purpose a small artificial dataset is generated which is consisting of 3variables and 10 observations. The dataset is represented in Table 4.1.

Table 4.1 Input and output variables of generated artificial dataset

Observations	Variable1	Variable2	Variable3	Outputs
1. observation	15.00	56.00	10.33	58.77
2. observation	14.30	55.00	12.43	58.93
3. observation	9.98	8.60	50.00	120.40
4. observation	9.56	7.90	51.20	122.00
5. observation	10.12	30.10	49.80	123.50
6. observation	11.00	29.90	50.44	120.18
7. observation	8.77	7.80	51.87	131.11
8. observation	23.80	86.50	45.87	75.00
9. observation	26.23	89.00	44.90	73.20
10. observation	24.76	85.40	43.12	76.00

Firstly the dataset is divided into two parts randomly in Matlab as training and validation phases in order to implement fuzzy functions algorithm. Training data set constitutes the seventy percent of all data and remained observations of the data constitute the validation data which is thirty percent of all data. Thus there are 7 observations for training data and 3 observations for validation data. Training and validation datasets which are randomly selected in Matlab are shown respectively in Table 4.2 and Table 4.3.

Table 4.2 Input and output variables of training dataset

Observations	Variable1	Variable2	Variable3	Outputs
2. observation	14.30	55.00	12.43	58.93
4. observation	9.56	7.90	51.20	122.00
6. observation	11.00	29.90	50.44	120.18
7. observation	8.77	7.80	51.87	131.11
8. observation	23.80	86.50	45.87	75.00
9. observation	26.23	89.00	44.90	73.20
10. observation	24.76	85.40	43.12	76.00

Table 4.3 Input and output variables of validation data

Observations	Variable1	Variable2	Variable3	Outputs
1. observation	15.00	56.00	10.33	58.77
3. observation	9.98	8.60	50.00	120.4
5. observation	10.12	30.10	49.80	123.5

After training and validation datasets are introduced, the algorithm is applied step by step.

Step 1: Firstly "*c*" the optimum number of cluster should be found out and degree of fuzziness should be decided. In order to find out the best partition "fuzzy clustering toolbox" which was prepared in Matlab by Balasko, Abonyi and Feil (2005) is used. For the artificial dataset the best partition is found as 3.

Step 2: In this step, according to the optimum number of cluster, the membership valuesare found out for training and validation data with FCM algorithm. In Table 4.4 and Table 4.5 membership values of training and validation data are shown respectively.

Table 4.4 Membership values of training data

Membership Values of Training Data			
Observations of dataset	**Cluster 1** **i=1**	**Cluster 2** **i=2**	**Cluster 3** **i=3**
2. observation	0.0003	0.0004	0.9994
4. observation	0.9759	0.0089	0.0152
6. observation	0.8662	0.0513	0.0824
7. observation	0.9748	0.0094	0.0158
8. observation	0.0005	0.9982	0.0013
9. observation	0.0011	0.9962	0.0027
10. observation	0.0009	0.9969	0.0022

Table 4.5 Membership values of validation data

Membership Values of Validation Data			
Observations of dataset	**Cluster 1** **i=1**	**Cluster 2** **i=2**	**Cluster 3** **i=3**
1. observation	0.0007	0.0011	0.9982
3. observation	0.9791	0.0076	0.0132
5. observation	0.8614	0.0524	0.0861

Step 3: After the membership valuesare found out for training data, membership valuesand their transformation such as $\exp(u)$, $\exp(u)^2$, $1/\exp(u)$ and $\mathrm{u} * \log(1+u)$ are added to original data matrix for each cluster. These transformations are defined by user. For this numerical example only membership values are decided to be added as new variables. The new augmented matrix is shown in Table 4.6.

Table 4.6 Membership values and input variables of training data for cluster 1

	Membership degrees	Variable1	Variable2	Variable3
Observations	0.0003	14.30	55.00	12.43
	0.9759	9.56	7.90	51.20
	0.8662	11.00	29.90	50.44
	0.9748	8.77	7.80	51.87
	0.0005	23.80	86.50	45.87
	0.0011	26.23	89.00	44.90
	0.0009	24.76	85.40	43.12

Step 4: Regression coefficients are found out for each cluster by using the regression equation; $\beta_i = ({\Phi_i}^T \Phi_i)^{-1}({\Phi_i}^T Y_i)$. More information on least square estimation (LSE) could be found in Appendix 1. As it can be seen from the equation, when all algorithms of fuzzy functions are applied, there will be "*c*" number of column matrix that consists of regression coefficients. In other words until the number of cluster "*c*" is reached, the same procedures are repeated and regression coefficients are found out for all clusters. In Table 4.7 final data matrix X, which consists of original input variables, membership values and identity matrix and corresponding output matrix are shown for cluster 1. Until we reach cluster number 3, same procedures are repeated. Also final input data matrixes and output data matrix for cluster 2 and 3 are shown in Table 4. 9 and Table 4.11

Table 4.7 Final input (Φ_i) and output data matrix of the training algorithm for cluster 1 (for i=1)

	Identity matrix	Membership values	Original Data Matrix-Inputs (X)			Output Matrix (Y)
			Variable1	Variable2	Variable3	Outputs
Observations	1	0.0003	14.30	55.00	12.43	58.93
	1	0.9759	9.56	7.90	51.20	122.00
	1	0.8662	11.00	29.90	50.44	120.18
	1	0.9748	8.77	7.80	51.87	131.11
	1	0.0005	23.80	86.50	45.87	75.00
	1	0.0011	26.23	89.00	44.90	73.20
	1	0.0009	24.76	85.40	43.12	76.00

The obtained regression coefficients by applying the equation $\beta_i = (\Phi_i{}^T\Phi_i)^{-1}(\Phi_i{}^T Y_i)$, for cluster 1 is shown in Table 4.8.

Table 4.8 Obtained regression coefficients for cluster 1

Regression Coefficients for Cluster 1 (β_1)
65.7958
26.3913
-1.1911
-0.0151
0.8936

Table 4.9 Final input (Φ_i) and output data matrix of the training algorithm for cluster 2 (for i=2)

	Identit y matrix	**Membership values**	**Original Data Matrix-Inputs (X)**			**Output Matrix (Y)**
			Variable1	**Variable2**	**Variable 3**	**Outputs**
Observations	1	0.0004	14.30	55.00	12.43	58.93
	1	0.0089	9.56	7.90	51.20	122.00
	1	0.0513	11.00	29.90	50.44	120.18
	1	0.0094	8.77	7.80	51.87	131.11
	1	0.9982	23.80	86.50	45.87	75.00
	1	0.9962	26.23	89.00	44.90	73.20
	1	0.9969	24.76	85.40	43.12	76.00

The regression coefficients for cluster 2 are shown in Table 4.10.

Table 4.10 Obtained regression coefficients for cluster 2

Regression Coefficients for Cluster 2 (β_2)
67.0901
-11.9701
-1.3130
-0.1231
1.4122

Table 4.11 Final input (Φ_i) and output data matrix of the training algorithm for cluster 3 (for i=3)

	Identity matrix	**Membership values**	**Original Data Matrix-Inputs (X)**			**Output Matrix (Y)**
			Variable1	**Variable2**	**Variable3**	**Outputs**
Observations	1	0.9994	14.30	55.00	12.43	58.93
	1	0.0152	9.56	7.90	51.20	122.00
	1	0.0824	11.00	29.90	50.44	120.18
	1	0.0158	8.77	7.80	51.87	131.11
	1	0.0013	23.80	86.50	45.87	75.00
	1	0.0027	26.23	89.00	44.90	73.20
	1	0.0022	24.76	85.40	43.12	76.00

The regression coefficients for cluster 3 are shown in Table 4.12 and thus all computing process for the regression coefficients is completed. The obtained regression coefficients for all clusters are shown in Table 4.13.

Table 4.12 Obtained regression coefficients for cluster 3

Regression Coefficients for Cluster 3 (β_3)
104.2185
-15.9448
-2.567
-0.0711
0.9152

Table 4.13 Obtained regression coefficients for all clusters

Regression Coefficients Matrix for All Clusters ($\beta_1, \beta_2, \beta_3$)		
Cluster 1 **i=1**	**Cluster 2** **i=2**	**Cluster 3** **i=3**
65.7958	67.0901	104.2185
26.3913	-11.9701	-15.9448
-1.1911	-1.3130	-2.5670
-0.0151	-0.1231	-0.0711
0.8936	1.4122	0.9152

Step 5: After regression coefficients are found (with regression equation of LSE), the estimated output values of training data for each cluster are calculated. The equation (4.14) expresses the general regression form of a multi-input single output model. In equation (4.15) the long form of regression model is expressed. Executing the equation (4.15), there will be "k" number of predicted values for all clusters ("c") for the training data. In the equation "k" is the vector identifier, "i" is the cluster identifier. The open forms of regression equations are also shown below for all clusters.

$$Y = X\beta + \varepsilon \tag{4.14}$$

$$y_{k,i} = \Phi_{k,j} * \beta_{j,i} \qquad k = 1, \dots, n \ , \quad i = 1, \dots, c \quad j = 1, \dots (nv + nm + 1) \tag{4.15}$$

$$y_{1,1}{}^{trn} = \Phi_{1,1} * \beta_{1,1} + \Phi_{1,2} * \beta_{2,1} + \Phi_{1,3} * \beta_{3,1} + \Phi_{1,4} * \beta_{4,1} + \Phi_{1,5} * \beta_{5,1} \tag{4.16}$$

$$y_{2,1}{}^{trn} = \Phi_{2,1} * \beta_{1,1} + \Phi_{2,2} * \beta_{2,1} + \Phi_{2,3} * \beta_{3,1} + \Phi_{2,4} * \beta_{4,1} + \Phi_{2,5} * \beta_{5,1} \tag{4.17}$$

$$y_{3,1}{}^{trn} = \Phi_{3,1} * \beta_{1,1} + \Phi_{3,2} * \beta_{2,1} + \Phi_{3,3} * \beta_{3,1} + \Phi_{3,4} * \beta_{4,1} + \Phi_{3,5} * \beta_{5,1} \tag{4.18}$$

$$y_{4,1}{}^{trn} = \Phi_{4,1} * \beta_{1,1} + \Phi_{4,2} * \beta_{2,1} + \Phi_{4,3} * \beta_{3,1} + \Phi_{4,4} * \beta_{4,1} + \Phi_{4,5} * \beta_{5,1} \tag{4.19}$$

$$y_{5,1}{}^{trn} = \Phi_{5,1} * \beta_{1,1} + \Phi_{5,2} * \beta_{2,1} + \Phi_{5,3} * \beta_{3,1} + \Phi_{5,4} * \beta_{4,1} + \Phi_{5,5} * \beta_{5,1} \tag{4.20}$$

$$y_{6,1}{}^{trn} = \Phi_{6,1} * \beta_{1,1} + \Phi_{6,2} * \beta_{2,1} + \Phi_{6,3} * \beta_{3,1} + \Phi_{6,4} * \beta_{4,1} + \Phi_{6,5} * \beta_{5,1} \tag{4.21}$$

$$y_{7,1}{}^{trn} = \Phi_{7,1} * \beta_{1,1} + \Phi_{7,2} * \beta_{2,1} + \Phi_{7,3} * \beta_{3,1} + \Phi_{7,4} * \beta_{4,1} + \Phi_{7,5} * \beta_{5,1} \tag{4.22}$$

$$y_{1,2}{}^{trn} = \Phi_{1,1} * \beta_{1,2} + \Phi_{1,2} * \beta_{2,2} + \Phi_{1,3} * \beta_{3,2} + \Phi_{1,4} * \beta_{4,2} + \Phi_{1,5} * \beta_{5,2} \tag{4.23}$$

$$y_{2,2}{}^{trn} = \Phi_{2,1} * \beta_{1,2} + \Phi_{2,2} * \beta_{2,2} + \Phi_{2,3} * \beta_{3,2} + \Phi_{2,4} * \beta_{4,2} + \Phi_{2,5} * \beta_{5,2} \tag{4.24}$$

$$y_{3,2}{}^{trn} = \Phi_{3,1} * \beta_{1,2} + \Phi_{3,2} * \beta_{2,2} + \Phi_{3,3} * \beta_{3,2} + \Phi_{3,4} * \beta_{4,2} + \Phi_{3,5} * \beta_{5,2} \tag{4.25}$$

$$y_{4,2}{}^{trn} = \Phi_{4,1} * \beta_{1,2} + \Phi_{4,2} * \beta_{2,2} + \Phi_{4,3} * \beta_{3,2} + \Phi_{4,4} * \beta_{4,2} + \Phi_{4,5} * \beta_{5,2} \tag{4.26}$$

$$y_{5,2}{}^{trn} = \Phi_{5,1} * \beta_{1,2} + \Phi_{5,2} * \beta_{2,2} + \Phi_{5,3} * \beta_{3,2} + \Phi_{5,4} * \beta_{4,2} + \Phi_{5,5} * \beta_{5,2} \tag{4.27}$$

$$y_{6,2}{}^{trn} = \Phi_{6,1} * \beta_{1,2} + \Phi_{6,2} * \beta_{2,2} + \Phi_{6,3} * \beta_{3,2} + \Phi_{6,4} * \beta_{4,2} + \Phi_{6,5} * \beta_{5,2} \tag{4.28}$$

$$y_{7,2}{}^{trn} = \Phi_{7,1} * \beta_{1,2} + \Phi_{7,2} * \beta_{2,2} + \Phi_{7,3} * \beta_{3,2} + \Phi_{7,4} * \beta_{4,2} + \Phi_{7,5} * \beta_{5,2} \tag{4.29}$$

$$y_{1,3}{}^{trn} = \Phi_{1,1} * \beta_{1,3} + \Phi_{1,2} * \beta_{2,3} + \Phi_{1,3} * \beta_{3,3} + \Phi_{1,4} * \beta_{4,3} + \Phi_{1,5} * \beta_{5,3} \tag{4.30}$$

$$y_{2,3}{}^{trn} = \Phi_{2,1} * \beta_{1,3} + \Phi_{2,2} * \beta_{2,3} + \Phi_{2,3} * \beta_{3,3} + \Phi_{2,4} * \beta_{4,3} + \Phi_{2,5} * \beta_{5,3} \tag{4.31}$$

$$y_{3,3}{}^{trn} = \Phi_{3,1} * \beta_{1,3} + \Phi_{3,2} * \beta_{2,3} + \Phi_{3,3} * \beta_{3,3} + \Phi_{3,4} * \beta_{4,3} + \Phi_{3,5} * \beta_{5,3} \tag{4.32}$$

$$y_{4,3}{}^{trn} = \Phi_{4,1} * \beta_{1,3} + \Phi_{4,2} * \beta_{2,3} + \Phi_{4,3} * \beta_{3,3} + \Phi_{4,4} * \beta_{4,3} + \Phi_{4,5} * \beta_{5,3} \tag{4.33}$$

$$y_{5,3}{}^{trn} = \Phi_{5,1} * \beta_{1,3} + \Phi_{5,2} * \beta_{2,3} + \Phi_{5,3} * \beta_{3,3} + \Phi_{5,4} * \beta_{4,3} + \Phi_{5,5} * \beta_{5,3} \tag{4.34}$$

$$y_{6,3}{}^{trn} = \Phi_{6,1} * \beta_{1,3} + \Phi_{6,2} * \beta_{2,3} + \Phi_{6,3} * \beta_{3,3} + \Phi_{6,4} * \beta_{4,3} + \Phi_{6,5} * \beta_{5,3} \tag{4.35}$$

$$y_{7,3}{}^{trn} = \Phi_{7,1} * \beta_{1,3} + \Phi_{7,2} * \beta_{2,3} + \Phi_{7,3} * \beta_{3,3} + \Phi_{7,4} * \beta_{4,3} + \Phi_{7,5} * \beta_{5,3} \tag{4.36}$$

In order to predict the output values, data matrixes and coefficient matrixes for cluster 1, cluster 2 and cluster 3 are shown respectively in Table 4.14, 4.15 and 4.16.In order to facilitate to following up, all of the calculations are shown below one by one.

Table 4.14 Obtaining predicted output values of training data for cluster 1

Identity matrix	Membership degrees	Original data matrix-inputs (X)			Regression Coefficients Matrix for All Clusters		
		Variable 1	Variable 2	Variable 3	Cluster 1	Cluster 2	Cluster 3
1	0.0003	14.30	55.00	12.43	65.7958	67.0901	104.2185
1	0.9759	9.56	7.90	51.20	26.3913	-11.9701	-15.9448
1	0.8662	11.00	29.90	50.44	-1.1911	-1.3130	-2.5670
1	0.9748	8.77	7.80	51.87	-0.0151	-0.1231	-0.0711
1	0.0005	23.80	86.50	45.87	0.8936	1.4122	0.9152
1	0.0011	26.23	89.00	44.90			
1	0.0009	24.76	85.40	43.12			

$$\begin{aligned} y_{1,1}{}^{trn} &= 1 * 65.7958 + (0.0003) * (26.3913) + (14.30) * (-1.1911) + (55.00) * (-0.0151) \\ &\quad + (12.43) * (0.8936) = 59.0482 \end{aligned}$$

$$\begin{aligned} y_{2,1}{}^{trn} &= 1 * 65.7958 + (0.9759) * (26.3913) + (9.56) * (-1.1911) + (7.90) * (-0.0151) \\ &\quad + (51.20) * (0.8936) = 125.7977 \end{aligned}$$

$$\begin{aligned} y_{3,1}{}^{trn} &= 1 * 65.7958 + (0.8662) * (26.3913) + (11.00) * (-1.1911) + (29.90) * (-0.0151) \\ &\quad + (50.44) * (0.8936) = 120.1786 \end{aligned}$$

$$y_{4,1}{}^{trn} = 1 * 65.7958 + (0.9748) * (26.3913) + (8.77) * (-1.1911) + (7.80) * (-0.0151) + (51.87) * (0.8936) = 127.311786$$

$$y_{5,1}{}^{trn} = 1 * 65.7958 + (0.0005) * (26.3913) + (23.80) * (-1.1911) + (86.50) * (-0.0151) + (45.87) * (0.8936) = 77.1484$$

$$y_{6,1}{}^{trn} = 1 * 65.7958 + (0.0011) * (26.3913) + (26.23) * (-1.1911) + (89.00) * (-0.0151) + (44.90) * (0.8936) = 73.3648$$

$$y_{7,1}{}^{trn} = 1 * 65.7958 + (0.0009) * (26.3913) + (24.76) * (-1.1911) + (85.40) * (-0.0151) + (43.12) * (0.8936) = 73.5724$$

Table 4.15 Obtaining predicted output values of validation data for cluster 2

Identity matrix	**Membership degrees**	**Original data matrix-inputs (X)**			**Regression Coefficients Matrix For All Clusters**		
		Variable 1	**Variable 2**	**Variable 3**	**Cluster 1**	**Cluster 2**	**Cluster 3**
1	0.0004	14.30	55.00	12.43	65.7958	67.0901	104.2185
1	0.0089	9.56	7.90	51.20	26.3913	-11.9701	-15.9448
1	0.0513	11.00	29.90	50.44	-1.1911	-1.3130	-2.5670
1	0.0094	8.77	7.80	51.87	-0.0151	-0.1231	-0.0711
1	0.9982	23.80	86.50	45.87	0.8936	1.4122	0.9152
1	0.9962	26.23	89.00	44.90			
1	0.9969	24.76	85.40	43.12			

$$y_{1,2}{}^{trn} = 1 * 67.0901 + (0.0004) * (-11.9701) + (14.30) * (-1.3130) + (55.00) * (-0.1231) + (12.43) * (1.4122) = 59.0917$$

$$y_{2,2}{}^{trn} = 1 * 67.0901 + (0.0089) * (-11.9701) + (9.56) * (-1.3130) + (7.90) * (-0.1231) + (51.20) * (1.4122) = 125.7645$$

$$y_{3,2}{}^{trn} = 1 * 67.0901 + (0.0513) * (-11.9701) + (11.00) * (-1.3130) + (29.90) * (-0.1231) + (50.44) * (1.4122) = 119.5844$$

$$y_{4,2}{}^{trn} = 1 * 67.0901 + (0.0094) * (-11.9701) + (8.77) * (-1.3130) + (7.80) * (-0.1231) + (51.87) * (1.4122) = 127.7547$$

$$y_{5,2}{}^{trn} = 1 * 67.0901 + (0.9982) * (-11.9701) + (23.80) * (-1.3130) + (86.50) * (-0.1231) + (45.87) * (1.4122) = 78.0209$$

$$y_{6,2}{}^{trn} = 1 * 67.0901 + (0.9962) * (-11.9701) + (26.23) * (-1.3130) + (89.00) * (-0.1231) + (44.90) * (1.4122) = 73.1765$$

$$y_{7,2}{}^{trn} = 1 * 67.0901 + (09969) * (-11.9701) + (24.76) * (-1.3130) + (85.40) * (-0.1231) + (43.12) * (1.4122) = 73.0273$$

Table 4.16 Obtaining predicted output values of validation data for cluster 3

Identity matrix	Membership degrees	Original Data Matrix-Inputs (X)			Regression Coefficients Matrix for All Clusters		
		Variable 1	Variable 2	Variable 3	Cluster 1	Cluster 2	Cluster 3
1	0.9994	14.30	55.00	12.43	65.7958	67.0901	104.2185
1	0.0152	9.56	7.90	51.20	26.3913	-11.9701	-15.9448
1	0.0824	11.00	29.90	50.44	-1.1911	-1.3130	-2.5670
1	0.0158	8.77	7.80	51.87	-0.0151	-0.1231	-0.0711
1	0.0013	23.80	86.50	45.87	0.8936	1.4122	0.9152
1	0.0027	26.23	89.00	44.90			
1	0.0022	24.76	85.40	43.12			

$$y_{1,3}{}^{trn} = 1 * 104.2185 + (0.9994) * (-15.9448) + (14.30) * (-2.5670) + (55.00) * (-0.0711) + (12.43) * (0.9152) = 59.0418$$

$$y_{2,3}{}^{trn} = 1 * 104.2185 + (0.0152) * (-15.9448) + (9.56) * (-2.5670) + (7.90) * (-0.0711) + (51.20) * (0.9152) = 125.7344$$

$$y_{3,3}{}^{trn} = 1 * 104.2185 + (0.0824) * (-15.9448) + (11.00) * (-2.5670) + (29.90) * (-0.0711) + (50.44) * (0.9152) = 118.7064$$

$$y_{4,3}{}^{trn} = 1 * 104.2185 + (0.0158) * (-15.9448) + (8.77) * (-2.5670) + (7.80) * (-0.0711) + (51.87) * (0.9152) = 128.3727$$

$$y_{5,3}{}^{trn} = 1 * 104.2185 + (0.0013) * (-15.9448) + (23.80) * (-2.5670) + (86.50) * (-0.0711) + (45.87) * (0.9152) = 78.9363$$

$$y_{6,3}{}^{trn} = 1 * 104.2185 + (0.0027) * (-15.9448) + (26.23) * (-2.5670) + (89.00) * (-0.0711) + (44.90) * (0.9152) = 71.6103$$

$$y_{7,3}{}^{trn} = 1 * 104.2185 + (0.0022) * (-15.9448) + (24.76) * (-2.5670) + (85.40) * (-0.0711) + (43.12) * (0.9152) = 74.0182$$

The predicted output values for the training data for each cluster are shown in Table 4.17.

Table 4.17 Obtained predicted output values of training data for each cluster

Prediction Values The Observation of Training Data Set		
Cluster 1	**Cluster 2**	**Cluster 3**
59.0482	59.0917	59.0418
125.7977	125.7645	125.7344
120.1786	119.5844	118.7064
127.31	127.7547	128.3727
77.1484	78.0209	78.9363
73.3648	73.1765	71.6103
73.5724	73.0273	74.0182

As it can be seen from Table (4.17), we obtain a matrix that consists of “c” number of columns after regression equation is applied.

Step 6: In the final step, a single output is obtained for each observation by weighting the obtained output values with their corresponding membership values. In order to facilitate to follow up, the membership degree matrix is rewritten in the Table 4.18.

Table 4.18 Membership degrees of training data

Membership Degrees of Training Data			
Observations	**Cluster 1** **i=1**	**Cluster 2** **i=2**	**Cluster 3** **i=3**
2. observation	0.0003	0.0004	0.9994
4. observation	0.9759	0.0089	0.0152
6. observation	0.8662	0.0513	0.0824
7. observation	0.9748	0.0094	0.0158
8. observation	0.0005	0.9982	0.0013
9. observation	0.0011	0.9962	0.0027
10. observation	0.0009	0.9969	0.0022

Executing equation (4.37), membership matrix and predicted value matrix are multiplied and then divided into sum of membership values of k.th observation. In other words in this step, all these "c" number of predicted values are weighted with membership degrees in order to obtain a single predicted value for each observation.

$$\hat{Y}_k = \frac{\sum_i^c \mu_{ki} y_{ki}}{\sum_i^c \mu_{ki}} \tag{4.37}$$

For all observations the single final predicted output values are calculated as follows;

$$\hat{Y}_1^{trn} = \frac{(0.0003 * 59.0482 + 0.0004 * 59.0917 + 0.9994 * 59.0418)}{(0.0003 + 0.0004 + 0.9994)} = 59.0418$$

$$\hat{Y}_2^{trn} = \frac{(0.9759 * 125.7977 + 0.0089 * 125.7645 + 0.0152 * 125.7344)}{(0.9759 + 0.0089 + 0.0152)} = 125.7965$$

$$\hat{Y}_3^{trn} = \frac{(0.8662 * 120.1786 + 0.0513 * 119.5844 + 0.0824 * 118.7064)}{(0.8662 + 0.0513 + 0.0824)} = 120.0267$$

$$\hat{Y}_4^{trn} = \frac{(0.9748 * 127.31 + 0.0094 * 127.7547 + 0.0158 * 128.3727)}{(0.9748 + 0.0094 + 0.0158)} = 127.331$$

$$\hat{Y}_5^{trn} = \frac{(0.0005 * 127.31 + 0.9982 * 78.0209 + 0.0013 * 78.9363)}{(0.0005 + 0.9982 + 0.0013)} = 78.0216$$

$$\hat{Y}_6^{trn} = \frac{(0.0011 * 73.3648 + 0.9962 * 73.1765 + 0.0027 * 71.6103)}{(0.0011 + 0.9962 + 0.0027)} = 73.1725$$

$$\hat{Y}_7^{trn} = \frac{(0.0009 * 73.5724 + 0.9969 * 73.0273 + 0.0022 * 74.0182)}{(0.0009 + 0.9969 + 0.0022)} = 73.03$$

After weighting process is completed, the final predicted output values are obtained as shown in Table 4.19.

Table 4.19 Final single predicted values for training data

Predicted Values for Training Data
59.0418
125.7965
120.0267
127.3310
78.0216
73.1725
73.0300

R-square value which measure of how well future outcomes are likely to be predicted by the model is calculated at the final step for training data (Calculation of R-square value is explained in Appendix 2). R-square value for training data is found as 0.991.

Validation Data

In this section the same procedures are repeated for validation dataset. Based on the found out regression coefficients, output variables of the validation dataset are predicted and R-square value for the validation data is calculated.

Table 4.20 Randomly selected observations for validation data

	Variable1	Variable2	Variable3	Output
1. observation	15.00	56.00	10.33	58.77
3. observation	9.98	8.60	50.00	120.4
5. observation	10.12	30.10	49.80	123.5

Table 4.21 Membership degrees of validation dataset of artificial dataset

Membership Degrees of Validation Data			
Observations of dataset	Cluster 1 i=1	Cluster 2 i=2	Cluster 3 i=3
1. observation	0.0007	0.0011	0.9982
2. observation	0.9791	0.0076	0.0132
3. observation	0.8614	0.0524	0.0861

Table 4.22 Membership degrees and input variables of validation data for cluster 1

	Membership degrees	Variable1	Variable2	Variable3
Observations	0.0007	15.00	56.00	10.33
	0.9791	9.98	8.60	50.00
	0.8614	10.12	30.10	49.80

Table 4.23 Obtaining predicted output values of validation data for cluster 1

Identity matrix	Membership degrees	Original Data Matrix-Inputs (X)			Regression Coefficients Matrix for All Clusters		
		Variable 1	Variable 2	Variable 3	Cluster 1	Cluster 2	Cluster 3
1	0.0007	15.00	56.00	10.33	65.7958	67.0901	104.2185
1	0.9791	9.98	8.60	50.00	26.3913	-11.9701	-15.9448
1	0.8614	10.12	30.10	49.80	-1.1911	-1.3130	-2.5670
					-0.0151	-0.1231	-0.0711
					0.8936	1.4122	0.9152

$$y_{1,1}{}^{val} = 1 * 65.7958 + (0.0007) * (26.3913) + (15.00) * (-1.1911) + (56.00) * (-0.0151) + (10.33) * (0.8936) = 56.3353$$

$$y_{2,1}{}^{val} = 1 * 65.7958 + (0.9791) * (26.3913) + (9.98) * (-1.1911) + (8.60) * (-0.0151) + (50.00) * (0.8936) = 124.3011$$

$$y_{3,1}{}^{val} = 1 * 65.7958 + (0.8614) * (26.3913) + (10.12) * (-1.1911) + (30.10) * (-0.0151) + (49.80) * (0.8936) = 120.5256$$

Table 4.24 Membership degrees and input variables of validation data for cluster 2

	Membership degrees	Variable1	Variable2	Variable3
Observations	0.0011	15.00	56.00	10.33
	0.0076	9.98	8.60	50.00
	0.0524	10.12	30.10	49.80

Table 4.25 Obtaining predicted output values of validation data for cluster 2

Identity matrix	Membership degrees	Original Data Matrix-Inputs (X)			Regression Coefficients Matrix for All Clusters		
		Variable 1	Variable 2	Variable 3	Cluster 1	Cluster 2	Cluster 3
1	0.0011	15.00	56.00	10.33	65.7958	67.0901	104.2185
1	0.0076	9.98	8.60	50.00	26.3913	-11.9701	-15.9448
1	0.0524	10.12	30.10	49.80	-1.1911	-1.3130	-2.5670
					-0.0151	-0.1231	-0.0711
					0.8936	1.4122	0.9152

$$y_{1,2}{}^{val} = 1 * 67.0901 + (0.0011) * (-11.9701) + (15.00) * (-1.3130) + (56.00) * (-0.1231) + (10.33) * (1.4122) = 55.0751$$

$$y_{2,2}{}^{val} = 1 * 67.0901 + (0.0076) * (-11.9701) + (9.98) * (-1.3130) + (8.60) * (-0.1231) + (50.00) * (1.4122) = 123.448$$

$$y_{3,2}{}^{val} = 1 * 67.0901 + (0.0524) * (-11.9701) + (10.12) * (-1.3130) + (30.10) * (-0.1231) + (49.80) * (1.4122) = 119.7984$$

Table 4.26 Membership degrees and input variables of validation data for cluster 3

	Membership degrees	Variable1	Variable2	Variable3
Observations	0.9982	15.00	56.00	10.33
	0.0132	9.98	8.60	50.00
	0.0861	10.12	30.10	49.80

Table 4.27 Obtaining predicted output values of validation data for cluster 3

Identity matrix	Membership degrees	Original data matrix-inputs (X)			Regression Coefficients Matrix For All Clusters		
		Variable 1	Variable 2	Variable 3	Cluster 1	Cluster 2	Cluster 3
1	0.9982	15.00	56.00	10.33	65.7958	67.0901	104.2185
1	0.0132	9.98	8.60	50.00	26.3913	-11.9701	-15.9448
1	0.0861	10.12	30.10	49.80	-1.1911	-1.3130	-2.5670
					-0.0151	-0.1231	-0.0711
					0.8936	1.4122	0.9152

$$y_{1,3}{}^{val} = 1 * 104.2185 + (0.9982) * (-15.9448) + (15.00) * (-2.5670) + (56.00) * (-0.0711) + (10.33) * (0.9152) = 55.271$$

$$y_{2,3}{}^{val} = 1 * 104.2185 + (0.0132) * (-15.9448) + (9.98) * (-2.5670) + (8.60) * (-0.0711) + (50.00) * (0.9152) = 123.5394$$

$$y_{3,3}{}^{val} = 1 * 104.2185 + (0.0861) * (-15.9448) + (10.12) * (-2.5670) + (30.10) * (-0.0711) + (49.80) * (0.9152) = 120.3063$$

The final predicted values for each observation of validation data are obtained as shown in Table 4.28.

Table 4.28 Final single predicted values for validation data

Prediction values the observation of validation data set		
56.3353	55.0751	55.2710
124.3011	123.4480	123.5394
120.5256	119.7984	120.3063

Membership matrix and predicted value matrix are multiplied and divided into sum of membership values of $k.th$ observation of validation data by executing equation (4.37). In order to follow up easily, membership values of validation data are rewritten below.

Table 4.29 Membership degrees of validation data of artificial dataset

Membership Degrees of Validation Data			
Observations of dataset	Cluster 1 i=1	Cluster 2 i=2	Cluster 3 i=3
1. observation	0.0007	0.0011	0.9982
3. observation	0.9791	0.0076	0.0132
5. observation	0.8614	0.0524	0.0861

$$\hat{Y}_1^{val} = \frac{(0.0007 * 56.3353 + 0.0011 * 55.0751 + 0.9982 * 55.2710)}{(0.0007 + 0.0011 + 0.9982)} = 55.2716$$

$$\hat{Y}_2^{val} = \frac{(0.9791 * 124.3011 + 0.0076 * 123.4480 + 0.0132 * 123.5394)}{(0.9791 + 0.0076 + 0.0132)} = 124.2845$$

$$\hat{Y}_3^{val} = \frac{(0.8614 * 120.5256 + 0.0524 * 119.7984 + 0.0861 * 120.3063)}{(0.8614 + 0.0524 + 0.0861)} = 120.4686$$

Table 4.30 Final single predicted values for validation data

Predicted Values for Validation Data
55.2716
124.2845
120.4686

After final predicted values are calculated, R-square value is caluculated. R-square value for validation data is found as 0.9863. R-square value is also calculated for all-data which is found as 0.9896.

4.4Conclusions

In this chapter, Türkşen's fuzzy functions concept is introduced briefly. To sum up, fuzzy functions concept is recommended as an alternative to fuzzy rule bases in order to eliminate difficulties of it and enable to handle large and complex systems that fuzzy rule base system may remain incapable. The theory of fuzzy functions approach is based on membership values and regression functions and this constitutes the main difference of it. After the fundamental properties of fuzzy functions are introduced, the structure identification and reasoning mechanism of the fuzzy function approach for regression type models is explained. Finally a detailed computational example is provided in order to facilitate better understanding of the fuzzy functions approach.

In the next chapter, a new approach which makes use genetic programming in defining fuzzy functions instead of regressions equations is presented. It is aimed to investigate whether it is possible to further improve the performance of the fuzzy functions approach by integrating it with genetic programming approaches.

CHAPTER FIVE
A BRIEF OVERVIEW OF GENETIC PROGRAMMING

5.1 Introduction

Genetic algorithm (GA) which was proposed by Holland in the 1960s is a search and optimization technique and is based on the principles of natural selection. In GAs, each candidate solution is called an individual or a chromosome and aggregation of these chromosomes form the populations.

> The genetic algorithm (GA) transforms a population (set) of individual objects, each with an associated fitness value, into a new generation of the population using the principle of reproduction and survival of the fittest and analogs of naturally occurring genetic operations such as crossover and mutation (Koza, 1995, p. 589).

Palit and Popovic (2005) express the features of a typical GAs to be able to solve an optimization problem, as follows:

- Genetic representation of each possible solution,
- A population of encoded solutions,
- A evaluation function which evaluates the fittingness of each solution,
- Genetic operators that are used in order to form new populations,
- Control parameters such as population size and number of generations.

Broadly the three genetic operations which are selection, crossover and mutation constitute the concept of genetic algorithms. These operations are used in order to select the most proper offspring to be able to obtain succeeding generations. Firstly, from the current population the individuals are chosen and then mated in order to generate next generations. These operations are explained below briefly.

- **Selection:** Selection is the process where individuals are chosen in order to be processed. Selection process is based on the survival-of-the-fittest strategy which means that the individual compete with each other to be able to survive in the population. There are a number of selection methodologies and the most commonly known methods are fitness proportionate selection, greedy over-selection, and tournament selection.

- **Crossover:** Crossover operation is basically based on the swapping of genetic material between two parent strings. For crossover operations two individuals are needed and these individuals breed two different individuals for the new population. Crossover is a process of information exchange between two parent chromosomes and genetic materials that are coming from these two parent chromosomes are mixed to in order to generate an offspring.

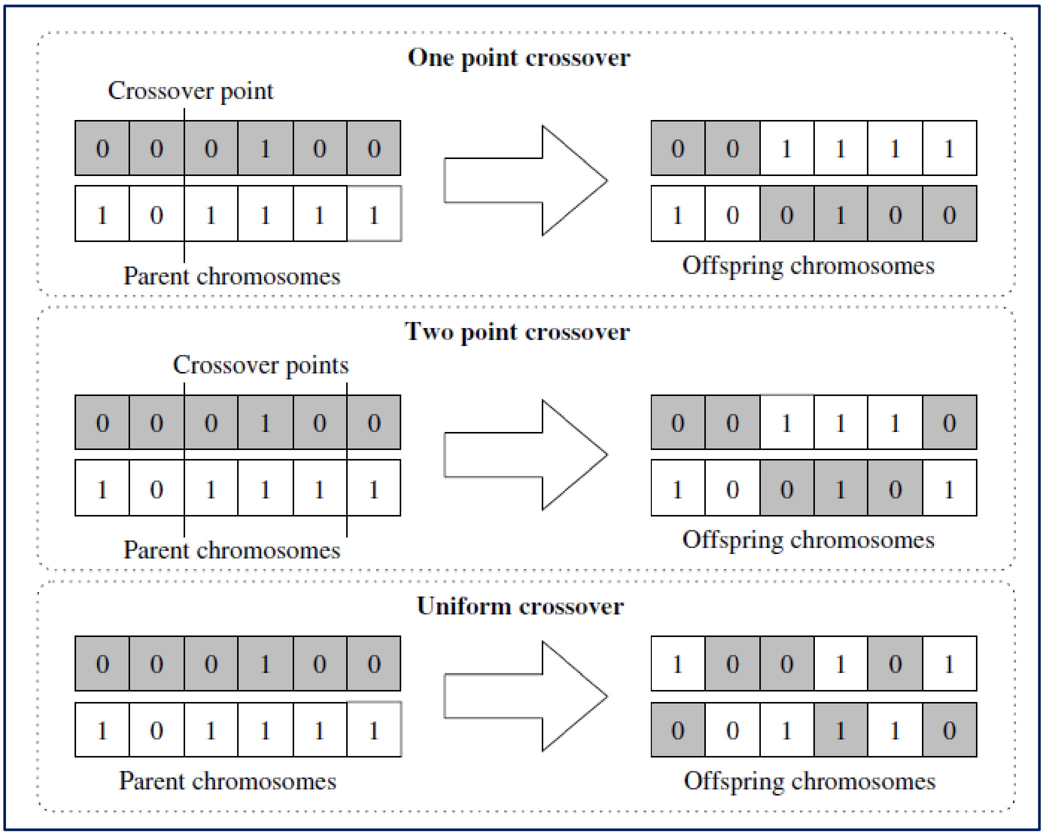


Figure 5.1 Sample representation of crossover operation (Sastry, Goldberg and Kandall, 2005)

- **Mutation:** Mutation operation operates on a single individual from the population and generates new genetic materials by which the diversity of the population is increased and the diversity of gene pool is maintained. By mutation operation one or more values are altered at randomly selected locations in randomly selected strings. Usually, mutation is applied after the crossover operation.

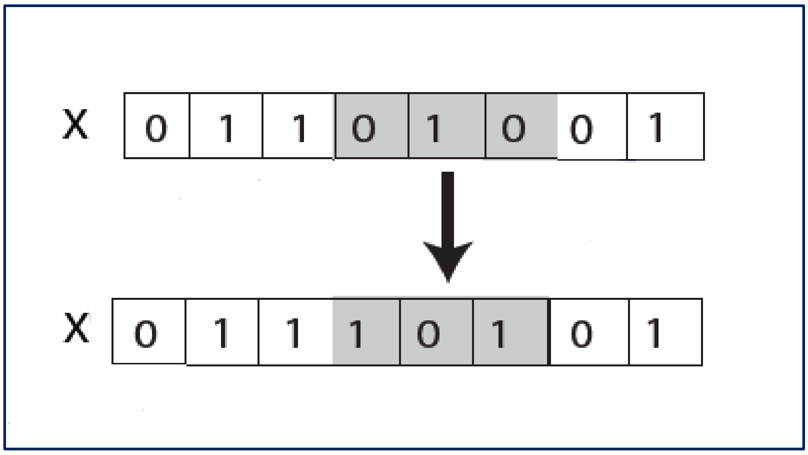


Figure 5.2 A sample representation of mutation of a chromosome X (Buttand Abhari, 2010)

Maulik and Bandyopadhyay (2000) described the application of GA, in their study as follows: Initially, a random population is created, which represents different points in the search space. An objective and fitness function is associated with each string that represents the degree of goodness of the string. Based on the principle of survival of the fittest, a few of the strings are selected and each is assigned a number of copies that go into the mating pool. Biologically inspired operators like crossover and mutation are applied on these strings to yield a new generation of strings. The process of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is satisfied (p. 1455).

Genetic algorithms provide a basis for many kinds of metaheuristicoptimization techniques with the combination of other modeling tools (Javadi, Farmani and Tan, 2005) and has been used in wide range of application areas for different kinds of problems such as data mining (Karthick, Saravanan and Vetrisalvan, 2012), clustering (Maulik and Bandyopadhyay, 2000) and business application (Grupe and Jooste, 2004) problems.

As Grupe and Jooste (2004) indicated in their study, when GAs are applied to the suitable problems they could be a very powerful techniques and capable of giving the closest solution to the optimum solutions. And it could be said that the underlying factor of the success of genetic algorithms is that genetic algorithms are able to consider many points simultaneously and provide nearly ideal solutions for many kinds of problems.

5.2 Genetic Programming

Genetic programming is a specialization of genetic algorithms and an evolutionary algorithm based machine learning technique in which each individual represented with a computer program and used in order to find out the best formula that represents the problem. By applying a number of processes that is consisting of reproduction, crossover and mutation operators, genetic programming generates the next population in which only the more successful genetic materials of individuals are existing.

As it was indicated before genetic programming is based on computer programs and the computer programs can give millions of solutions for a particular problem. Between these possible solutions, the best possible solution or solutions are chosen on the bases of some processes that are similar to principles of natural selection and evolution.

Koza (1995) explained the search space in genetic programming as the space of all possible computer programs which are composed of functions and terminals such as standard arithmetic operations, standard programming operations, standard mathematical functions, logical functions, or domain-specific functions.

In genetic programming each mathematical program is represented in a tree structure, where n trees form the population of size n. The crossover and mutation are applied on the population to obtain the new generation of computer programs. For each computer program, a fitness function is computed to scale its usefulness. In GP, usually one formula is obtained that can give the best answer (Hewai, 2012, p. 32).

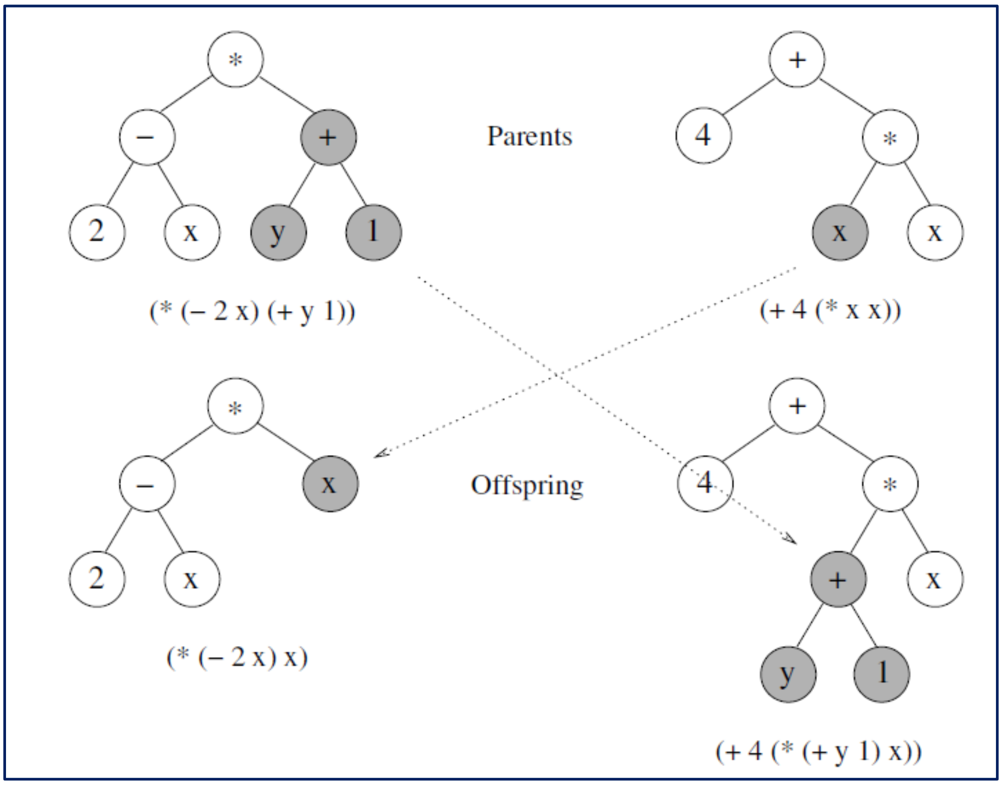


Figure 5.3A sample representation of a genetic programming tree (Brameier and Banzhaf,2007)

As Ponce-Cruz and Ramirez-Figueroa (2010) stated in their study, the basic difference between GA and GP is the evolution process while in GA strings of bits representing chromosomes are evolved, in genetic programming the whole structure of a computer program is evolved by the algorithm. And they indicated that thanks to this structure, genetic programming can handle the problems that are harder to manage by GAs.

As Cordon, Herrera, Hoffmann et al. (2001) indicated in their study, genetic programming has a wide range of application area and combining with different techniques, genetic programming has been applied to a variety of problems successfully by researchers (Al-Rahamneh, Reyalat, Sheta, Bani Ahmad and Al-Oqeili, 2011; Baykasoğlu, Gökçen and Özbakır, 2010; Çunkaş and Taşkıran, 2011; Fyfe, Marney and Tarbert, 1999; Chan, Kwong and Wong, 2011; Moreno-Torres, Llorà, Goldberg and Bhargava, 2013; Song and Zhang, 2012; Zhou et al., 2008).

Baykasoğlu et al. (2010) used genetic programming in data mining approaches in order to select dispatching rules according to subjected shop parameters. Chan et al. (2011) used also genetic programming for product development through modeling customer satisfaction, Zhou et al. (2008) used genetic programming in their study in order to propose a controller adaptive to traffic flows fordouble-deck elevator system. Al-Rahamneh et al. (2011) used genetic programming for the software reliability problems and built a software reliability growth model.

5.3 Fuzz Functions with Genetic Programming (GP)

In the present study, as a new contribution to existing studies on fuzzy functions it is proposed to use genetic programming in generating fuzzy functions as an alternative to using LSE or SVM with the intention of searching whether the performance of fuzzy functions approach could be improved by combining it with genetic programming or not. In this part of the study, this new approach is going to be introduced and also going to be supported with a numerical example in order to provide a better understanding.

Similar to fuzzy functions with LSE, the membership values and their transformations are used as new variables in fuzzy functions with GP. In order to find out the membership values FCM clustering algorithm is also used for the proposed model. Thereinafter the algorithm of the proposed model is introduced step by step and with an example all of these steps are explained numerically.

As it could be seen below,the steps in the algorithm of fuzzy functions with GP arequite similar to the steps in the algorithm of fuzzy functions with LSE. First of all, the parameters are decided to be able to execute FCM clustering algorithm and thenby executing FCM clustering algorithm, membership values are found out for each observation. In the following step, by adding these found out membership values and their transformations, the new data matrix is generated and the genetic programming is run in order to obtain the best formula for each cluster. Afterwards applying the found out formula,predicted values are obtained for each cluster. Finally same as in the algorithm of fuzzy functions with LSE, by weighting the obtained values with their corresponding membership values, a single predicted output value is obtained for all observations.

The algorithm of fuzzy function with GP is described below step by step;

Step 1: Firstly the parameters of the FCM clustering algorithm are decided;

- m≥1.1 (degree of fuzziness),
- c>1 (the number of clusters),
- ε (a termination threshold).

Step 2: Execute FCM clustering algorithm to find out cluster centers $v_i(xy)$of the dataset$Z(x, y)$.

$$\underset{\substack{1\leq i\leq c\\ 1\leq k\leq n}}{\forall} \mu_{ki}(xy) = \left(\sum_{j=1}^{c} \left(\frac{d_{ki}(xy)}{d_{kj}(xy)} \right)^{\frac{2}{m-1}} \right)^{-1} \quad d_{ki}^{xy} = \|(x_k, y_k) - v_i(x, y)\| \quad (5.1)$$

Step 3: Membership values are found out according to equation in (5.2);

$$\underset{\substack{1\leq i\leq c\\ 1\leq k\leq n}}{\forall} \mu_{ki}(x) = \left(\sum_{j=1}^{c} \left(\frac{d_{ki}(x)}{d_{kj}(x)} \right)^{2/(m-1)} \right)^{-1}, \ where d_{ki}(x) = \|x_k - v_i(x)\| \quad (5.2)$$

Step 4: Membership values of each input data sample, μ_{ki}, and their transformations are augmented to the original input matrix as shown in equation (5.3) for each cluster "i".

$$\Phi_i = \begin{bmatrix} \mu_{1,i}\, exp(\mu_{1,i})\,(\mu_{1,i})^p\, x_{1\times 1} & \cdots & x_{1\times nv} \\ \vdots \quad \vdots \quad \vdots \quad \vdots & \ddots & \vdots \\ \mu_{n,i} exp(\mu_{n,i})\,(\mu_{n,i})^p\, x_{n\times 1} & \cdots & x_{n\times nv} \end{bmatrix} \quad (5.3)$$

Step 5: After membership values are found out according to FCM clustering algorithm, Eureqa Formulize genetic programming software is run for all clusters individually and the equations that describe the data most appropriately is obtained for all new data matrixes that are generated by the addition of membership values. After the most appropriate equations are obtained, prediction process is carried out and predicted values are found out by applying the equation in (5.4). "α" represents the most appropriate equation for each cluster "i".

$$\hat{y}_{k,i} = \Phi_{k,i}\alpha_i \qquad (5.4)$$

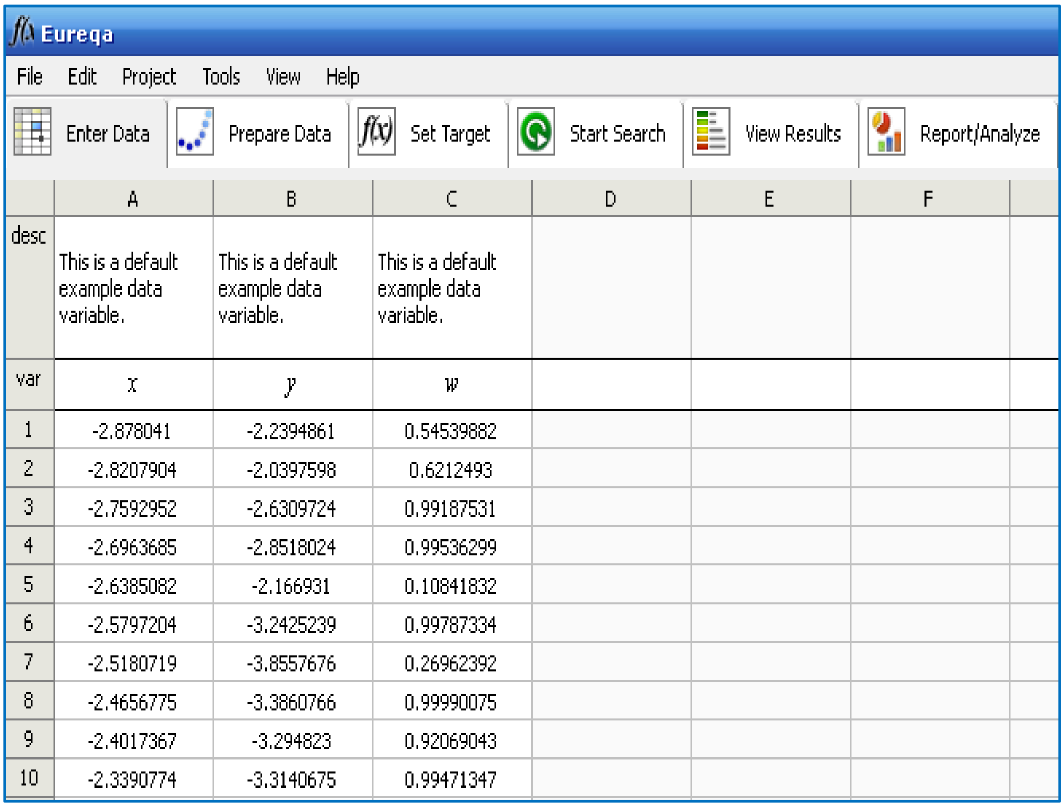
Step 6: Finally similar to fuzzy functions with LSE, single output values are calculated for each data vector by weighting predicted output values from each cluster with their corresponding membership values.

$$\hat{y}_k = \frac{\sum_i^c \hat{y}_{ki}\mu_{ki}}{\sum_i^c \mu_{ki}} \quad i = 1, \dots, c, \qquad k = 1, \dots, n \qquad (5.5)$$

In the following section Eureqa Formulize genetic programming software, which is used in the present thesis, is explained in order to provide a brief introduction.

5.3.1 The Introduction of the Eureqa Formulize Genetic Software Program

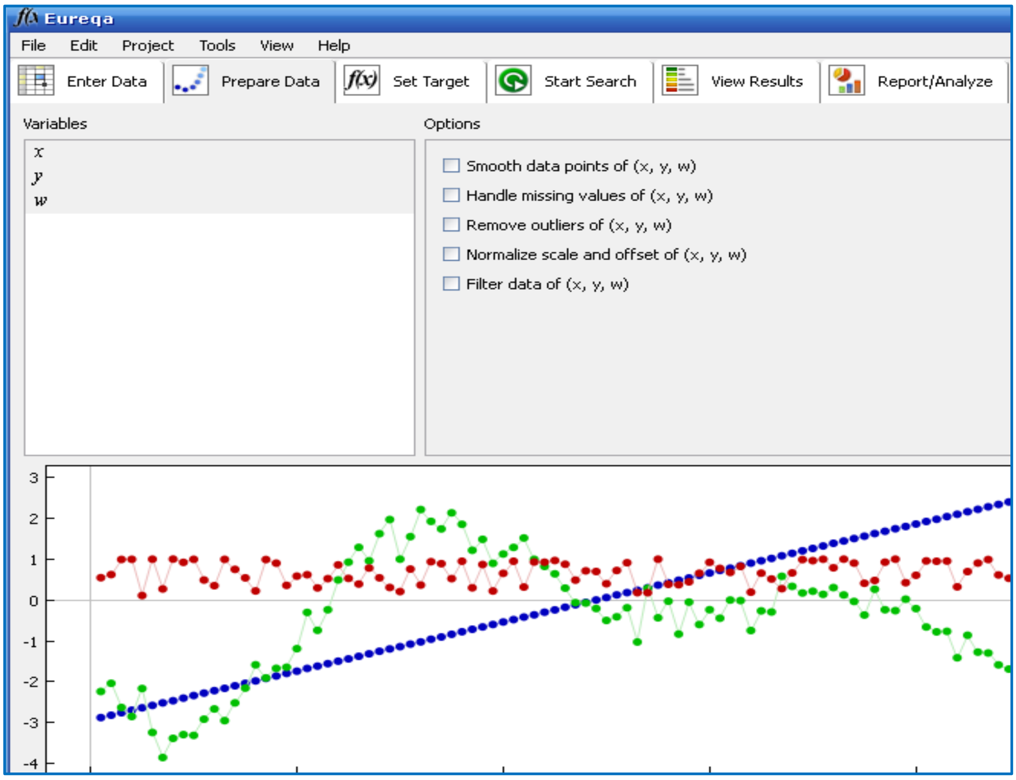
To give some brief information on "Eureqa Formulize" software program, firstly the dataset that is going to be searched is entered into the Eureqa Formulize program from the "Enter Data" tab as it is shown in the Figure 5.4.



	A	B	C	D	E	F
desc	This is a default example data variable.	This is a default example data variable.	This is a default example data variable.			
var	x	y	w			
1	-2.878041	-2.2394861	0.54539882			
2	-2.8207904	-2.0397598	0.6212493			
3	-2.7592952	-2.6309724	0.99187531			
4	-2.6963685	-2.8518024	0.99536299			
5	-2.6385082	-2.166931	0.10841832			
6	-2.5797204	-3.2425239	0.99787334			
7	-2.5180719	-3.8557676	0.26962392			
8	-2.4656775	-3.3860766	0.99990075			
9	-2.4017367	-3.294823	0.92069043			
10	-2.3390774	-3.3140675	0.99471347			

Figure 5.4 The screenshot of the “Enter Data” tab of Eureqa Formulize software program

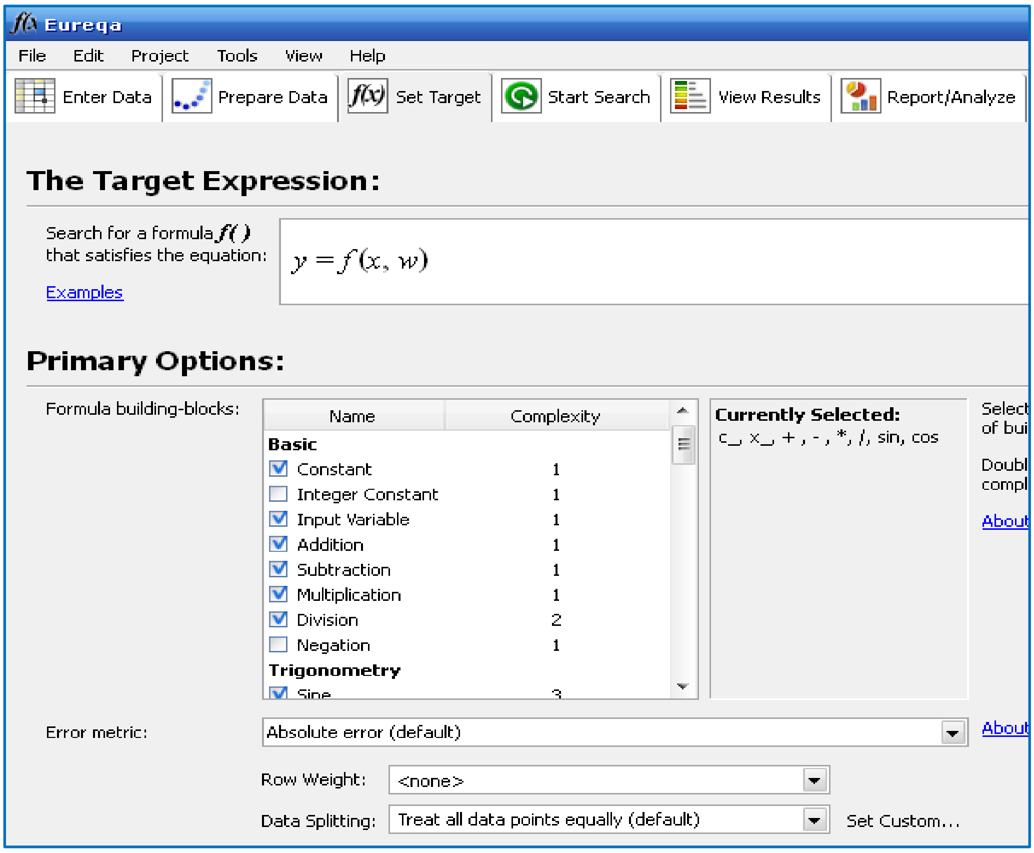
After the data is entered, with the “Prepare Data” tab the data can be prepared by smoothing the data, handling missing values, removing outliers, normalizing scale and offset or applying a filter (nutonian.com).The view of the “Prepare Data” tab window is shown in Figure 5.5.



□ Smooth data points of (x, y, w)
□ Handle missing values of (x, y, w)
□ Remove outliers of (x, y, w)
□ Normalize scale and offset of (x, y, w)
□ Filter data of (x, y, w)

Figure 5.5 The screenshot of the “Prepare Data” tab of Eureqa Formulize software program

As a next step, in the “Set Target” tabthe type of the formula that satisfies the equation is decided by choosing the operations (such as addition, subtraction, division, or sine) that we want to be in the equation.The view of the “Set Target” tab window is shown in Figure 5.6.



The Target Expression:

Search for a formula *f*( ) that satisfies the equation: $y = f(x, w)$

Examples

Primary Options:

Formula building-blocks:

Name	Complexity
Basic	
☑ Constant	1
□ Integer Constant	1
☑ Input Variable	1
☑ Addition	1
☑ Subtraction	1
☑ Multiplication	1
☑ Division	2
□ Negation	1
Trigonometry	
☑ Sine	3

Currently Selected: c_, x_, + , - , *, /, sin, cos

Select
of bui

Doubl
compl

About

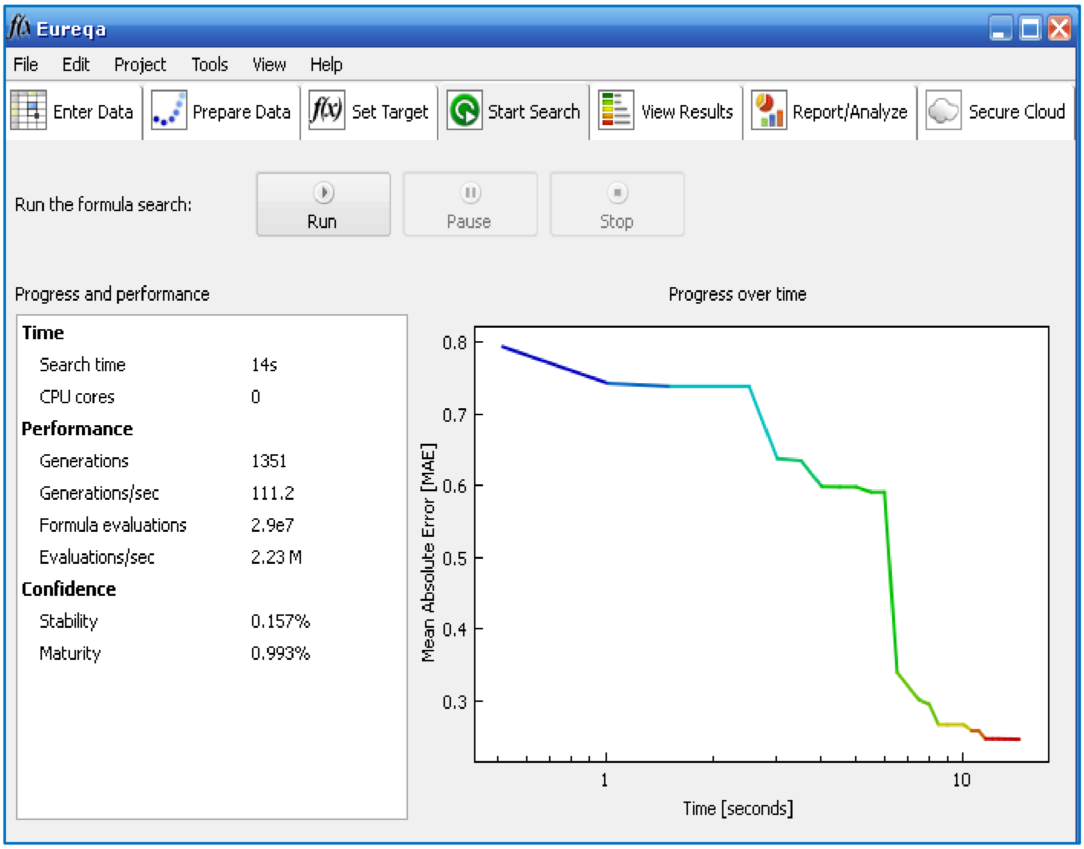
Error metric: Absolute error (default) About

Row Weight: <none>

Data Splitting: Treat all data points equally (default) Set Custom...

Figure 5.6 The screenshot of the “Set Target” window of Eureqa Formulize software program

After the parameters are determined to be in the formula, then with “Start Search” tab the search is started. The buttons in the “Start Search” tab provide to control the formula search. After stopping a search, clicking "Run" will give two options: continue the search from where it left off, or start fresh(nutonian.com). The screenshot of “Start Search” tab is presented in Figure 5.7.



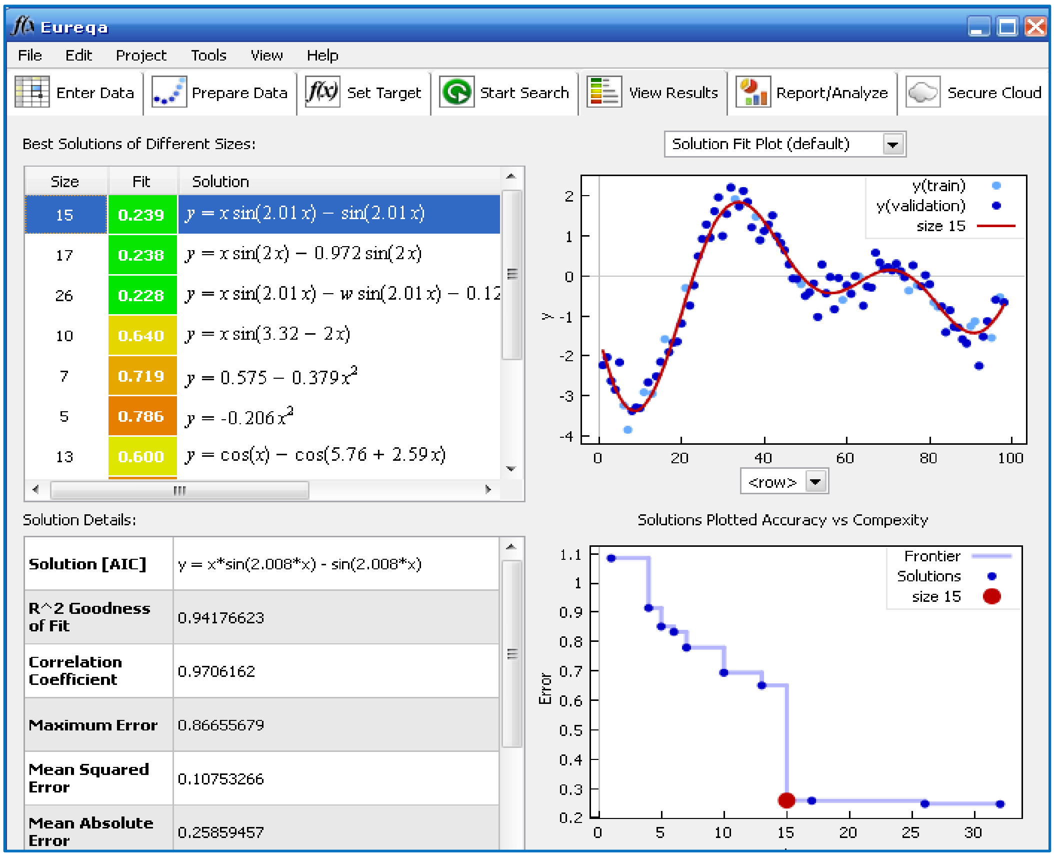
Run the formula search:

Progress and performance

Time	
Search time	14s
CPU cores	0
Performance	
Generations	1351
Generations/sec	111.2
Formula evaluations	2.9e7
Evaluations/sec	2.23 M
Confidence	
Stability	0.157%
Maturity	0.993%

Figure 5.7 The screenshot of the “Start Search” window of Eureqa Formulize software program

After the program is run for a period of time, with stop button, the search is ended and in “View Results” tab, the solutions that the program has found are shown. Between these solutions, the most appropriate equationsare chosen. To be an example the screenshot of the tab is depicted in Figure 5.8.



Best Solutions of Different Sizes:

Size	Fit	Solution
15	0.239	$y = x\sin(2.01\,x) - \sin(2.01\,x)$
17	0.238	$y = x\sin(2\,x) - 0.972\sin(2\,x)$
26	0.228	$y = x\sin(2.01\,x) - w\sin(2.01\,x) - 0.12$
10	0.640	$y = x\sin(3.32 - 2\,x)$
7	0.719	$y = 0.575 - 0.379x^2$
5	0.786	$y = -0.206\,x^2$
13	0.600	$y = \cos(x) - \cos(5.76 + 2.59\,x)$

Solution Details:

Solution [AIC]	y = x*sin(2.008*x) - sin(2.008*x)
R^2 Goodness of Fit	0.94176623
Correlation Coefficient	0.9706162
Maximum Error	0.86655679
Mean Squared Error	0.10753266
Mean Absolute Error	0.25859457

Figure 5.8 The screenshot of the “View Results” window of Eureqa Formulize software program

With the “Report/Analyze” tab as it is shown in Figure 5.9,some basic reports are provided. Selecting the desired report or tool from the "Select task" drop-down menu, and the necessary controls will appear (nutonian.com).

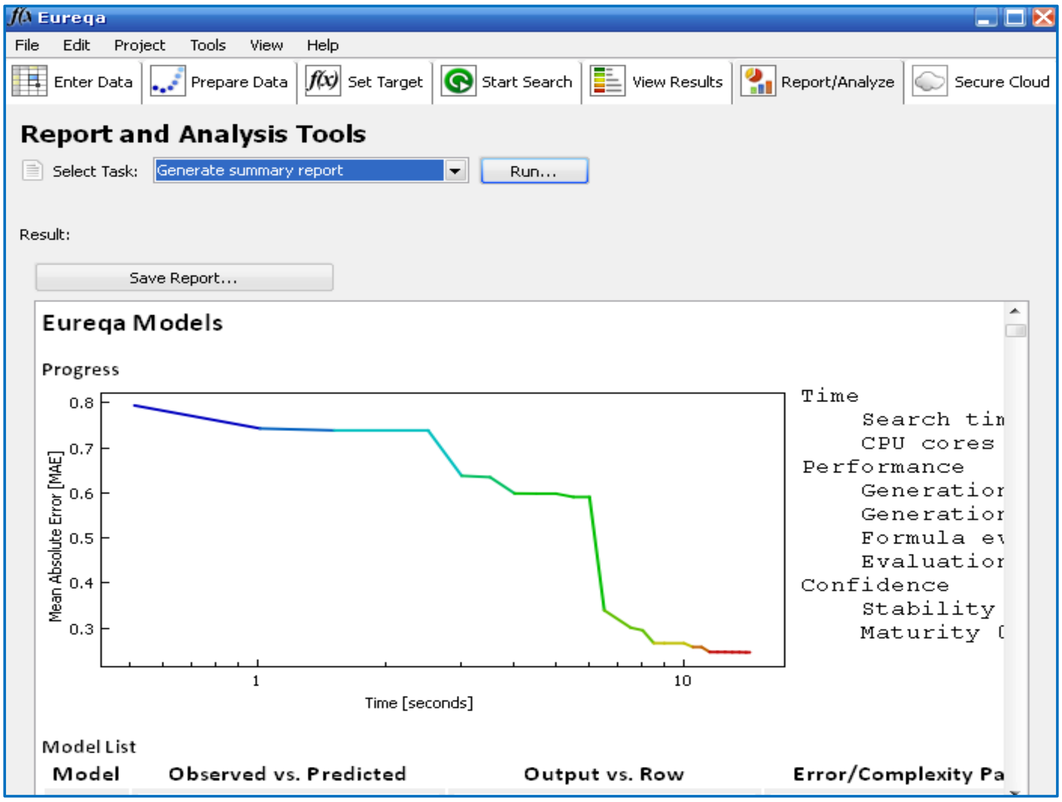
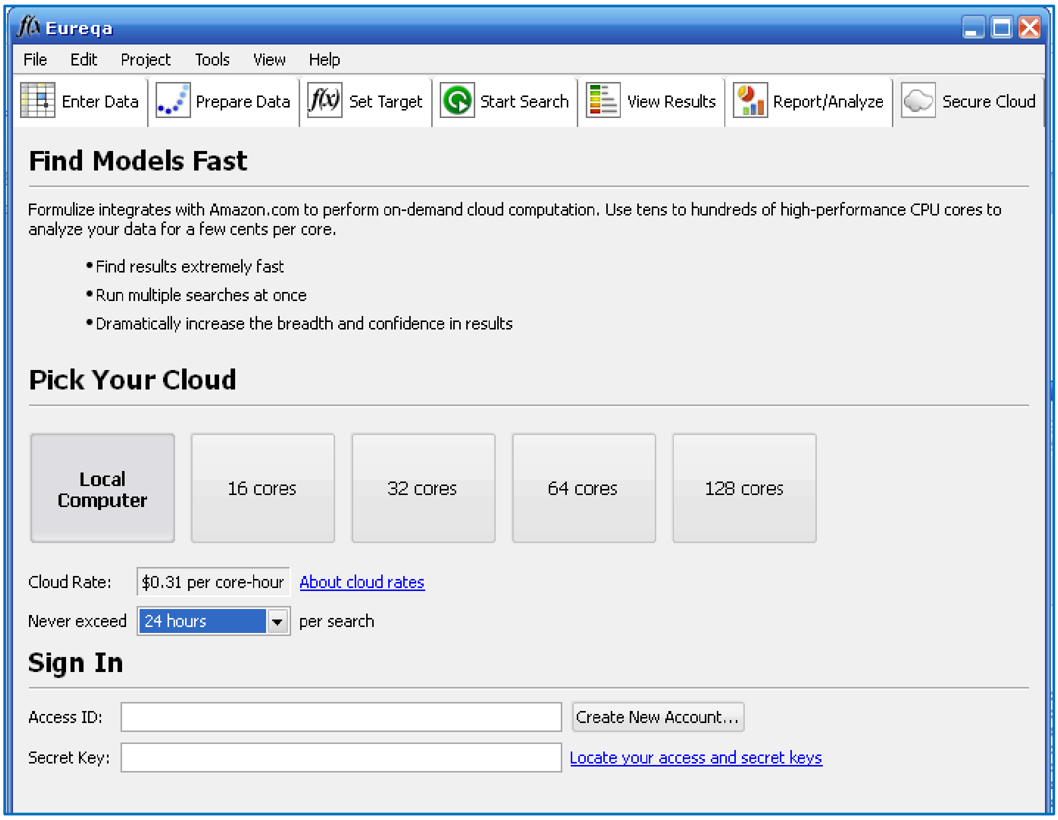


Figure 5.9 The screenshot of the “Report/Analyze” window of Eureqa Formulize software program

In “Secure Cloud” tab, the searches could be accelerated by enabling Formulize to use the Amazon Elastic Compute Cloud(Amazon EC2). A local computer typically has only four cores, which limits its search processing speed. By temporarily using additional cores, the search could be faster, deeper, and more confidence (nutonian.com). The screenshot of the tab is depicted in Figure 5.10.



Find Models Fast

Formulize integrates with Amazon.com to perform on-demand cloud computation. Use tens to hundreds of high-performance CPU cores to analyze your data for a few cents per core.

- Find results extremely fast
- Run multiple searches at once
- Dramatically increase the breadth and confidence in results

Pick Your Cloud

Cloud Rate: $0.31 per core-hour About cloud rates

Never exceed 24 hours per search

Sign In

Access ID:

Secret Key:

Figure 5.10 The screenshot of the “Secure cloud” window of Eureqa Formulize software program

5.3.2 Implementation of Fuzzy Functions with Genetic Programming

In this part, how the fuzzy functions are going to be implemented with genetic programming is going to be explained with the artificial dataset which is used for fuzzy functions with LSE in previous chapter. The artificial dataset is represented in Table 5.1 in order to following up easily.

Table 5.1 Input and output variables of generated artificial dataset

Observations	Variable1	Variable2	Variable3	Outputs
1. observation	15.00	56.00	10.33	58.77
2. observation	14.30	55.00	12.43	58.93
3. observation	9.98	8.60	50.00	120.40
4. observation	9.56	7.90	51.20	122.00
5. observation	10.12	30.10	49.80	123.50
6. observation	11.00	29.90	50.44	120.18
7. observation	8.77	7.80	51.87	131.11
8. observation	23.80	86.50	45.87	75.00
9. observation	26.23	89.00	44.90	73.20
10. observation	24.76	85.40	43.12	76.00

Step 1: Firstly "*c*" the optimum number of cluster should be found out and degree of fuzziness should be decided. As it can be remembered from the previous chapter the best partition was found as 3 for the artificial dataset.

Step 2: According to the optimum number of clusters, the membership values are found out with FCM algorithm. In Table 5.2 the obtained membership values of the data for all observations are shown.

Table 5.2 Membership values of the artificial data

Membership Values of the Data			
Observations of dataset	**Cluster 1 i=1**	**Cluster 2 i=2**	**Cluster 3 i=3**
1. observation	0.0007	0.0011	0.9982
2. observation	0.0003	0.0004	0.9994
3. observation	0.9791	0.0076	0.0132
4. observation	0.9759	0.0089	0.0152
5. observation	0.8614	0.0524	0.0861
6. observation	0.8662	0.0513	0.0824
7. observation	0.9748	0.0094	0.0158
8. observation	0.0005	0.9982	0.0013
9. observation	0.0011	0.9962	0.0027
10. observation	0.0009	0.9969	0.0022

Step 3: After the membership degrees are found out, membership degrees and their transformation such as $\exp(u)$, $\exp(u)^2$, $1/\exp(u)$ and $\mathrm{u} * \log(1+u)$ are added to original data matrix for each cluster. For this numerical example only membership values are decided to be added as new variables. The new augmented matrixes are respectively shown in Table 5.3, 5.4 and 5.5 for each cluster.

Table 5.3 Membership values and original input variables for cluster 1

	Membership degrees	**Variable1**	**Variable2**	**Variable3**
Observations	0.0007	15.00	56.00	10.33
	0.0003	14.30	55.00	12.43
	0.9791	9.98	8.60	50.00
	0.9759	9.56	7.90	51.20
	0.8614	10.12	30.10	49.80
	0.8662	11.00	29.90	50.44
	0.9748	8.77	7.80	51.87
	0.0005	23.80	86.50	45.87
	0.0011	26.23	89.00	44.90
	0.0009	24.76	85.40	43.12

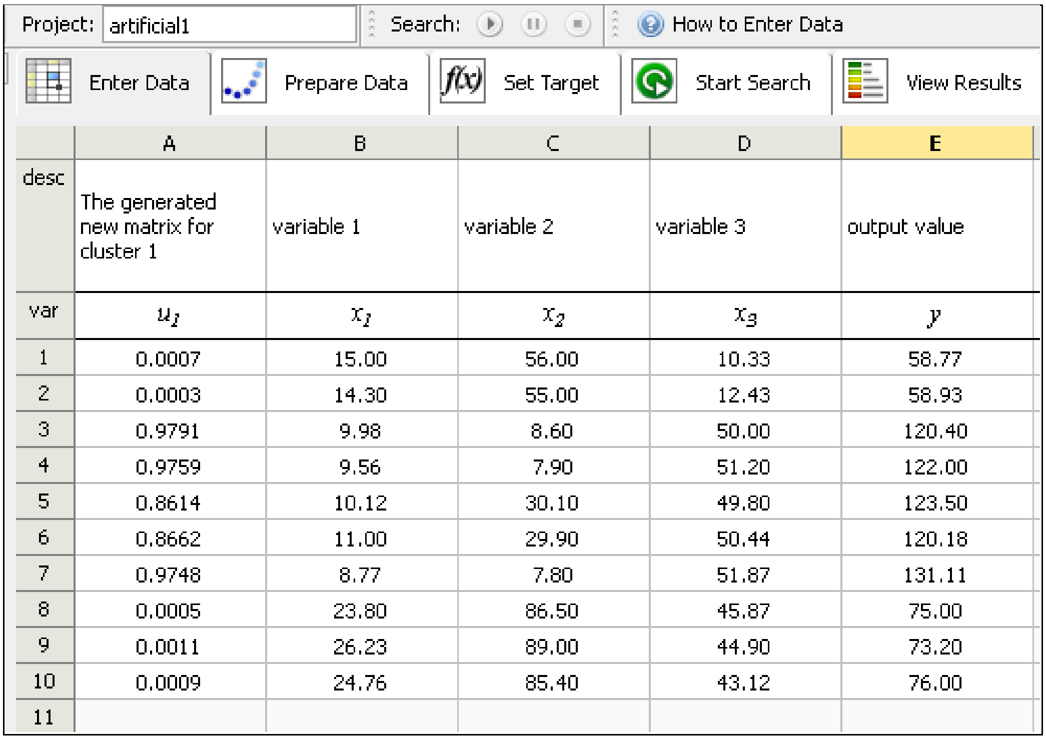
Table 5.4 Membership values and input variables for cluster 2

	Membership degrees	**Variable1**	**Variable2**	**Variable3**
Observations	0.0011	15.00	56.00	10.33
	0.0004	14.30	55.00	12.43
	0.0076	9.98	8.60	50.00
	0.0089	9.56	7.90	51.20
	0.0524	10.12	30.10	49.80
	0.0513	11.00	29.90	50.44
	0.0094	8.77	7.80	51.87
	0.9982	23.80	86.50	45.87
	0.9962	26.23	89.00	44.90
	0.9969	24.76	85.40	43.12

Table 5.5 Membership values and input variables for cluster 3

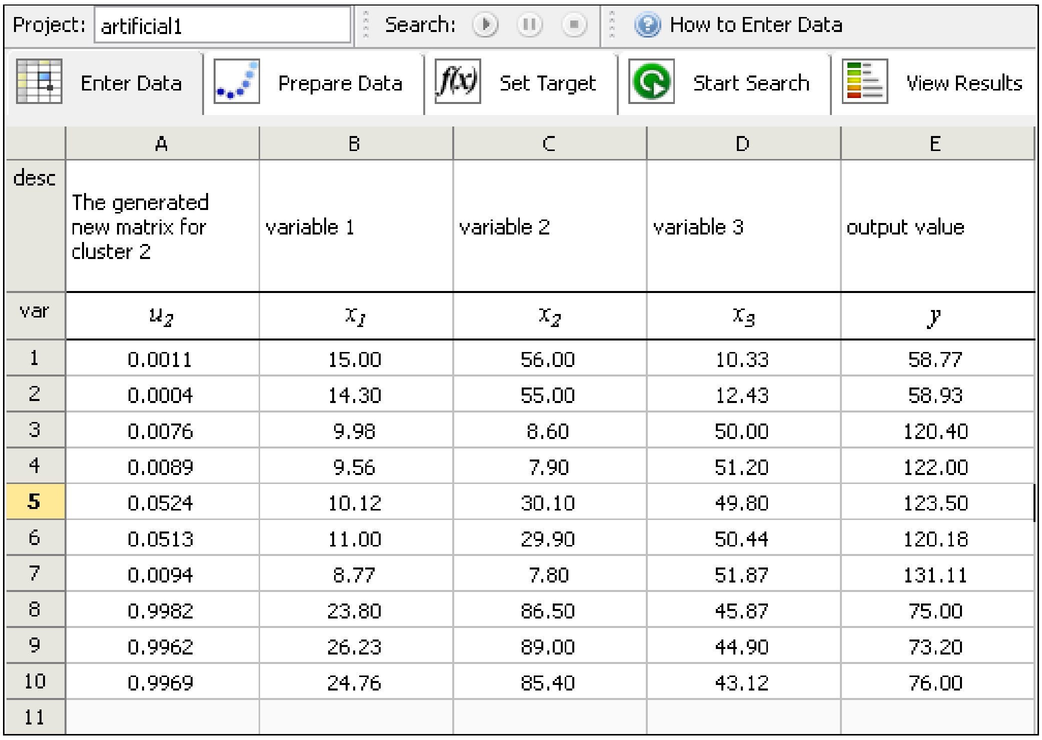
	Membership degrees	**Variable1**	**Variable2**	**Variable3**
Observations	0.9982	15.00	56.00	10.33
	0.9994	14.30	55.00	12.43
	0.0132	9.98	8.60	50.00
	0.0152	9.56	7.90	51.20
	0.0861	10.12	30.10	49.80
	0.0824	11.00	29.90	50.44
	0.0158	8.77	7.80	51.87
	0.0013	23.80	86.50	45.87
	0.0027	26.23	89.00	44.90
	0.0022	24.76	85.40	43.12

The views of new augmented matrixes in genetic programming software are also shown respectively in Figure 5.11, Figure 5.12 and Figure 5.13 for all clusters.



	A	B	C	D	E
desc	The generated new matrix for cluster 1	variable 1	variable 2	variable 3	output value
var	u_1	x_1	x_2	x_3	y
1	0.0007	15.00	56.00	10.33	58.77
2	0.0003	14.30	55.00	12.43	58.93
3	0.9791	9.98	8.60	50.00	120.40
4	0.9759	9.56	7.90	51.20	122.00
5	0.8614	10.12	30.10	49.80	123.50
6	0.8662	11.00	29.90	50.44	120.18
7	0.9748	8.77	7.80	51.87	131.11
8	0.0005	23.80	86.50	45.87	75.00
9	0.0011	26.23	89.00	44.90	73.20
10	0.0009	24.76	85.40	43.12	76.00
11					

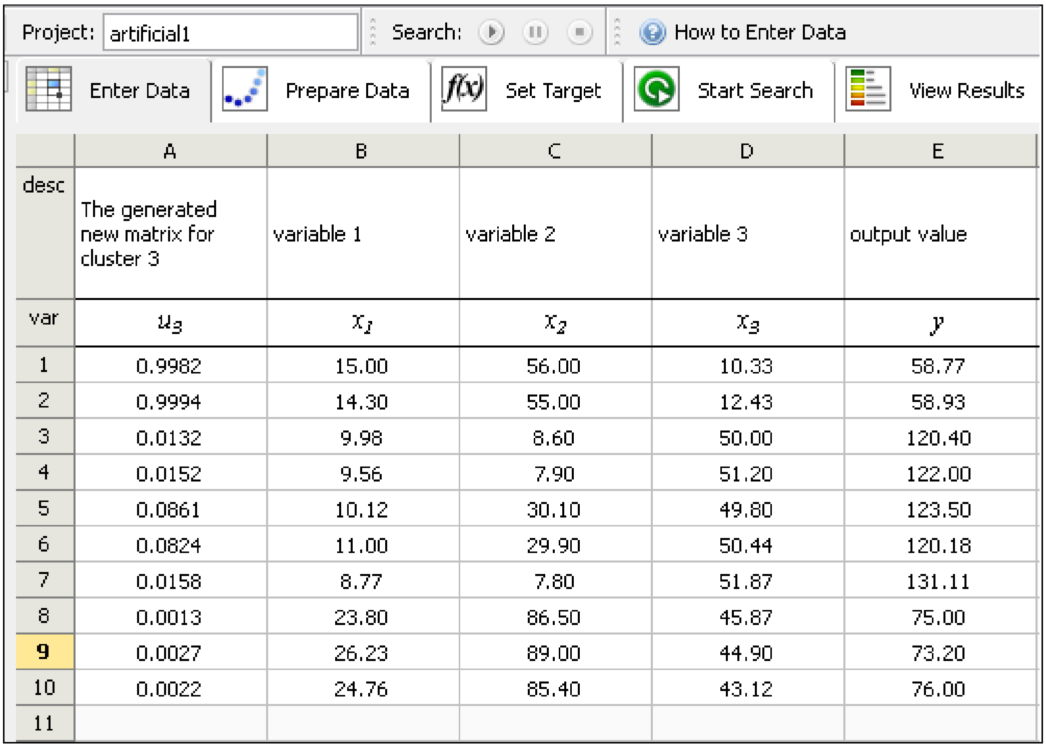
Figure 5.11 Eureqa-formulize screenshot of the artificial dataset for cluster 1



Project: artificial1

	A	B	C	D	E
desc	The generated new matrix for cluster 2	variable 1	variable 2	variable 3	output value
var	u_2	x_1	x_2	x_3	y
1	0.0011	15.00	56.00	10.33	58.77
2	0.0004	14.30	55.00	12.43	58.93
3	0.0076	9.98	8.60	50.00	120.40
4	0.0089	9.56	7.90	51.20	122.00
5	0.0524	10.12	30.10	49.80	123.50
6	0.0513	11.00	29.90	50.44	120.18
7	0.0094	8.77	7.80	51.87	131.11
8	0.9982	23.80	86.50	45.87	75.00
9	0.9962	26.23	89.00	44.90	73.20
10	0.9969	24.76	85.40	43.12	76.00
11					

Figure 5.12 Eureqa-formulize screenshot of the artificial dataset for cluster 2

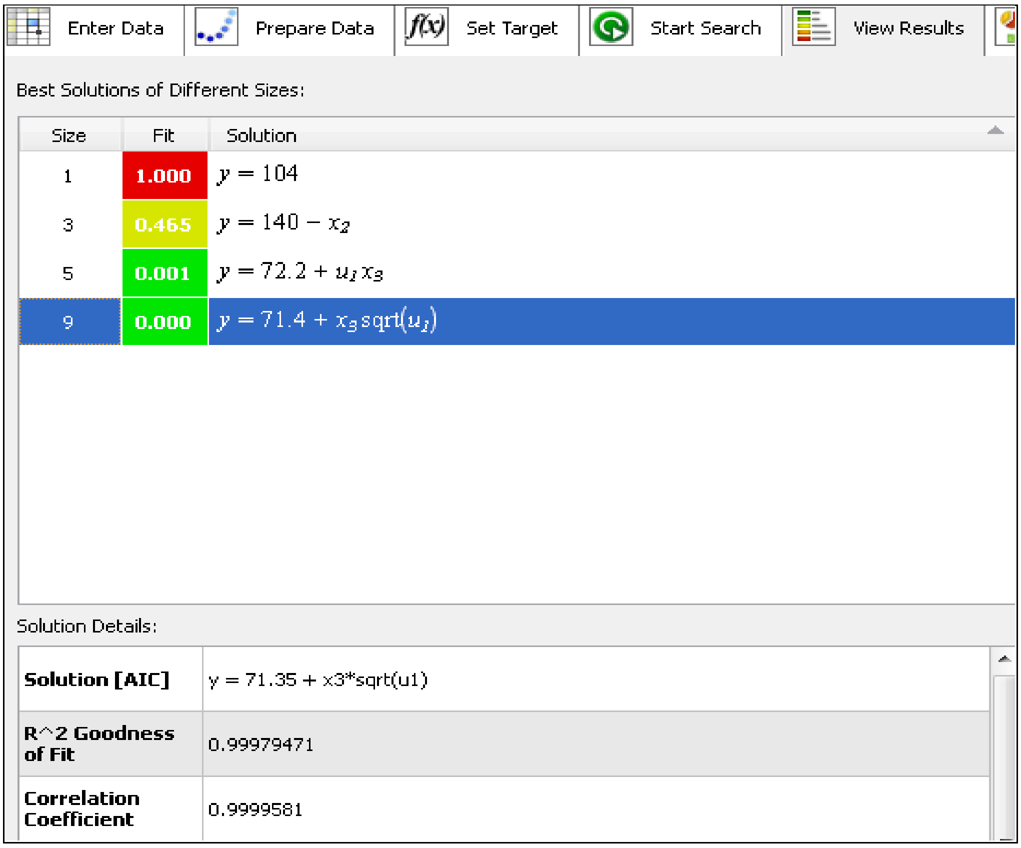


Project: artificial1

	A	B	C	D	E
desc	The generated new matrix for cluster 3	variable 1	variable 2	variable 3	output value
var	u_3	x_1	x_2	x_3	y
1	0.9982	15.00	56.00	10.33	58.77
2	0.9994	14.30	55.00	12.43	58.93
3	0.0132	9.98	8.60	50.00	120.40
4	0.0152	9.56	7.90	51.20	122.00
5	0.0861	10.12	30.10	49.80	123.50
6	0.0824	11.00	29.90	50.44	120.18
7	0.0158	8.77	7.80	51.87	131.11
8	0.0013	23.80	86.50	45.87	75.00
9	0.0027	26.23	89.00	44.90	73.20
10	0.0022	24.76	85.40	43.12	76.00
11					

Figure 5.13 Eureqa-formulize screenshot of the artificial dataset for cluster 3

Step 4: After the new matrixes are generated,with the usage of different parameters (such as addition, subtraction, division, cosine) Eureqa Formulize software program is run and the equations that describe the data most appropriately is tried to be found out. In the Figure 5.14 the obtained results and selected equation are shown for the first cluster. From the Figure 5.14 it could be seen that most appropriate formula is found as $y = 7.15 + x_3 sqrt(u_1)$.



Best Solutions of Different Sizes:

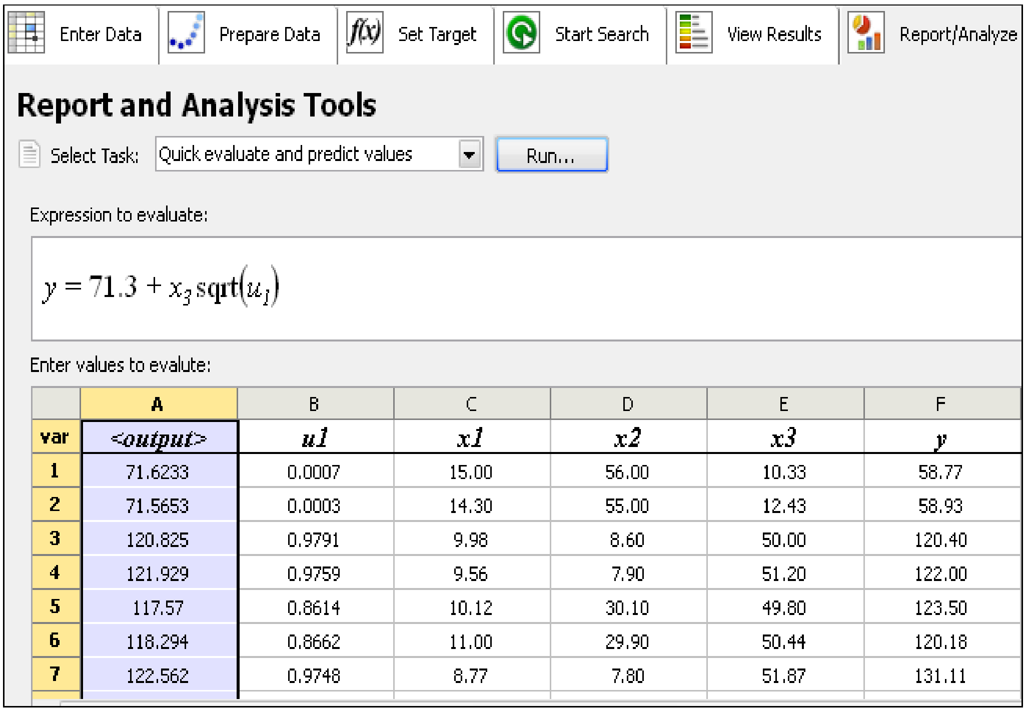
Size	Fit	Solution
1	1.000	$y = 104$
3	0.465	$y = 140 - x_2$
5	0.001	$y = 72.2 + u_1 x_3$
9	0.000	$y = 71.4 + x_3\,\mathrm{sqrt}(u_1)$

Solution Details:

Solution [AIC]	y = 71.35 + x3*sqrt(u1)
R^2 Goodness of Fit	0.99979471
Correlation Coefficient	0.9999581

Figure 5.14 The screenshot of the results page for cluster 1 and selected equation

Step 5: In this step, according to best fitting equations, the output values are predicted for each cluster. The screenshot of the predicted output values are show in Figure 5.15, Figure 5.16 and Figure 5.17 respectively.



Report and Analysis Tools

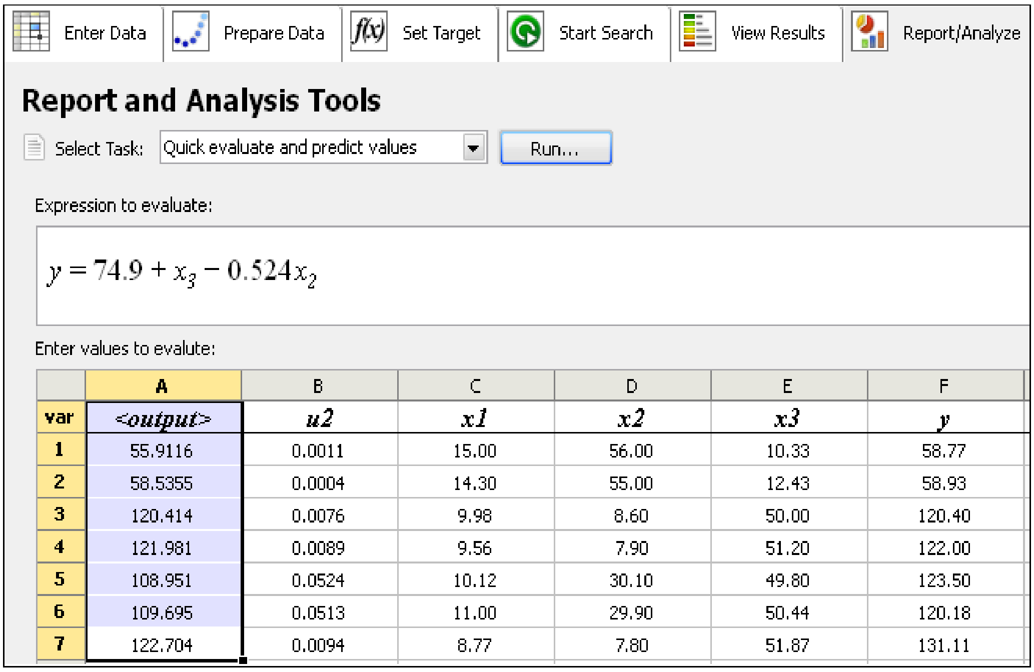
Expression to evaluate:

$$y = 71.3 + x_3 \,\mathrm{sqrt}(u_1)$$

Enter values to evalute:

	A	B	C	D	E	F
var	<output>	u1	x1	x2	x3	y
1	71.6233	0.0007	15.00	56.00	10.33	58.77
2	71.5653	0.0003	14.30	55.00	12.43	58.93
3	120.825	0.9791	9.98	8.60	50.00	120.40
4	121.929	0.9759	9.56	7.90	51.20	122.00
5	117.57	0.8614	10.12	30.10	49.80	123.50
6	118.294	0.8662	11.00	29.90	50.44	120.18
7	122.562	0.9748	8.77	7.80	51.87	131.11

Figure 5.15 Predicted output values of artificial dataset for cluster 1



Report and Analysis Tools

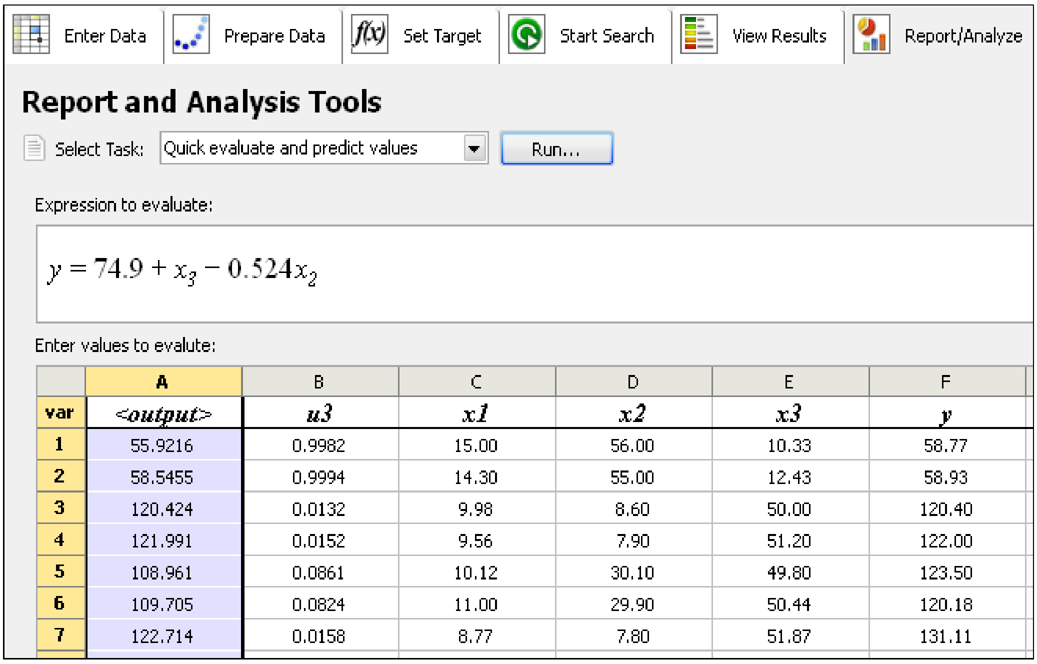
Expression to evaluate:

$$y = 74.9 + x_3 - 0.524x_2$$

Enter values to evalute:

	A	B	C	D	E	F
var	<output>	u2	x1	x2	x3	y
1	55.9116	0.0011	15.00	56.00	10.33	58.77
2	58.5355	0.0004	14.30	55.00	12.43	58.93
3	120.414	0.0076	9.98	8.60	50.00	120.40
4	121.981	0.0089	9.56	7.90	51.20	122.00
5	108.951	0.0524	10.12	30.10	49.80	123.50
6	109.695	0.0513	11.00	29.90	50.44	120.18
7	122.704	0.0094	8.77	7.80	51.87	131.11

Figure 5.16 Predicted output values of artificial dataset for cluster 2



Report and Analysis Tools

Select Task: Quick evaluate and predict values

Expression to evaluate:

$$y = 74.9 + x_3 - 0.524x_2$$

Enter values to evalute:

	A	B	C	D	E	F
var	<output>	*u3*	*x1*	*x2*	*x3*	*y*
1	55.9216	0.9982	15.00	56.00	10.33	58.77
2	58.5455	0.9994	14.30	55.00	12.43	58.93
3	120.424	0.0132	9.98	8.60	50.00	120.40
4	121.991	0.0152	9.56	7.90	51.20	122.00
5	108.961	0.0861	10.12	30.10	49.80	123.50
6	109.705	0.0824	11.00	29.90	50.44	120.18
7	122.714	0.0158	8.77	7.80	51.87	131.11

Figure 5.17 Predicted output values of artificial dataset for cluster 3

After all processes are finished we obtain “c” number of predicted values for each observation. The obtained predicted output values are shown in Table 5.6.

Table 5.6 Obtained predicted values for all clusters

	The predicted output values for all clusters		
	Cluster 1	**Cluster 2**	**Cluster 3**
$y_{1,i}$	71.6233	55.9116	55.9216
$y_{2,i}$	71.5653	58.5355	58.5455
$y_{3,i}$	120.825	120.414	120.424
$y_{4,i}$	121.929	121.981	121.991
$y_{5,i}$	117.57	108.951	108.961
$y_{6,i}$	118.294	109.695	109.705
$y_{7,i}$	122.562	122.704	122.714
$y_{8,i}$	72.3757	75.4726	75.4827
$y_{9,i}$	72.8392	73.1929	73.2029
$y_{10,i}$	72.6436	73.2989	73.3089

Step 6: **S**ame as in the fuzzy functions with LSE, finally single output values are calculated for each data vector by weighting predicted output values from each cluster with their corresponding membership values as shown in equation (5.6).

$$\hat{y}_k = \frac{\sum_i^c \mu_{ki} \hat{y}_{ki}}{\sum_i^c \mu_{ki}} \quad i = 1, \ldots, c, \qquad k = 1, \ldots, nd \tag{5.6}$$

For all observations the single final predicted output values are calculated as follows;

$$\hat{Y}_1 = \frac{(0.0007 * 71.6233 + 0.0011 * 55.9116 + 0.9982 * 55.9216)}{(0.0007 + 0.0011 + 0.9982)} = 55.93258$$

$$\hat{Y}_2 = \frac{(0.0003 * 71.5653 + 0.0004 * 58.5355 + 0.9994 * 58.5455)}{(0.0003 + 0.0004 + 0.9994)} = 58.55526$$

$$\hat{Y}_3 = \frac{(0.9791 * 120.825 + 0.0076 * 120.414 + 0.0132 * 120.424)}{(0.9791 + 0.0076 + 0.0132)} = 120.8045$$

$$\hat{Y}_4 = \frac{(0.9759 * 121.929 + 0.0089 * 121.981 + 0.0152 * 121.991)}{(0.9759 + 0.0089 + 0.0152)} = 121.9304$$

$$\hat{Y}_5 = \frac{(0.8614 * 117.57 + 0.0524 * 108.951 + 0.0861 * 108.961)}{(0.8614 + 0.0524 + 0.0861)} = 116.3654$$

$$\hat{Y}_6 = \frac{(0.8662 * 118.294 + 0.0513 * 109.695 + 0.0824 * 109.705)}{(0.8662 + 0.0513 + 0.0824)} = 117.1333$$

$$\hat{Y}_7 = \frac{(0.9748 * 122.562 + 0.0094 * 122.704 + 0.0158 * 122.714)}{(0.9748 + 0.0094 + 0.0158)} = 122.5657$$

$$\hat{Y}_8 = \frac{(0.0005 * 72.3757 + 0.9982 * 75.4726 + 0.0013 * 75.4827)}{(0.0005 + 0.9982 + 0.0013)} = 75.47106$$

$$\hat{Y}_9 = \frac{(0.0011 * 72.8392 + 0.9962 * 73.1929 + 0.0027 * 73.2029)}{(0.0011 + 0.9962 + 0.0027)} = 73.19254$$

$$\hat{Y}_{10} = \frac{(0.0009 * 72.6436 + 0.9969 * 73.2989 + 0.0022 * 73.3089)}{(0.0009 + 0.9969 + 0.0022)} = 73.29833$$

The weighted predicted output values are represented in Table 5.7.

Table 5.7 Obtained single predicted values for all observations

Predicted Values for Artificial Data
55.93258
58.55526
120.8045
121.9304
116.3654
117.1333
122.5657
75.47106
73.19254
73.29833

R-square value is found as 0.9813 for the numerical example with fuzzy functions with GP.

5.4 Conclusion

In this part of the study, genetic programming which forms the main points of the proposed modelis tried to be represented broadly. For that purpose, firstlygenetic algorithms which are robust search and optimization techniques and form the basis of genetic programming are reviewed briefly. Afterwards, the basis of the proposed model is introduced and its algorithm is explainedstep by step. Finally, with an example the steps of the algorithm are explained numerically in order to be sure that the algorithm is comprehended clearly.

In the following chapter, the datasets that are taken from the literature are applied to fuzzy functions with LSE and fuzzy functions with GP.Then the prediction performances of both models are compared based on the obtained results.

CHAPTER SIX
CASE STUDIES

6.1 Introduction

In this chapter, 8 datasets that are taken from *Uci Machine Learning Repository*(UCI Machine Learning Repository)are applied for the purpose of evaluating the performance of fuzzy functions with LSE and the proposed model, fuzzy functions with GP. Afterwards the results of both models are compared with each other and the prediction performance of the proposed model is assessed. For the evaluation and comparison process, the flow of chapter is as follows; initially the datasets are introduced briefly in the next section. Then cluster validity indexes are determined in order to find out the best partitions for each dataset.For this study it is decided to choose 3 different cluster numbers that are thought to represent the best partitions. Then by executing the FCM algorithm,according to these cluster numbersmembership values are obtained. Afterwards, adding the membership values and their different transformations as new variables, fuzzy functions with LSE and fuzzy functions with GP methods are applied to these datasets. Then according to R-square results both models are compared and evaluated both in itself and between each other.

6.2 Introduction of the Datasets

6.2.1 Abalone Dataset

Abalone data is about predicting the age of abalone from physical measurements. The number of instances is 4177 and number of attributes is 8. In the original dataset the first attribute is nominal and indicates the sex of abalone whether female, male or infant. Since, in this study regression equation is used, the first linguistic attribute “sex” is not taken as a parameter. In the original data the aim is to predict the ring of abalones, in other saying predicting the age of abalones. But in this study, number of rings is used as an input parameter and shell weight is tried to be predicted. The parameters of the dataset are depicted in Table 6.1.

Table 6.1 Abalone dataset parameters

Input parameters	Output parameter	Type of data
Length	Shell weight	Classification type data
Diameter		
Height		
Whole weight		
Shucked weight		
Viscera weight		
Rings		

6.2.2 Auto-Mpg Dataset

Auto-mpg data set deals with city fuel consumption in miles per consumption. In this data set originally there are 9 attributes; 1 attribute is output parameter and the other remaining attributes are input parameters. But due to using regression analysis in this study the last linguistic attribute “car name” removed from the data set. After the 6 observations which have missing values in horsepower variable have removed from the dataset the remained number of observation is 392. The parameters of the auto-mpg dataset are shown in Table 6.2.

Table 6.2 Auto-mpg dataset parameters

Input parameters	Output parameter	Type of data
Cylinders	Mpg	Regression type data
Displacement		
Horsepower		
Weight		
Acceleration		
Model year		
Origin		

6.2.3 Concrete Compressive Strength Dataset

Concrete compressive strength dataset is a regression type problem. In this data, concrete compressive strength is tried to be predicted with some different ingredientsunder some conditions. In the datasets there are 1030 instances and no missing values. There are 9 attributes and concrete compressive strength is the output variable. The parameters of the datasets are represented in Table 6.3.

Table 6.3 Concrete compressive dataset parameters

Input parameters	**Output parameter**	**Type of data**
Cement Blast Furnace Slag Slag Fly Ash Water Super plasticizer Coarse Aggregate Fine Aggregate Age	Concrete compressive strength	Regression type data

6.2.4 Ecoli Dataset

In ecoli dataset there are no missing values. The dataset consist of 336 instances and in the original data there are 8 attributes. But in our study 1 linguistic attribute is removed from the data in order to fit regression analysis.

Ecoli dataset is a classification type data. Therefore to be able to use fuzzy functions one attribute is chosen as the output parameter. The attributes are listed in Table 6.4.

Table6.4 Ecoli dataset parameters

Input parameters	Output parameter	Type of data
Mcg Gvh Lip Chg Aac Alm1	Alm2	Classification type data

6.2.5 Glass Dataset

Glass identification dataset is an example of classification type problem and consisting of ten parameters. In the original dataset, the last parameter is the type of glass and indicates cluster numbers. Due to using regression function in fuzzy functions,last parameter is removed from the dataset and refractive index (RI) is chosen as output parameter. Remaining parameters are used as input parameters. In glass data there are 214 observations, 8 input variables and 1 outputparameter. These parameters are represented in Table 6.5.

Table 6.5 Glass dataset parameters

Input parameters	Output parameter	Type of data
Na: Sodium Mg: Magnesium Al: Aluminum Si: Silicon K: Potassium Ca: Calcium Ba: Barium Fe: Iron	RI: refractive index	Classification type data

6.2.6 Housing Dataset

Housing data is about housing values in the suburbs of Boston. There are 506 observations and 14 attributes, 13 of them are continuous attributes and the remaining observation is a binary valued attribute. There are no missing values. The attributes of the housing data are explained in Table 6.6.

Table 6.6 Housing data parameters

Input parameters		**Output parameter**	**Type of data**
Crim	Age	Medv	Regression type data
Zn	Dis		
Indus	Rad		
Chas	Tax		
Rox	Ptratio		
Rm	B		
Lstat			

6.2.7 Iris Dataset

Fisher's Iris dataset is about cluster analysis and data mining. There are no missing values in the dataset. This dataset consist of 3 clusters which represent the species of Iris data (Iris Setosa, Iris Versicolour and Iris Virginica). Each of these clusters has 50 instances. Each species of Iris data contains 4 attributes. These are introduced in Table 6.7. Iris data is a classification type data and for this study first three attributes are chosen as input parameters and the last one which is petal width is chosen as output parameter. The parameters of the iris data are shown in Table 6.7.

Table 6.7 Iris dataset parameters

Input parameters	Output parameter	Type of data
sepal length sepal width petal length	petal width continuous	Classification type data

6.2.8 Wine dataset

The wine dataset is about chemical analysis of wines grown in the same region in Italy and derived from three different cultivars. The dataset is classification type data and contains 178 observations.The dataset consists of 13 attributes which are depicted in Table 6.8. For this study one of them is chosen as an output variable and remaining attributes are taken as input variables.

Table 6.8 Wine dataset parameters

Input parameters	Output parameter	Type of data
Alcohol Malic acid Ash Alcalinity of ash Magnesium Total phenols Flavanoids Nonflavanoid phenols Proanthocyanins Color intensity Hue Proline	OD280/OD315 of diluted wines	Classification type data

6.3 Defining the Best Possible Number of Clusters

In this section optimum number of clusters are tried to be found out. As it was mentioned in previous chapters, in order to find out the optimum number of clusters, partition coefficient (PC), classification entropy (CE), partition index (SC), separation index (S), Xie and Beni (XB) index, Dunn index and Alternative Dunn index are used. These cluster validity indexes are found via “fuzzy clustering and data analysis toolbox” which is prepared for using with Matlabby Balasko, Abonyi and Feil(2005).Since the monotonic decreasing of partition coefficient with c and monotonic increasing of classification entropy with c, it could be said that these validity indexes are not connected with data directly. Due to this reason, partition coefficient and classification entropy are not taken into consideration and are slurred over.

Balasko et al (2005), in their study indicated that no validation index could be reliable alone and due to this reason the optimum cluster number should be detected with the comparison of all cluster validity results. Also they indicated that, when the differences between the values of a validation index are minor, choosing the less cluster numbers are better.

6.3.1 Optimum Number of Clusters for Abalone Dataset

When we look at the graph in Figure 6.1, the decrease at cluster number 3 for partition (SC) index and also for separation index (S) can be seen clearly. Then separation index values continue to decrease until cluster number 6 and then continue to decrease monotonically. Due to that fact optimum number of clusters could be thought as 3, 4 and 5. For Xie and Beni (XB) index, there is a decline at cluster number 3, then it increases at cluster number 5 and again it decreases at cluster number 7 and continues to decrease. And finally reaches the minimum value at cluster number 9. Dunn index reaches the maximum values at cluster number 3 and 5. ADI index reaches minimum values at 3 and 9. By considering that fewer clusters are better, and considering all these results, we decided to take 3, 4 and 5 as optimum cluster numbers.

Table 6.9 Cluster validity index results for abalone data

	Cluster number								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PC ↑	0.72693	0.70150	0.62910	0.57953	0.52149	0.50247	0.47780	0.46963	0.44463
CE ↓	0.42915	0.54171	0.70932	0.84101	0.98516	1.07110	1.13144	1.19162	1.27245
SC ↓	2.45403	**1.18303**	**1.02855**	0.96543	0.96431	0.93809	0.73777	0.75702	0.77150
S ↓	0.00059	**0.00044**	**0.00038**	**0.00037**	0.00038	0.00035	0.00029	0.00029	0.00029
XB ↓	4.73834	**4.63486**	4.88133	**4.59593**	4.71698	**3.75419**	**2.52050**	**2.30811**	2.82300
DI ↑	0.00621	0.00682	0.00516	0.00669	0.00603	0.00520	0.00621	0.00573	0.00617
ADI ↓	0.03903	0.00091	0.00923	0.00818	0.00423	0.00399	0.00045	0.00019	0.00080

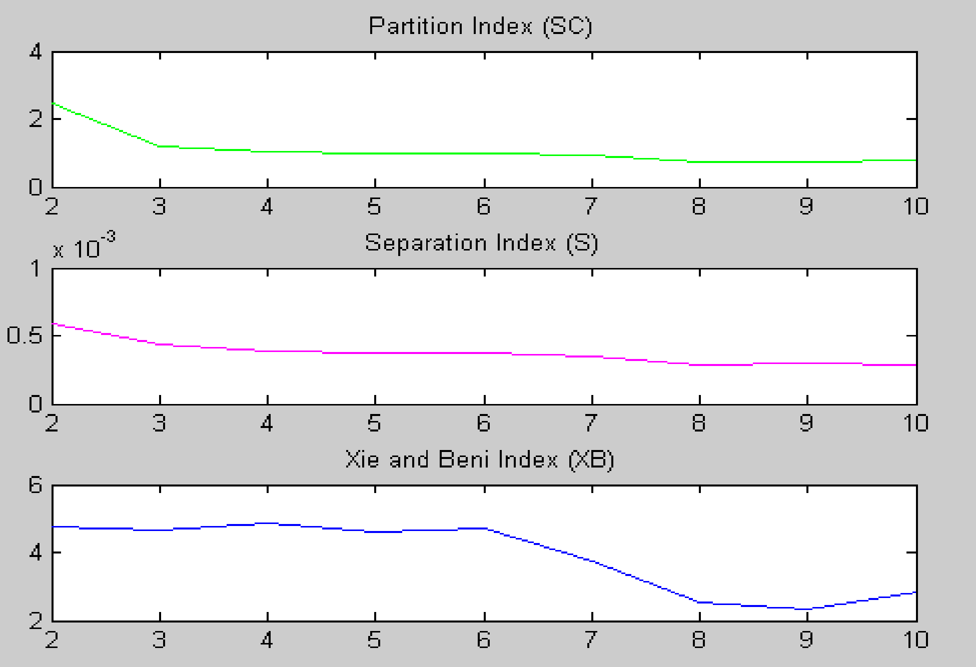


Figure 6.1 Values of Partition Index, Separation Index and Xie and Beni Index for abalonedataset

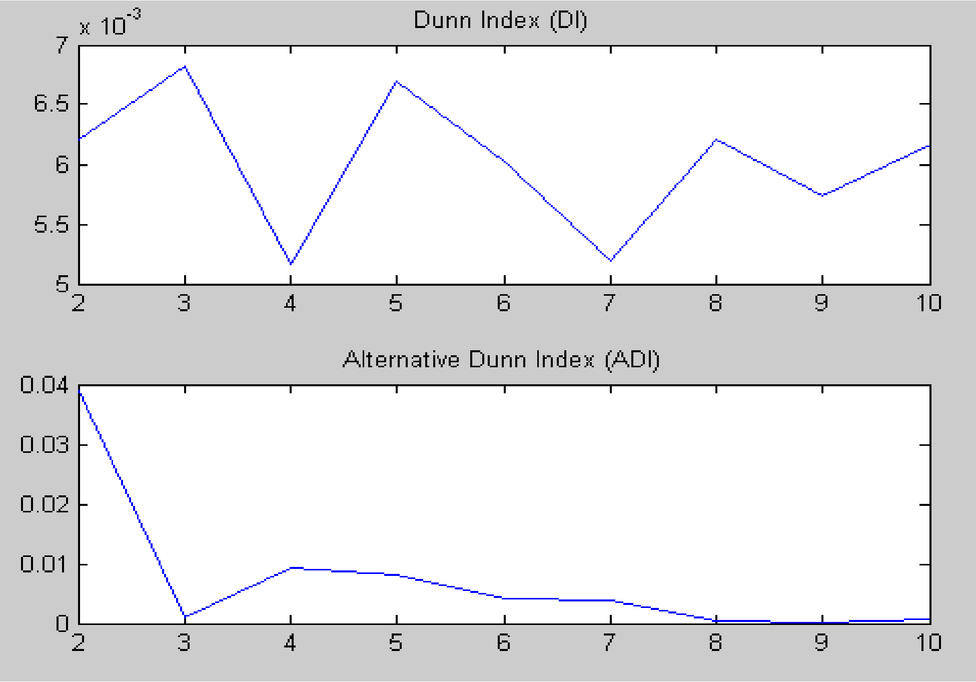


Figure 6.2 Values of Dunn Index and Alternative Dunn Index for abalone dataset

6.3.2 Optimum Number of Clusters for Auto-mpg Dataset

If we interpret the Table 6.10 and graphs in Figure 6.3 and 6.4, partition index reaches minimum values at 3, 6 and 8. Separation index reaches minimum value at 5. Xie and Beni index also reaches minimum values at 3, 8, 9 and 10. If we look at the graph in Figure 6.4, the optimum number of clusters according to Dunn index is 6 and 8 at which the maximum valuesare reached. Considering all these results the optimum number of clusters for auto-mpg data are taken as 3, 5 and 8.

Table 6.10 Cluster validity index results for auto-mpg data

	Cluster number								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PC ↑	0.92326	0.88486	0.85638	0.84335	0.82560	0.80787	0.79443	0.78540	0.77179
CE ↓	0.43030	0.66421	0.84505	0.94050	1.05805	1.17789	1.26968	1.33852	1.43106
SC ↓	1.42870	**1.15714**	1.32095	1.29934	1.21520	1.44681	1.39096	1.51760	1.62233
S ↓	**0.00364**	0.00428	0.00503	**0.00442**	0.00489	0.00495	0.00509	0.00517	0.00570
XB ↓	2.50645	**1.74099**	2.00629	2.13677	2.06232	1.78113	**1.68870**	1.60234	1.37205
DI ↑	0.27587	0.03632	0.05664	0.03633	**0.07435**	0.05820	**0.07308**	0.06845	0.07011
ADI ↓	0.03403	0.00187	0.00212	0.00124	0.00131	0.00040	0.00211	0.00011	0.00001

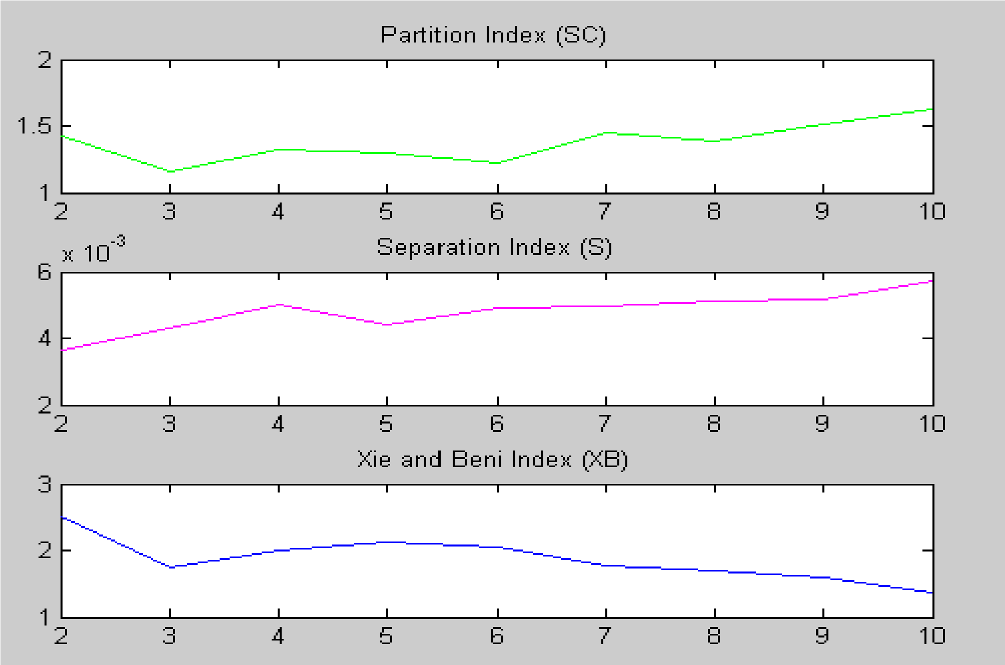


Figure 6.3 Values of Partition Index, Separation Index and Xie and Beni Index for auto-mpgdataset

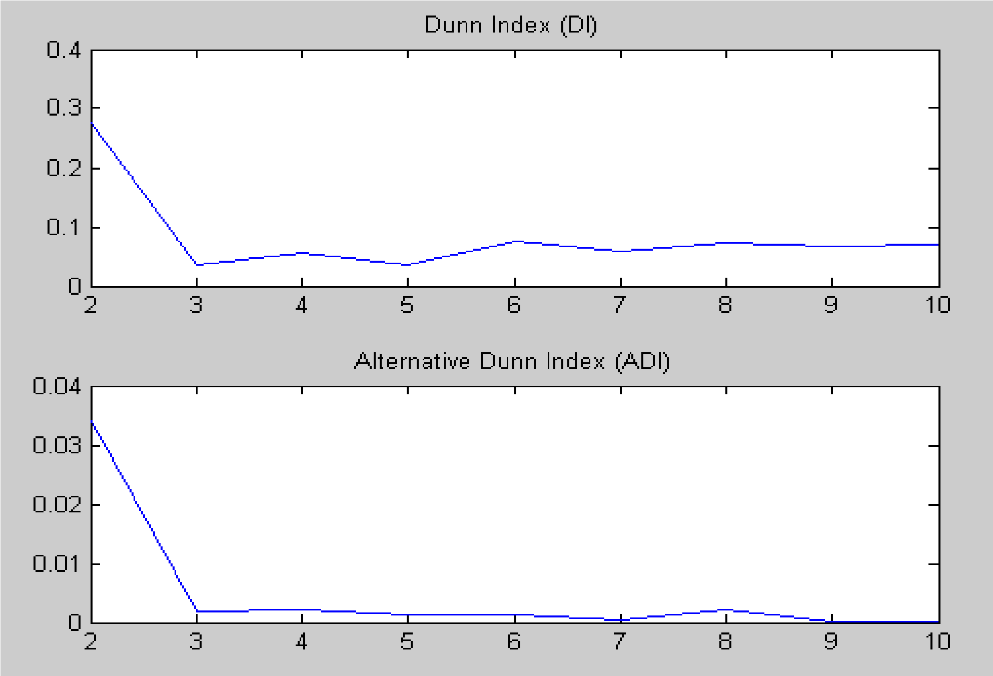


Figure 6.4 Values of Dunn Index and Alternative Dunn Index for auto-mpg dataset

6.3.3 Optimum Number of Clusters for Concrete Dataset

When we look at the results in Table 6.11 and graphs in Figure 6.5 and Figure 6.6,for concrete dataset each the validity index points different cluster numbers. The validity index values reaches minimum at 5, 7 and 9 for partition index. At cluster number 5 the value is decreasing, at 7 the value continues to increasing, but at 9 it is decreasing again. Because of that it could not be wrong to say that 5 and 9 is more appropriate as optimum number of clusters. For separation index, values reaches minimum at 5, 7 and 9. Because of that the values are hardly decreasing at cluster number 5 and 9,same as partition index 5 and 9 is more appropriate for separation index. Dunn index reaches at 4 and 8 to maximum numbers. In conclusion for concrete dataset the optimal cluster numbers are chosen as 4, 5 and 9.

Table 6.11 Cluster validity index results for concrete dataset

	Cluster number								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PC ↑	0.89149	0.83480	0.79526	0.77244	0.74806	0.72739	0.70682	0.69583	0.68205
CE ↓	0.59131	0.93227	1.18349	1.34200	1.50807	1.65143	1.79521	1.87933	1.98172
SC ↓	6.45980	5.32736	5.49949	**4.46829**	5.04430	**5.17987**	6.20888	**5.43847**	5.95852
S ↓	0.00627	**0.00596**	0.00769	**0.00536**	0.00729	**0.00693**	0.00892	**0.00675**	0.00840
XB ↓	1.42823	1.21890	1.12854	0.94961	0.93879	0.80727	**0.67716**	0.71748	**0.65987**
DI ↑	**0.18651**	0.04121	**0.05687**	0.01032	0.01032	0.01071	**0.02798**	0.00655	**0.00756**
ADI ↓	0.00480	0.00370	0.00253	0.00209	0.00149	0.00028	0.00024	**0.00005**	**0.00006**

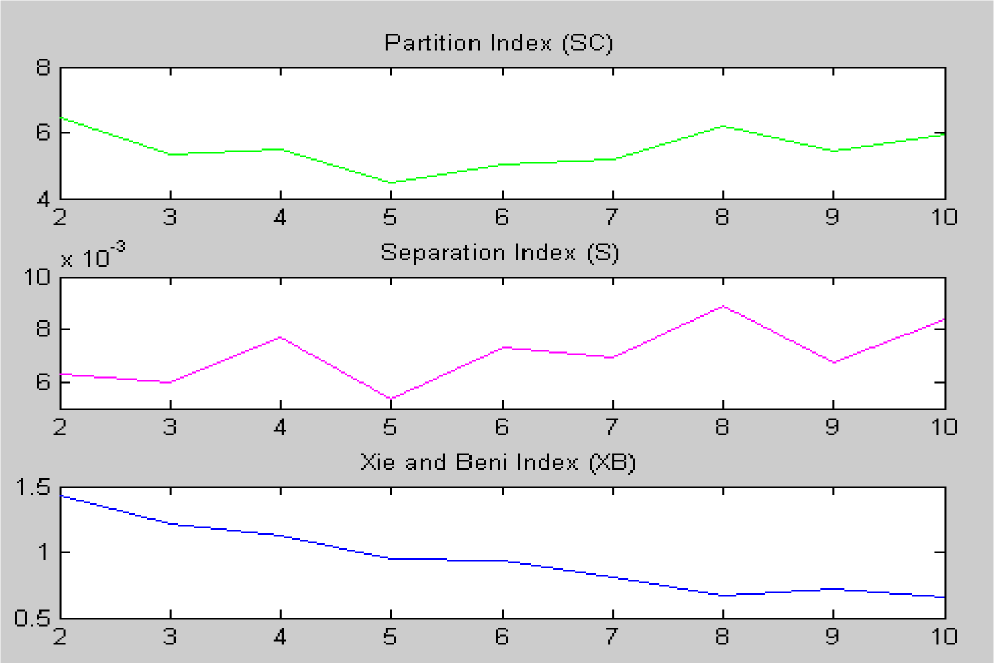


Figure 6.5 Values of Partition Index, Separation Index and Xie and Beni Index for concrete dataset

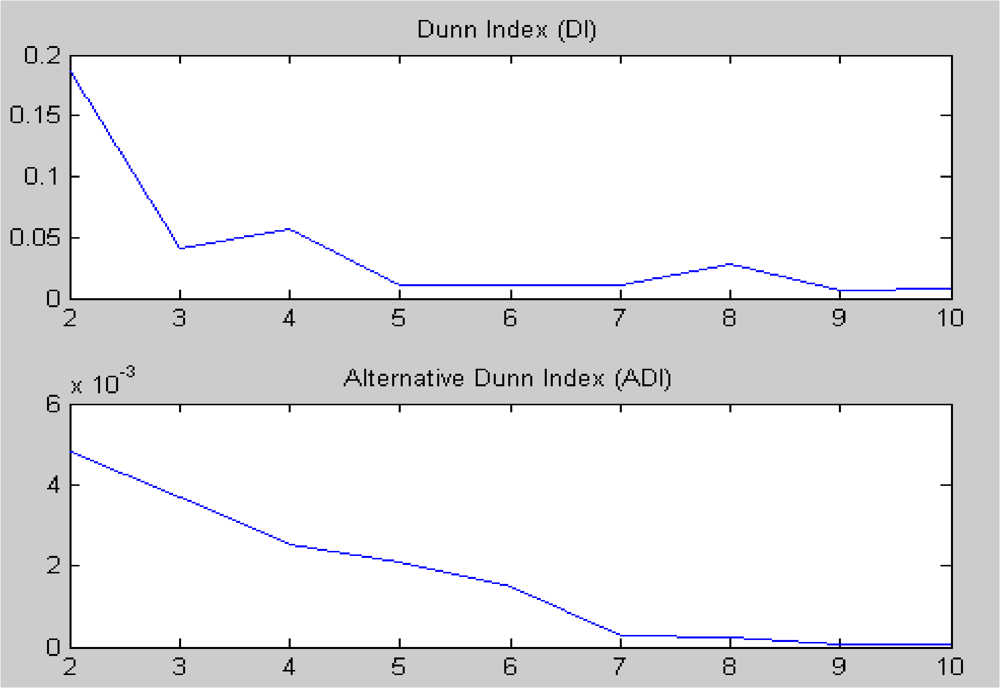


Figure 6.6 Values of Dunn Index and Alternative Dunn Index for concrete dataset

6.3.4 Optimum Number of Clusters for Ecoli Dataset

To interpret the Table 6.12, Figure 6.7 and Figure 6.8 for partition index, optimum cluster numbers are 3, 5, 9 and 10. For separation index the results reaches minimum degrees at 3, 5, 9 and 10 and at cluster number 3 and 5, the results are decreasing suddenly. XB index and ADI values are decreasing monotonically, because of that we did not define any cluster number for XB and ADI. According to Dunn index, the optimum values are 4 and 6 which are reachingto maximum degrees. According to these results for ecoli data optimum numbers of clusters are chosen as 4, 5 and 6.

Table 6.12 Cluster validity index results for ecoli dataset

	Cluster number								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PC ↑	0.69747	0.61029	0.47734	0.42445	0.36728	0.33283	0.30435	0.28189	0.25804
CE ↓	0.46722	0.69649	0.97881	1.13147	1.30247	1.43599	1.55803	1.64357	1.74292
SC ↓	2.78739	**1.74669**	1.98565	**1.47959**	1.59222	1.57655	1.68135	**1.42028**	**1.37807**
S ↓	0.00830	**0.00616**	0.00912	**0.00582**	0.00703	0.00697	0.00814	**0.00654**	**0.00579**
XB ↓	3.15834	2.31858	2.13907	1.74446	1.38030	1.25782	1.04028	0.91969	0.86232
DI ↑	**0.04830**	0.03694	**0.04641**	0.02984	**0.03928**	0.02984	0.03015	**0.03284**	0.01606
ADI↓	0.06114	0.00419	0.00227	0.00118	**0.00094**	**0.00016**	**0.00002**	**0.00007**	**0.00019**

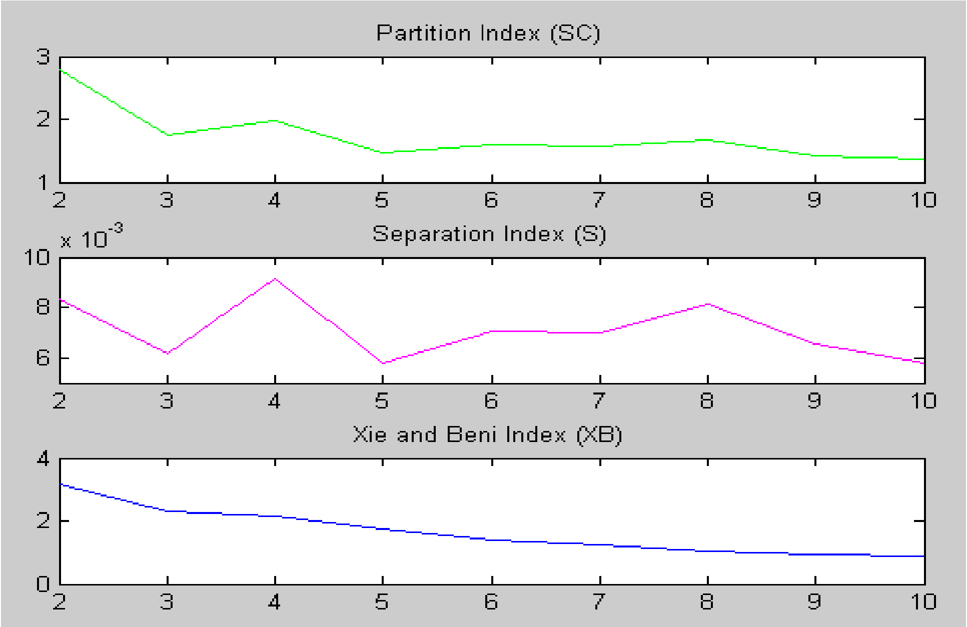


Figure 6.7 Values of Partition Index, Separation Index and Xie and Beni Index for ecolidataset

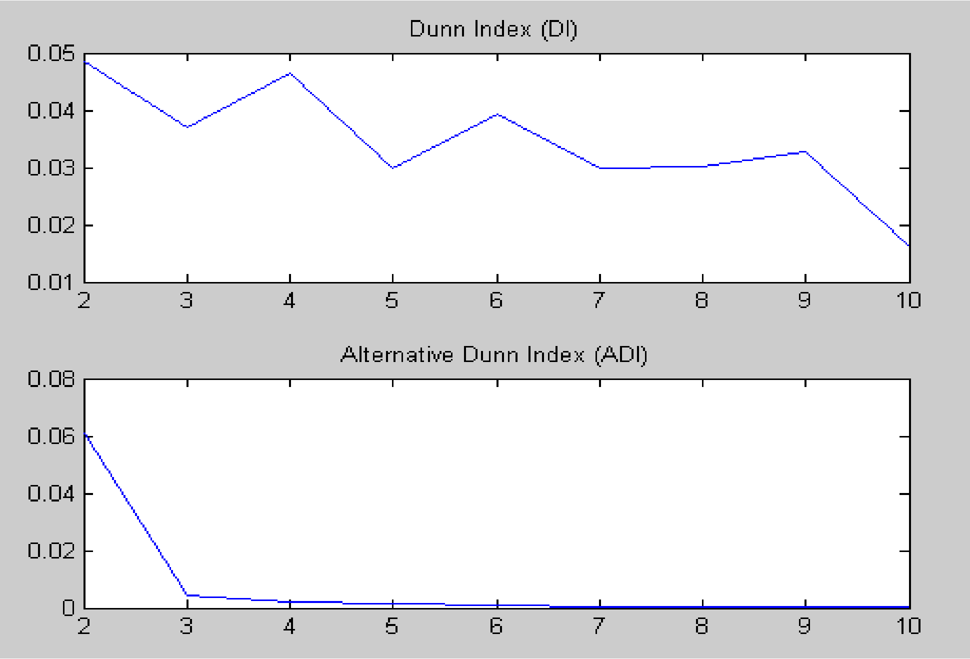


Figure 6.8 Values of Dunn Index and Alternative Dunn Index for ecoli dataset

6.3.5 Optimum Number of Clusters for Glass Dataset

If we look at the graph in Figure 6.10, for Dunn index and also Alternative Dunn index there is no certain values that we can say this is the best partition for glass data. For this reason, we take no account of DI and ADI. But as it can be seen in Figure 6.9 and in Table 6.13, partition index takes the minimum values at cluster number 7 and 9. Separation index takes the minimum values at 7 and 8. XB index also takes the minimum value at 7 and 9 same as partition index. According to these results, we take the optimum cluster numbers as 7, 8 and 9.

Table 6.13 Cluster validity index results for glass dataset

	Cluster number								
	2	3	4	5	6	7	8	9	10
PC ↑	0.82454	0.68357	0.64927	0.55926	0.48555	0.50252	0.4557	0.44492	0.40884
CE ↓	0.33793	0.63226	0.75683	0.97317	1.17612	1.18156	1.31015	1.38023	1.48637
SC ↓	1.88266	1.47443	1.32515	1.16045	1.1976	**1.41284**	0.99068	**1.15042**	0.89998
S ↓	0.00806	0.00792	**0.00666**	**0.00662**	0.00851	**0.00609**	**0.00606**	0.00908	0.00643
XB ↓	1.78527	3.32253	2.68297	1.41631	1.16428	**1.10712**	1.18523	**1.0789**	1.18014
DI ↑	0.11589	0.0326	0.0326	0.01517	0.02736	0.01491	0.01808	0.01803	0.01517
ADI ↓	0.07706	0.00041	0.0002	0.00013	6.9E-06	7.7E-05	0.00012	7.4E-05	0.00011

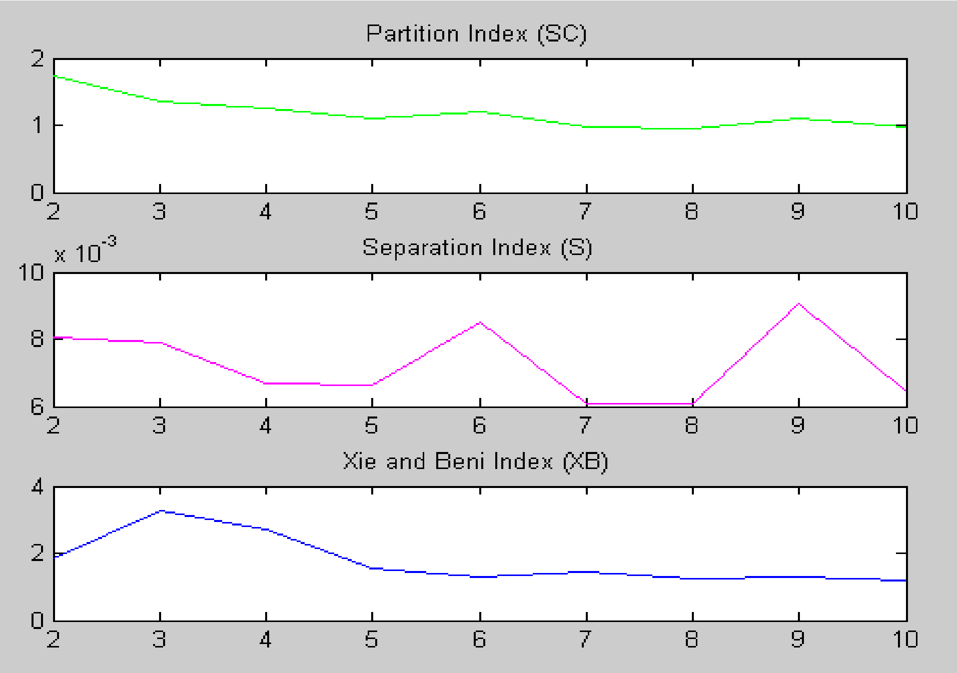


Figure 6.9 Values of Partition Index, Separation Index and Xie and Beni Index for glass dataset

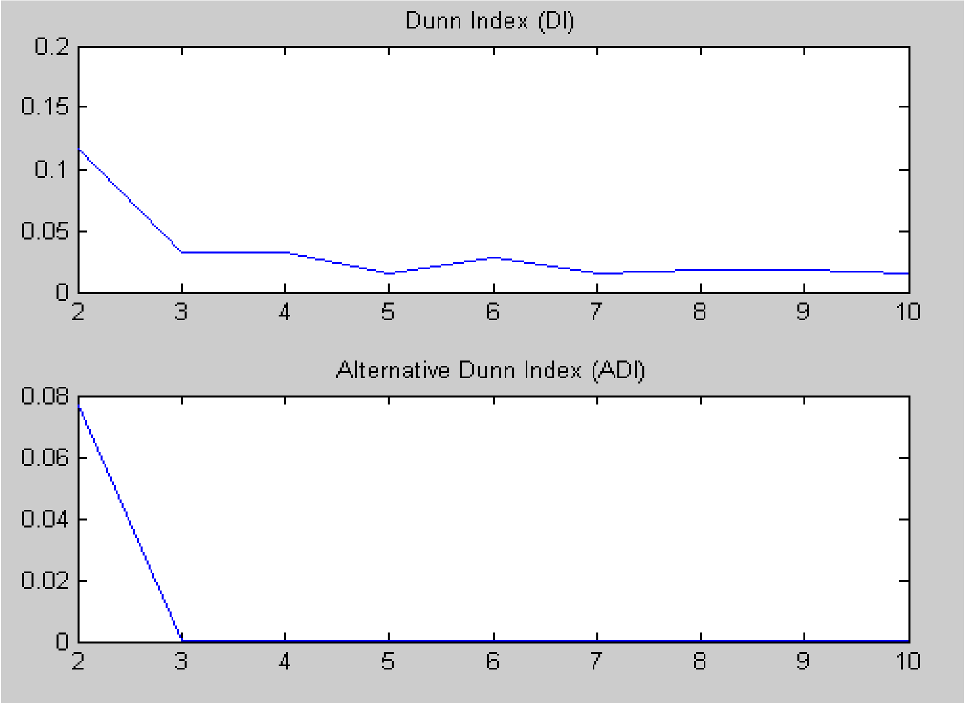


Figure 6.10 Values of Dunn Index and Alternative Dunn Index for glass dataset

6.3.6 Optimum Number of Clusters for Housing Dataset

For XB index, there is not a certain value and the index is monotonically decreasing. When we look at the obtained graphs in Figure 6.11 and Figure 6.12, we can see that at the points of 3, 6 and 8 partition index reaches minimum values. For separation index, minimum values are obtained at cluster number 6 and 8 too. And for Dunn index, the value increasing at 6 and takes the second largest value at cluster number 6. Also Dunn index reaches minimum value at 6. According to these results, optimum cluster numbers are taken for 3, 6 and 8.

Table 6.14 Cluster validity index results for housing dataset

	Cluster number								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PC ↑	0.92249	0.86388	0.81936	0.78648	0.77749	0.75412	0.73718	0.72243	0.70750
CE ↓	0.43440	0.77451	1.04642	1.25767	1.32877	1.48540	1.59837	1.71267	1.82102
SC ↓	1.50632	**1.53431**	1.97621	2.41764	**1.51946**	1.78438	**1.63370**	2.18032	2.49230
S ↓	0.00298	0.00443	0.00556	0.00673	**0.00451**	0.00518	**0.00477**	0.00631	0.00710
XB ↓	1.81961	1.80884	1.32860	1.06105	1.01079	0.81323	0.72453	0.69892	0.65451
DI ↑	0.24156	0.04821	0.05323	0.03517	**0.06439**	0.03388	0.03544	0.03705	0.02623
ADI ↓	0.03112	**0.00453**	0.00261	0.00199	**0.00039**	0.00277	**0.00092**	**0.00021**	**0.00020**

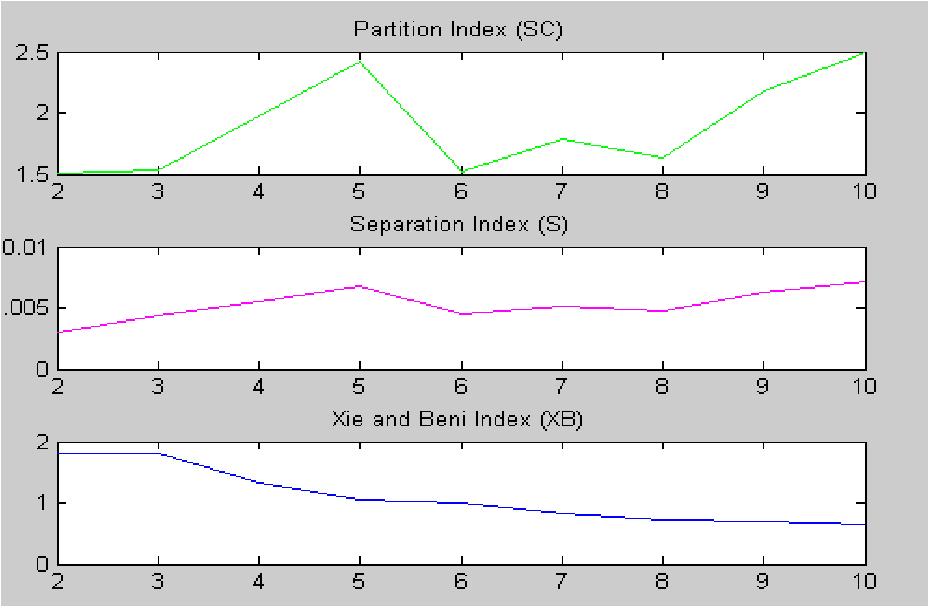


Figure 6.11 Values of Partition Index, Separation Index and Xie and Beni Index forhousing dataset

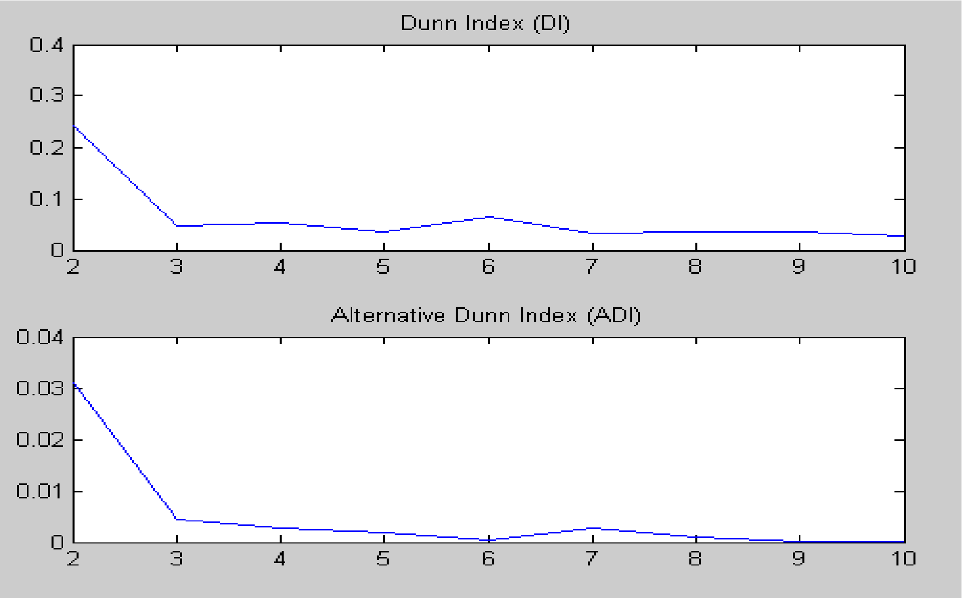


Figure 6.12 Values of Dunn Index and Alternative Dunn Index for housing dataset

6.3.7 Optimum Number of Clusters for Iris Dataset

When we look at Table 6.15, Figure 6.13 and Figure 6.14, partition index reaches minimum values at 3, 7 and 9for iris dataset. Separation index takes the minimum values at 2, 8 and 9. XB index value is decreasing at 4 and then continues to decrease monotonically. For Dunn Index the maximum value is obtained at cluster number 2. By considering that the minimum value is better; according to these results we chose the optimum number of clusters as 2, 3 and 4.

Table 6.15 Cluster validity index results for iris dataset

	Cluster number								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PC ↑	0.84878	0.73117	0.63672	0.59296	0.54608	0.52462	0.49546	0.48653	0.46682
CE ↓	0.26321	0.48860	0.69020	0.81734	0.95148	1.00088	1.10019	1.12444	1.19366
SC ↓	0.99076	**0.88489**	0.97520	0.93921	1.04914	**0.68642**	0.75822	**0.52489**	**0.51289**
S ↓	**0.00661**	0.00855	0.00899	0.00987	0.01053	0.00737	0.00774	**0.00541**	**0.00549**
XB ↓	5.97166	7.96675	**4.33773**	3.87654	3.44132	1.83890	1.78263	1.43446	1.44847
DI ↑	**0.10744**	0.05733	0.03618	0.05345	0.06936	0.05445	0.05445	0.05445	0.05445
ADI↓	0.01049	0.00632	0.00422	0.00294	0.00075	0.00120	0.00201	0.00061	0.00001

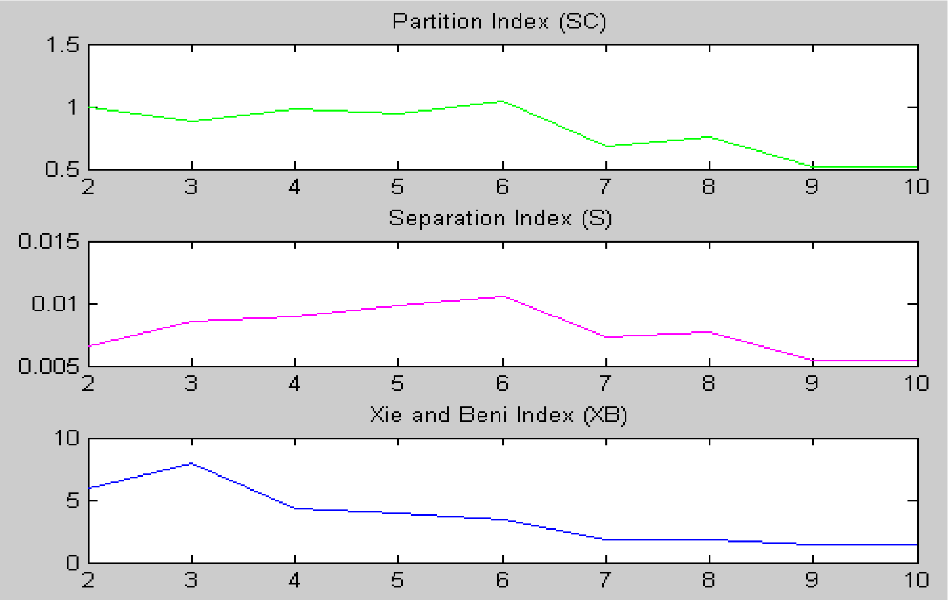


Figure 6.13 Values of Partition Index, Separation Index and Xie and Beni Index for iris dataset

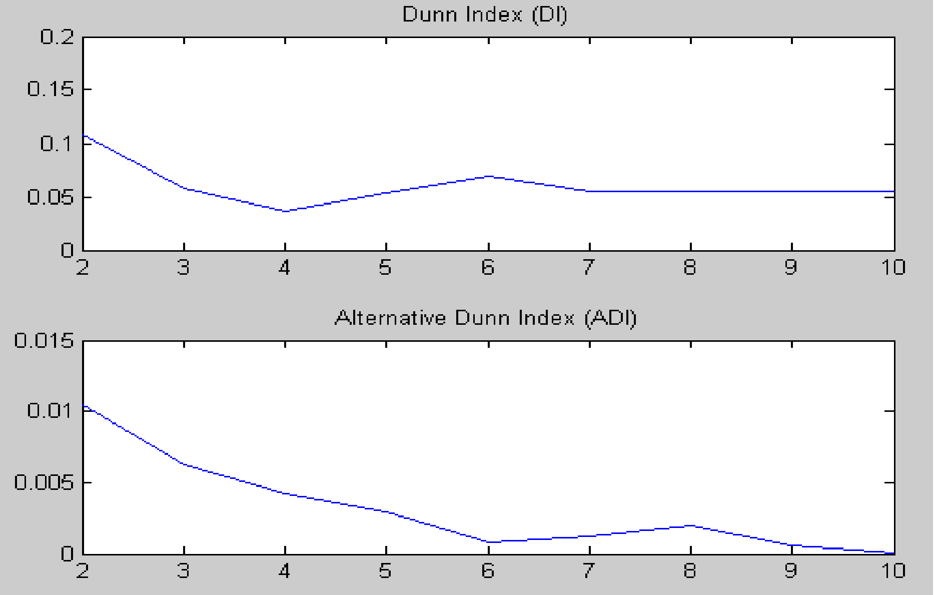


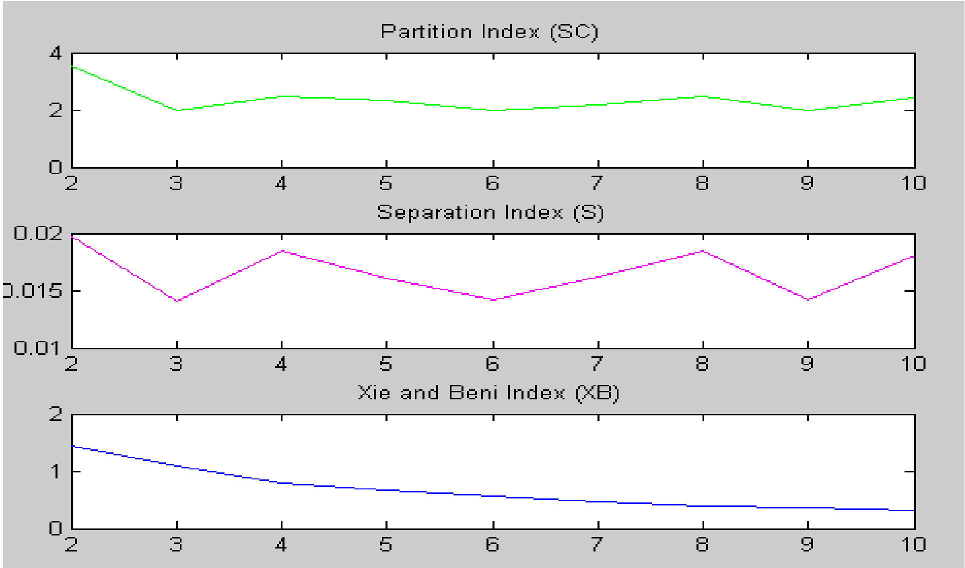
Figure 6.14 Values of Dunn Index and Alternative Dunn Index for iris dataset

6.3.8 Optimum Number of Clusters for Wine Dataset

As we can see from Table 6.16 and the graph in Figure 6.15, partition index reaches the minimum value at cluster number 3, 6 and 9. Separation index also reaches the minimum value at cluster number 3, 6 and 9. XB index values continue to decrease monotonically while the number of cluster is increasing, because of that for XB index the optimum number of clusters cannot be decided clearly. When we look at the Figure 6.16, Dunn index takes the maximum value at cluster number 4 and 6. For ADI minimum values are obtained at cluster number 5, 6, 8 and 9. Eventually for wine dataset we decided to take the optimum cluster numbers as 3, 4 and 6.

Table 6.16 Cluster validity index results for wine dataset

	Cluster number								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PC ↑	0.60770	0.48865	0.36502	0.29222	0.24520	0.20983	0.18251	0.16403	0.14515
CE ↓	0.57794	0.87670	1.17730	1.39949	1.56894	1.72742	1.87044	1.97348	2.09529
SC ↓	3.50236	**1.98958**	2.45271	2.32039	**1.99495**	2.16858	2.45271	**1.99941**	2.43172
S ↓	0.01968	**0.01412**	0.01844	0.01610	**0.01414**	0.01617	0.01843	**0.01416**	0.01807
XB ↓	1.43259	1.09641	0.79604	0.67052	0.55346	0.46064	0.39802	0.36756	0.31865
DI ↑	**0.15643**	0.15001	**0.17403**	0.13744	**0.15368**	0.14628	0.14440	0.14628	0.12509
ADI↓	0.01013	0.01619	0.01492	**0.00249**	**0.00129**	0.00700	**0.00068**	**0.00024**	0.00118


Partition Index (SC)

Separation Index (S)

Xie and Beni Index (XB)

Figure 6.15 Values of Partition Index, Separation Index and Xie and Beni Index for winedataset

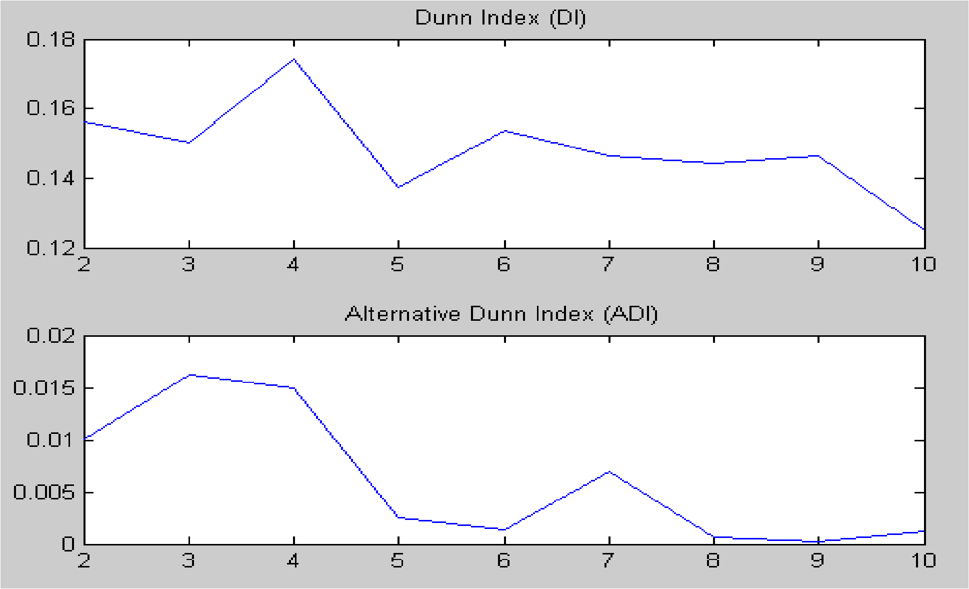


Figure 6.16 Values of Dunn Index and Alternative Dunn Index for wine dataset

6.4 Application of Fuzzy Functions with LSE

After the optimum numbers of clusters are decided for all datasets, for the next step, fuzzy functions approach with LSE is going to be implemented for all datasets. In order to apply fuzzy functions algorithm, Matlab program is used. All codes for fuzzy functions with LSE are written in Matlab and also R-square values are calculated in Matlab.

To be able to measure the effect of fuzzy functions, firstly regression analysis is implemented to the original datasets. Then for the next step membership values and some of their transformations are used for fuzzy functions with LSE. Respectively only membership values, membership values and two of their transformationsand finally membership values and four of their transformations are used as additional variables for fuzzy functions. Respectively these transformations are;$exp(\mu_i)$, $\exp(\mu_i)^2$and$exp(\mu_i)$, $\exp(\mu_i)^2$, $\frac{1}{exp\,(\mu_i)}$,$(\mu_i) * \log\big(1 + (\mu_i)\big)$.Afterwards the results obtained from these 3 different methods are compared. For each data, the algorithm is iterated six times in Matlab and the average R-square values are calculated. In the following section, obtained R-square values for all datasets are shown in the tables and graphs. To be able compare “fuzzy functions with LSE” with the proposed model, “fuzzy functions with GP”, R-square values are also calculated for whole datasetsand used for the comparison and also these R-square values are taken as a basis when the results are depicted in graphs.

As it can be seen in the Figure 6.17, for all chosen optimum cluster numbers, using both membership degrees and membership degrees and their transformations for fuzzy functions increased R-square values. Also for abalone data it could be said that, using membership degrees and their transformations as additional variables increased the R-square values more than using only membership degrees as additional variables.

Table 6.17 R-square values for abalone dataset

		R^2 results for fuzzy functions with LSE for **abalone** dataset				R^2 with only LSE
			Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations	
Cluster number	**3**	R^2train	0.89218	0.89360	0.89487	**0.8654**
		R^2val	0.89423	0.87668	0.88425	
		R^2test	0.87040	0.89087	0.90462	
		R^2all	**0.89022**	**0.89178**	**0.89478**	
	4	R^2train	0.89165	0.89340	0.89432	
		R^2val	0.89230	0.88290	0.88948	
		R^2test	0.88108	0.87482	0.89073	
		R^2all	**0.89038**	**0.89078**	**0.89358**	
	5	R^2train	0.89423	0.89237	0.89472	
		R^2val	0.86360	0.89380	0.89832	
		R^2test	0.87530	0.88165	0.88467	
		R^2all	**0.88950**	**0.89143**	**0.89405**	

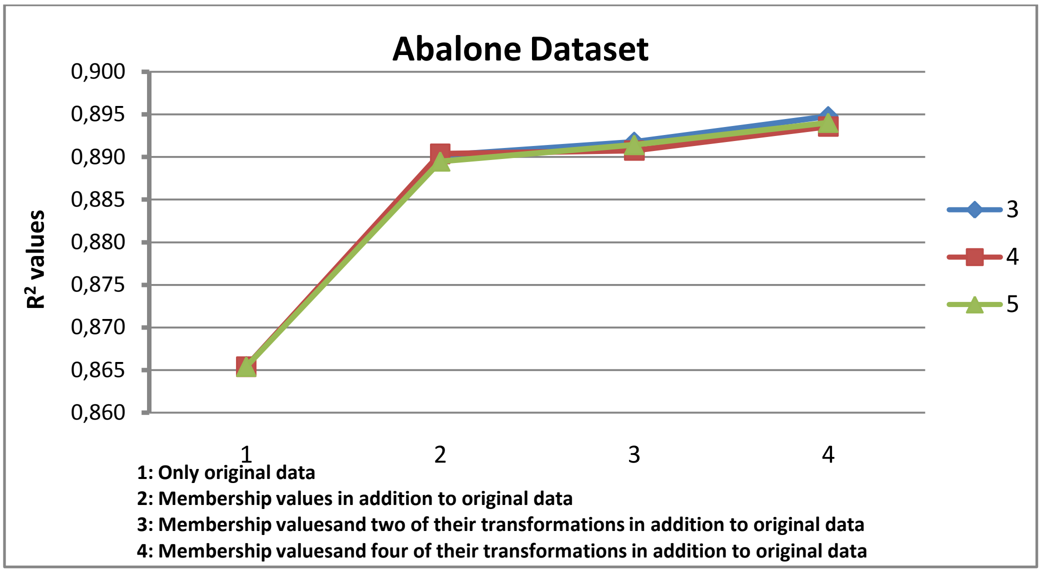


Figure 6.17 Graphical representation of R^2all values for each chosen optimum cluster numberfor abalone dataset

As it can be seen from the Figure 6.18, using fuzzy functions with both membership values and their transformations also increased R-square values for auto-mpg data. According to the graph the same inference could be made that using membership values and their transformations as additional variables improved performance of auto-mpg data more than using only membership degrees.

Table 6.18 R-square values for auto-mpg dataset

		R^2 results for fuzzy functions with LSE for **auto-mpg** dataset				R^2 with only LSE
			Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations	
Cluster number	**5**	R^2train	0.84153	0.85232	0.85923	**0.8151**
		R^2val	0.84210	0.85012	0.85515	
		R^2test	0.81780	0.84865	0.83487	
		R^2all	**0.83942**	**0.85185**	**0.85715**	
	8	R^2train	0.84197	0.86317	0.85702	
		R^2val	0.82353	0.78795	0.85930	
		R^2test	0.82833	0.84503	0.85655	
		R^2all	**0.83973**	**0.85567**	**0.85770**	
	3	R^2train	0.84392	0.84422	0.84293	
		R^2val	0.79965	0.81888	0.84313	
		R^2test	0.83695	0.80263	0.81392	
		R^2all	**0.83932**	**0.83865**	**0.84108**	

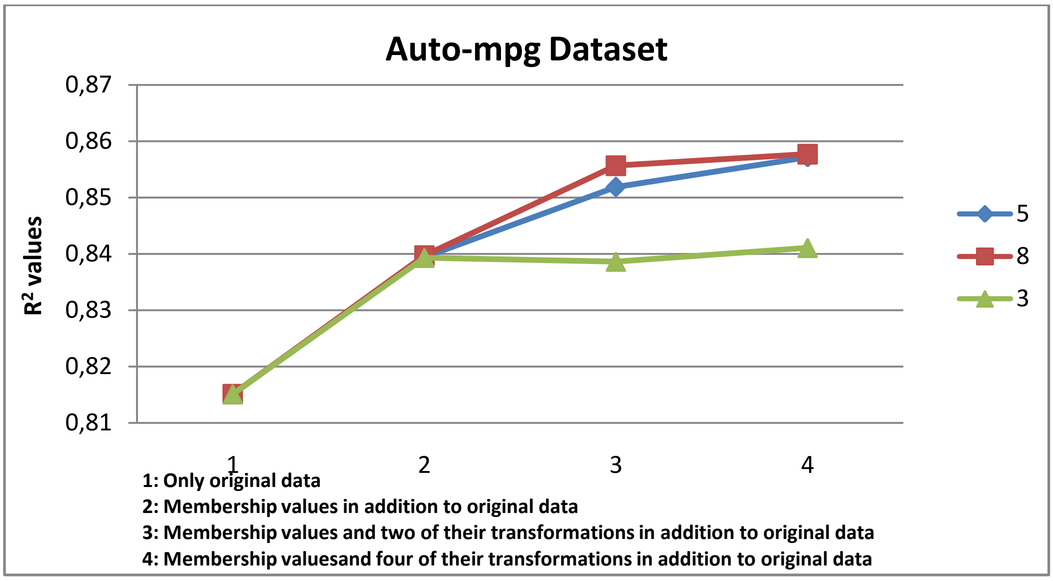


Figure 6.18 Graphical representation of R^2all values for each chosen optimum cluster number for auto-mpg dataset

For concrete dataset, as it could be understood from Figure 6.19, all chosen cluster numbers do not have the same effect. Choosing 9 as cluster number decreased the R-square values at the point of membership degrees and four of their transformations. Choosing 5 also decreased the R-square values a bit for membership degrees and four of their transformations.

Table 6.19 R-square values for concrete dataset

			R^2 results for fuzzy functions with LSE for **concrete** dataset			R^2 with only LSE
			Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations	
Cluster number	**4**	R^2train	0.61822	0.63260	0.63048	**0.6152**
		R^2val	0.62138	0.60662	0.61380	
		R^2test	0.57883	0.57985	0.57960	
		R^2all	**0.61610**	**0.62710**	**0.62543**	
	9	R^2train	0.62193	0.62395	0.56178	
		R^2val	0.55082	0.59883	0.57110	
		R^2test	0.62445	0.61877	0.59720	
		R^2all	**0.61842**	**0.62315**	**0.56758**	
	5	R^2train	0.61762	0.62660	0.61250	
		R^2val	0.60502	0.58298	0.60373	
		R^2test	0.61292	0.63737	0.62107	
		R^2all	**0.61763**	**0.62472**	**0.61372**	

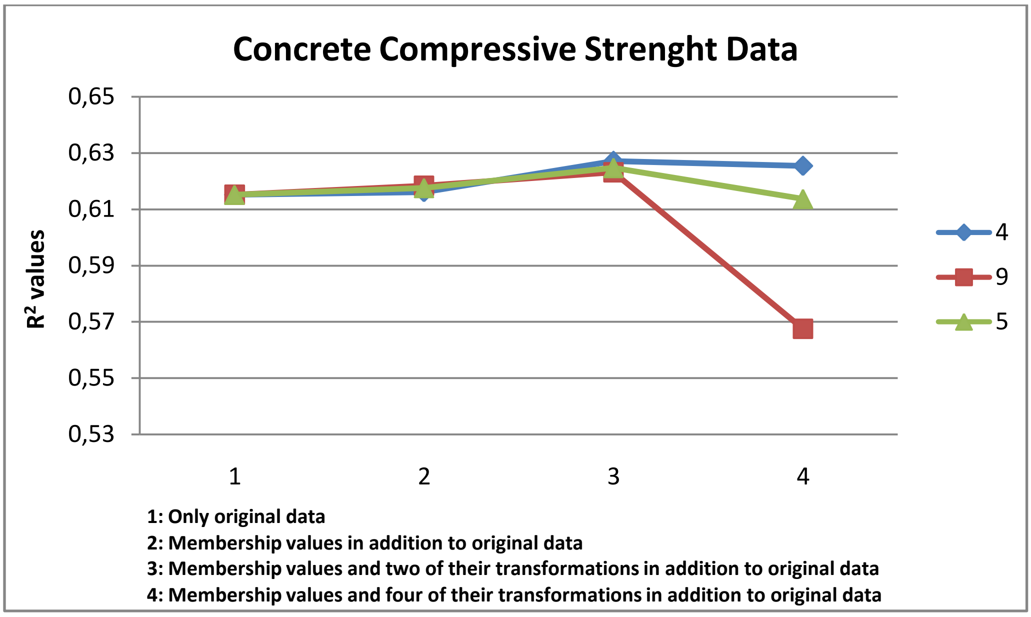


Figure 6.19 Graphical representation of R^{2}all values for each chosen optimum cluster numberfor concrete compressive strength dataset

If we look at the Figure 6.20, we can say that using fuzzy functions with membership values or with membership values and their transformations improved the prediction performance of ecoli data. Also for ecoli dataset it could be said that using transformations of membership values as additional variables improved the performance of fuzzy functions more than using only membership values.

Table 6.20 R-squarevalues for ecoli dataset

		R^2 results for fuzzy functions with LSE for **ecoli** dataset				R^2 with only LSE
			Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations	
Cluster number	**4**	R^{2}train	0.73783	0.75105	0.77197	**0.7375**
		R^{2}val	0.79458	0.77997	0.61908	
		R^{2}test	0.77173	0.73332	0.73327	
		R^{2}all	**0.74832**	**0.75515**	**0.75718**	
	6	R^{2}train	0.74608	0.75268	0.75540	
		R^{2}val	0.72118	0.74867	0.72295	
		R^{2}test	0.74518	0.72452	0.73763	
		R^{2}all	**0.74653**	**0.75367**	**0.75302**	
	5	R^{2}train	0.76092	0.76735	0.78037	
		R^{2}val	0.64585	0.68670	0.74078	
		R^{2}test	0.68947	0.70170	0.62760	
		R^{2}all	**0.74377**	**0.75463**	**0.76320**	

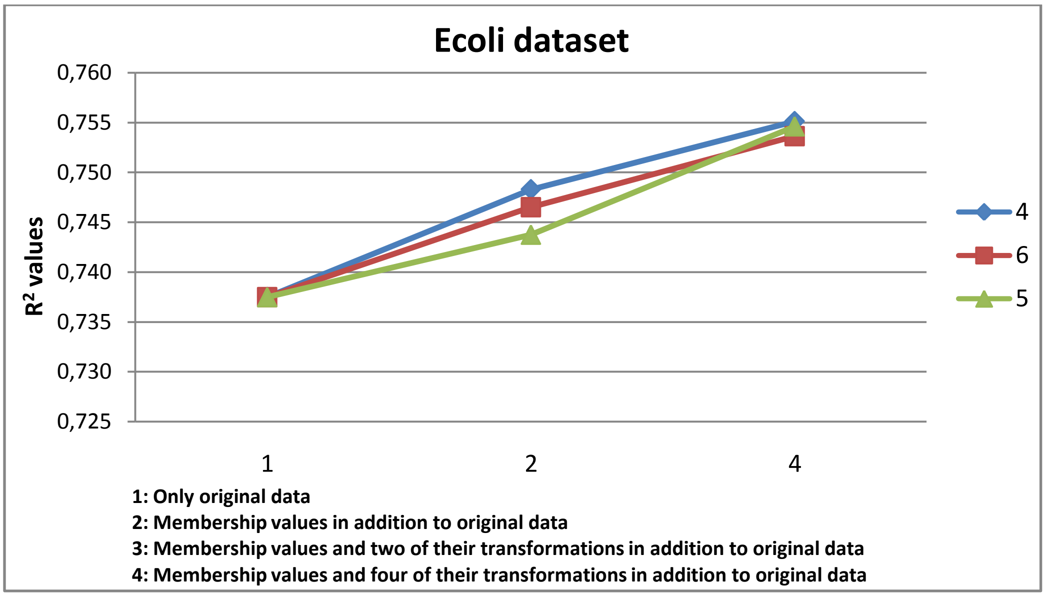


Figure 6.20 Graphical representation of R^2all values for each chosen optimum cluster number for ecoli dataset

As it can be seen from the Figure 6.21, fuzzy function has a significant effect on glass dataset and has improved the prediction performance of the regression analysis prominently.

Table 6.21 R-square values for glass dataset

		R^2 results for fuzzy functions with LSE for **glass** dataset				R^2 with only LSE
			Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations	
Cluster number	**7**	R^2train	0.89997	0.90117	0.92035	**0.6536**
		R^2val	0.82140	0.86152	0.85000	
		R^2test	0.84510	0.85323	0.85417	
		R^2all	**0.89443**	**0.89627**	**0.91170**	
	9	R^2train	0.90122	0.89590	0.91395	
		R^2val	0.79433	0.75117	0.77890	
		R^2test	0.87735	0.77858	0.78478	
		R^2all	**0.89520**	**0.87650**	**0.90033**	
	8	R^2train	0.89640	0.89928	0.90275	
		R^2val	0.90930	0.85073	0.84353	
		R^2test	0.75593	0.85462	0.86823	
		R^2all	**0.89608**	**0.89755**	**0.90277**	

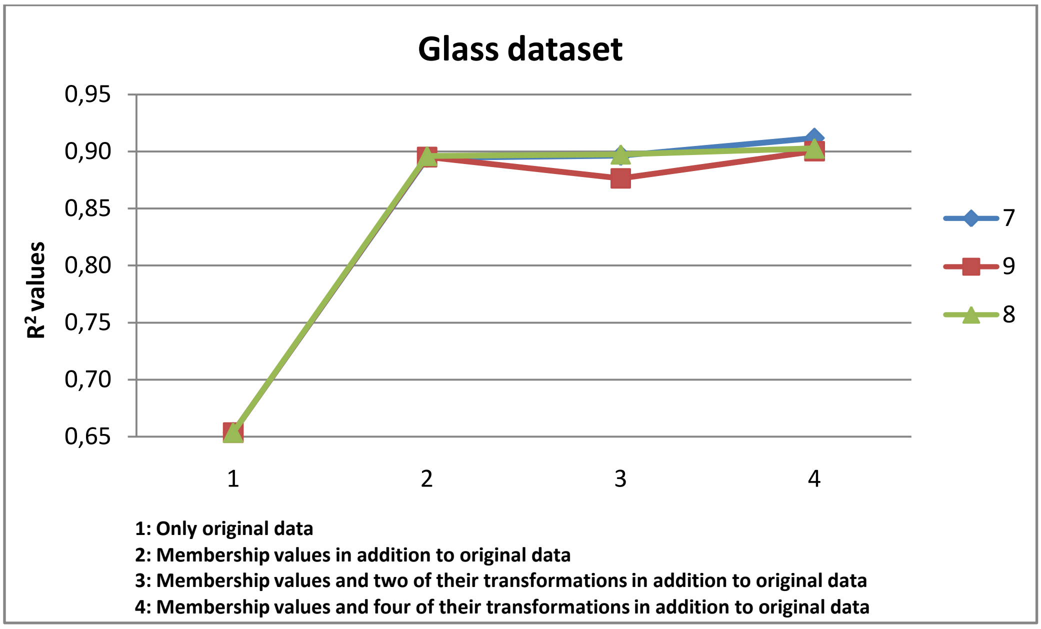


Figure 6.21 Graphical representation of R^2all values for each chosen optimum cluster number for glass dataset

Using fuzzy functions with LSE also affect the prediction performance of housing dataset positively. Using membership values and their transformations provide a regular and explicit increase for R-square values.

Table 6.22 R-square values for housing dataset

		R^2 results for fuzzy functions with LSE for **housing** dataset				R^2 with only LSE
			Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations	
Cluster number	**8**	R^2train	0.74312	0.76535	0.76685	**0.7137**
		R^2val	0.72773	0.70912	0.67987	
		R^2test	0.72040	0.64442	0.75967	
		R^2all	**0.74058**	**0.74762**	**0.75862**	
	3	R^2train	0.75198	0.74362	0.75460	
		R^2val	0.72867	0.76232	0.76780	
		R^2test	0.69350	0.72288	0.67927	
		R^2all	**0.74605**	**0.74627**	**0.74957**	
	6	R^2train	0.75258	0.74667	0.75945	
		R^2val	0.64932	0.71772	0.73062	
		R^2test	0.70790	0.74778	0.69272	
		R^2all	**0.74132**	**0.74612**	**0.75065**	

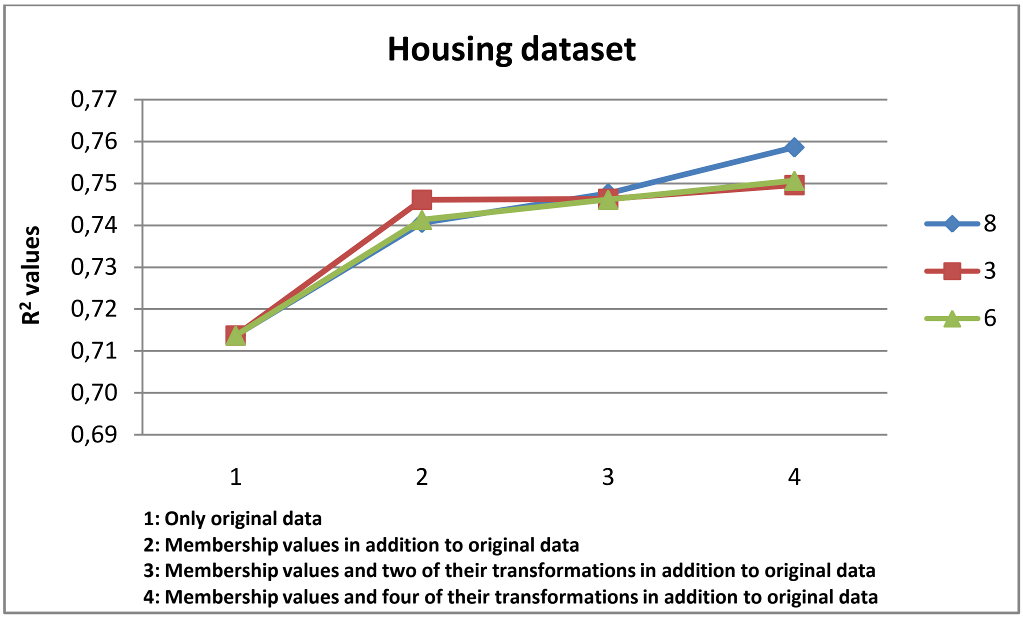


Figure 6.22 Graphical representation of R^2all values for each chosen optimum cluster numberfor housing dataset

As it can be seen in Table 6.23 and Figure 6.23, except membership values and four of their transformations at cluster number 4, fuzzy functions increased the performance of iris dataset regularly.

Table 6.23 R-square values for iris dataset

		R^2 results for fuzzy functions with LSE for **iris** dataset				R^2 with only LSE
			Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations	
Cluster number	**3**	R^2train	0.94430	0.94490	0.94825	**0.9371**
		R^2val	0.94292	0.94885	0.95967	
		R^2test	0.91537	0.92492	0.92307	
		R^2all	**0.94307**	**0.94585**	**0.94792**	
	2	R^2train	0.94230	0.94160	0.94378	
		R^2val	0.91800	0.94495	0.93973	
		R^2test	0.92210	0.93280	0.93768	
		R^2all	**0.93852**	**0.94283**	**0.94342**	
	4	R^2train	0.93985	0.94545	0.94047	
		R^2val	0.92333	0.94113	0.95327	
		R^2test	0.92748	0.91678	0.91950	
		R^2all	**0.93890**	**0.94355**	**0.94030**	

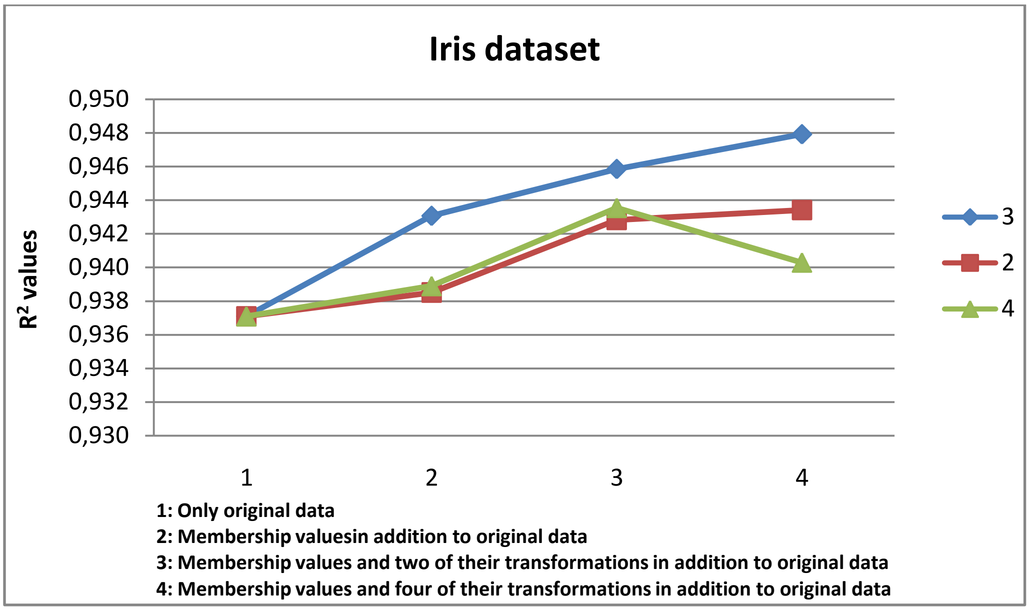


Figure 6.23 Graphical representation of R^2all values for each chosen optimum cluster number for iris dataset

For wine dataset, using fuzzy functions do not provide a regular increase and R-square value is decreasing at cluster number 3 for membership values and two of their transformations. But except this point, R-square values show an increasing trend for the other points and it would not be wrong to say that fuzzy functions give better results compared to regression analysis.

Table 6.24 R-square values for wine dataset

			R^2 results for fuzzy functions with LSE for **wine** dataset			R^2 with only LSE
			Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations	
Cluster number	**3**	R^2train	0.74892	0.75640	0.75915	0.7322
		R^2val	0.69997	0.64325	0.69455	
		R^2test	0.60047	0.68263	0.69972	
		R^2all	**0.73697**	**0.74020**	**0.74900**	
	4	R^2train	0.74602	0.78005	0.76657	
		R^2val	0.68418	0.52653	0.66292	
		R^2test	0.67888	0.61378	0.61602	
		R^2all	**0.73843**	**0.74262**	**0.74843**	
	6	R^2train	0.75920	0.75017	0.77253	
		R^2val	0.69683	0.71373	0.69605	
		R^2test	0.56965	0.68883	0.60248	
		R^2all	**0.73605**	**0.74452**	**0.75168**	

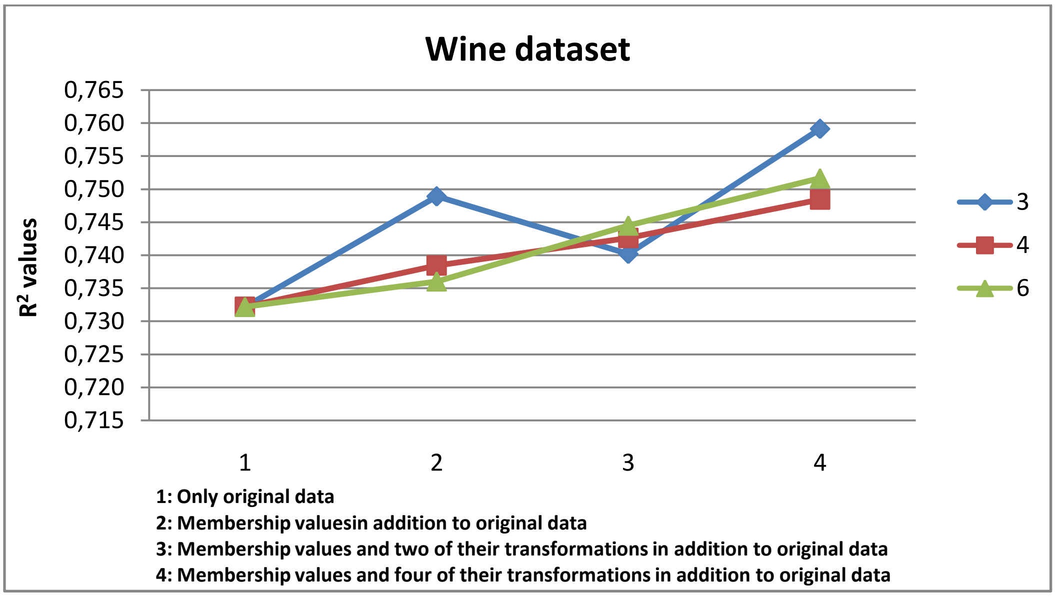


Figure 6.24 Graphical representation of R^2all values for each chosen optimum cluster number for wine dataset

Making a general interpretation, as it could be seen in the tables and graphs, using fuzzy functions have generally improved the predictions performance of regression analysis with a few exceptions. Also it would not be wrong to say that using transformations of membership values in addition to membership values have also improved the performance of fuzzy functions more than using only membership values.

6.5 Application of Fuzzy Functions with GP

Genetic programming on its own is an efficient and powerful method for data analysis. From this point of view it is expected that using genetic programming with fuzzy functions will increase the prediction performance of fuzzy functions.

In this section, the proposed algorithm, fuzzy functions with GP, is applied to the same datasets and R-square values are calculated for selected number of clusters for each dataset.

To interpret Table 6.25 and Figure 6.25, the effect of fuzzy functions with GP is not same for all clusters and has not provide a regular increase for abalone dataset. While at some points it is led to the decrease of R-square values, at some points it is led to increase of R-square values.

Table 6.25 R-square values of genetic fuzzy functions for abalone dataset

		R^2 results of fuzzy functions with genetic programming for **abalone** dataset			
		Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
Chosen optimum cluster number	3	0.9171	0.9180	0.9109	0.9194
	4		0.9091	0.9275	0.9194
	5		0.9143	0.9172	0.9131

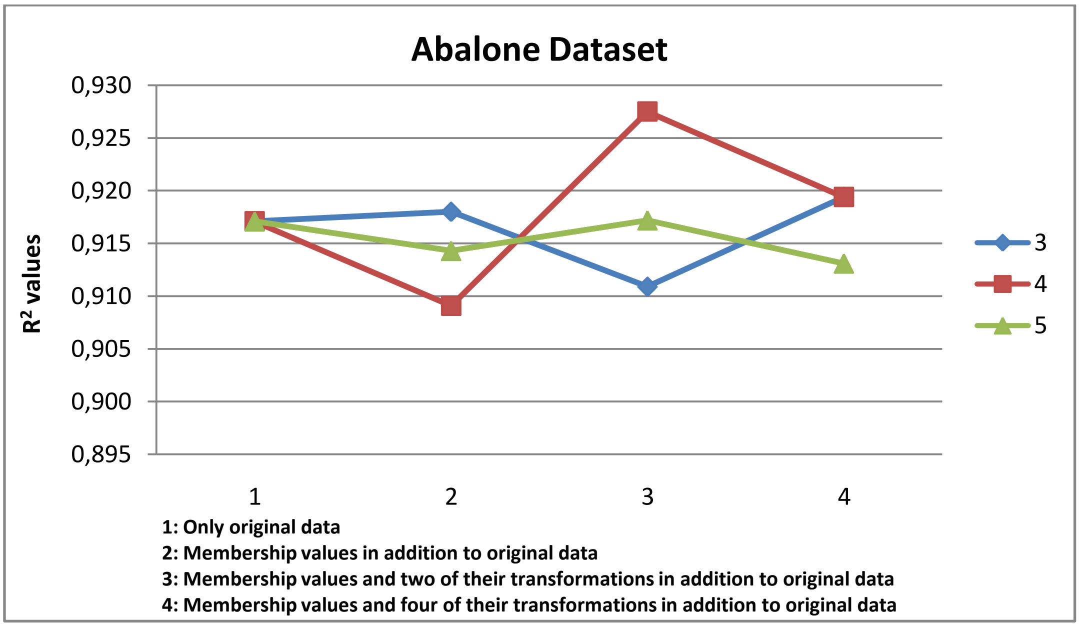


Figure 6.25 Graphical representation of R^2 values for fuzzy functionsgeneticwith programming forabalone dataset

As it can be seen in Table 6.26 and Figure 6.26, using fuzzy functions has increased the R-square values except at cluster number 5 for membership values and four of their transformations.

Table 6.26 R-square values of genetic fuzzy functions for auto-mpg dataset

		R^2 results of fuzzy functions with genetic programming for **auto-mpg** dataset			
		Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
Chosen optimum cluster number	5	0.7623	0.8589	0.8432	0.7185
	8		0.8564	0.8034	0.8630
	3		0.8489	0.8392	0.8559

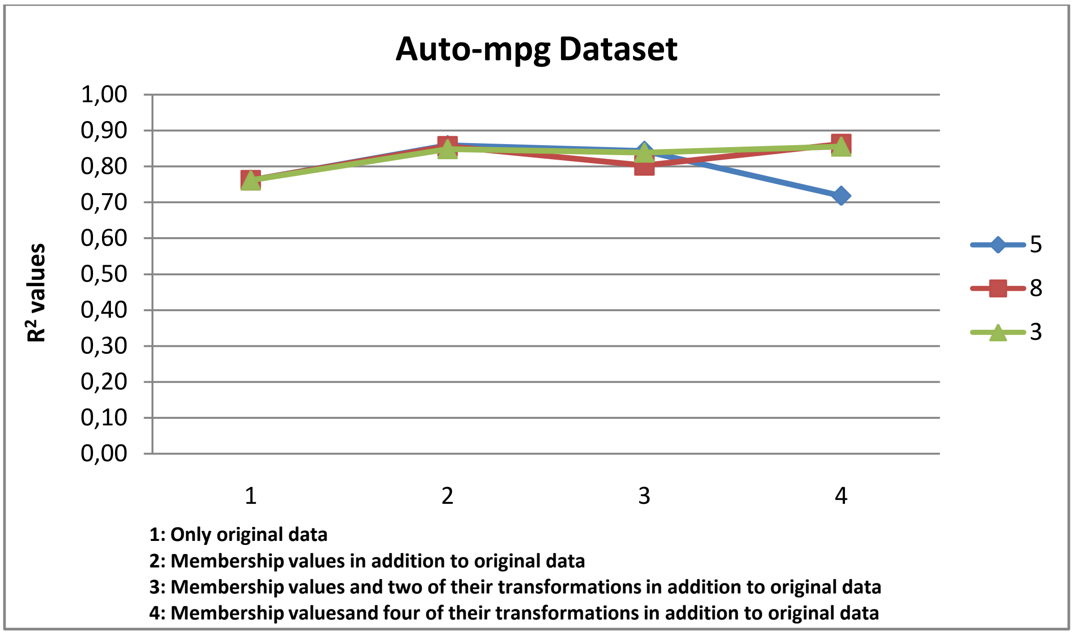


Figure 6.26 Graphical representation of R^2 values for fuzzy functionswith genetic programming for auto-mpg dataset

For concrete compressive strength dataset, there is not a regular increase as it can be seen clearly from Figure 6.27. Although at cluster number 5, R-squarevalues shows a substantial increase, at cluster number 4 and 9, the results of R-squarevalues are inconstant.

Table 6.27 R-square values of genetic fuzzy functions for concrete dataset

		R^2 results of fuzzy functions with genetic programming for **concrete** dataset			
		Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
Chosen optimum cluster number	4	0.7988	0.7703	0.7060	0.7826
	9		0.8068	0.7223	0.7135
	5		0.7839	0.8175	0.8051

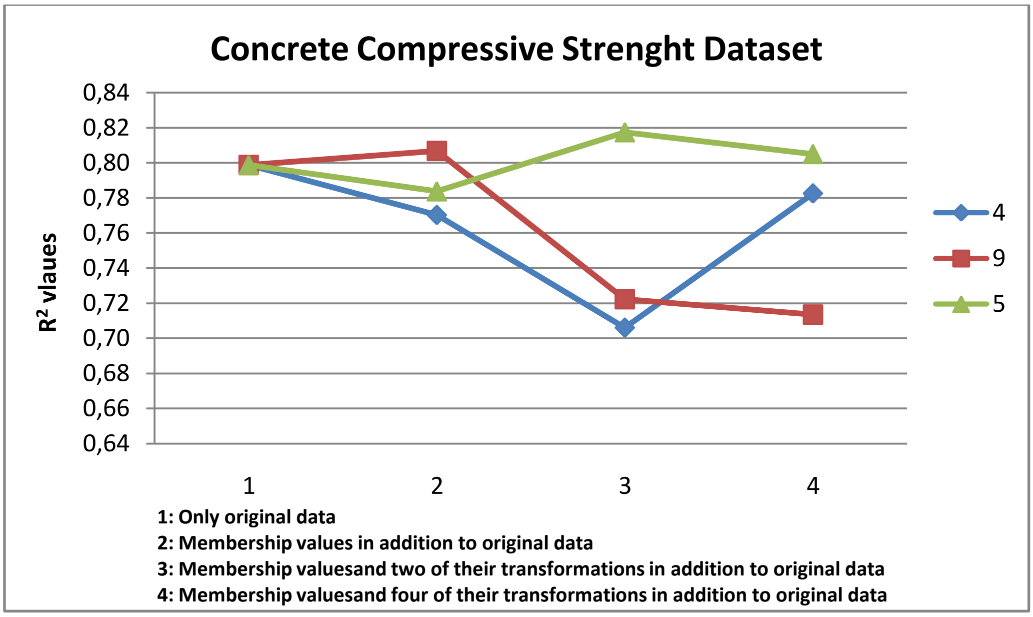


Figure 6.27 Graphical representation of R^2 values for fuzzy functions with genetic programming for concrete compressive strength dataset

There is not also a regular increase for ecoli dataset as it can be seen in Table 6.28. While at some points R-square values shows an increase, generally there is a decrease for R-squarevalues.

Table 6.28 R-square values of genetic fuzzy functions for ecoli dataset

		R^2 results of fuzzy functions with genetic programming for **ecoli** dataset			
		Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
Chosen optimum cluster number	4	0.7855	0.7967	0.7503	0.7603
	6		0.7784	0.7823	0.7723
	5		0.7790	0.7720	0.7831

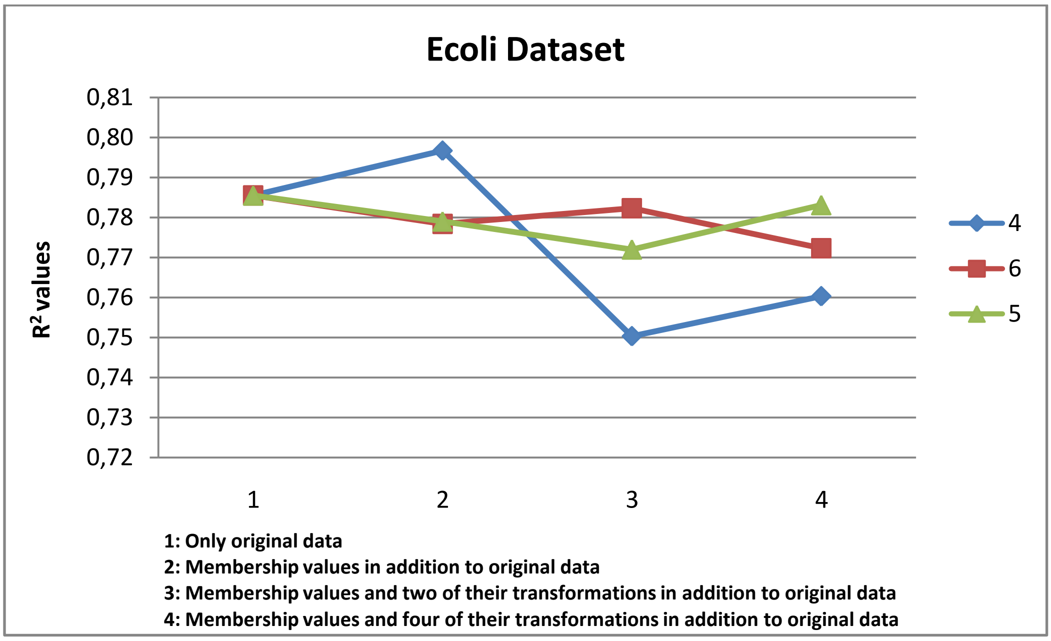


Figure 6.28 Graphical representation of R^2 values for fuzzy functions with genetic programming for ecoli dataset

As it could be seen in Figure 6.29, for glass dataset, using fuzzy functions has improved the predictions performance of genetic programming for all chosen cluster numbers despite the some declines at some points.

Table 6.29 R-squarevalues of genetic fuzzy functions for glass dataset

		R^2 results of fuzzy functions with genetic programming for **glass** dataset			
		Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
Chosen optimum cluster number	7	0.7552	0.8490	0.8659	0.8625
	9		0.7982	0.8726	0.8337
	8		0.8614	0.8461	0.8332

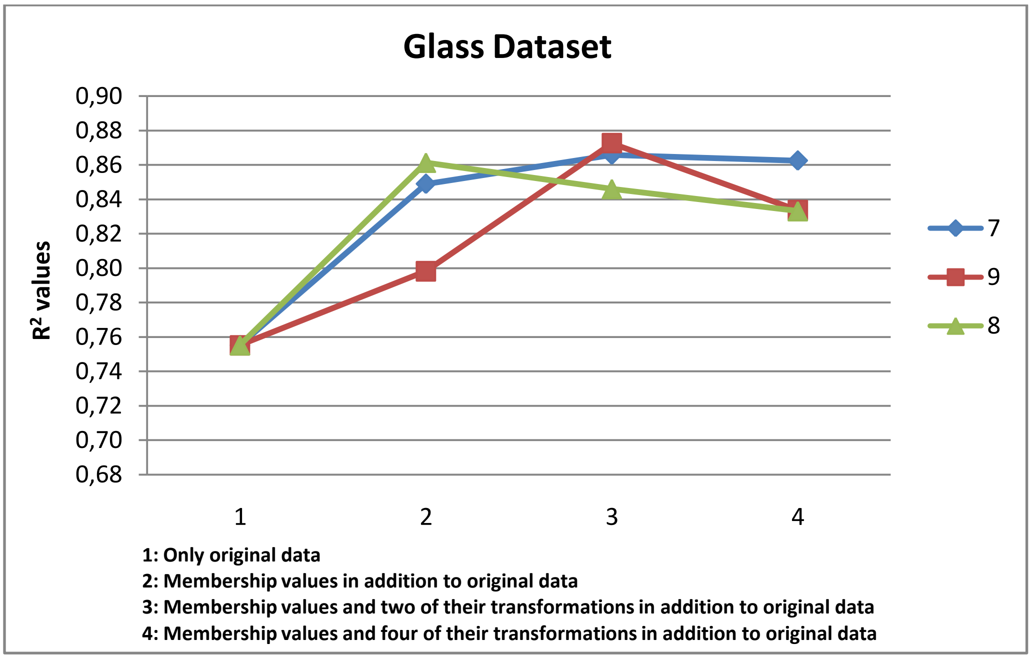


Figure 6.29 Graphical representation of R^2 values for fuzzy functionsgenetic programming for glass dataset

As it could be seen in Figure 6.30, for housing dataset, using fuzzy functions concept has generally improved the predictions performance of genetic programming except the point at which membership values and two of their transformations are used as additional variables.

Table 6.30 R-squarevalues of genetic fuzzy functions for housing dataset

		R^2 results of fuzzy functions with genetic programming for **housing** dataset			
		Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
Chosen optimum cluster number	8	0.7319	0.7342	0.6829	0.7924
	3		0.7631	0.7060	0.7777
	6		0.7488	0.7014	0.7850

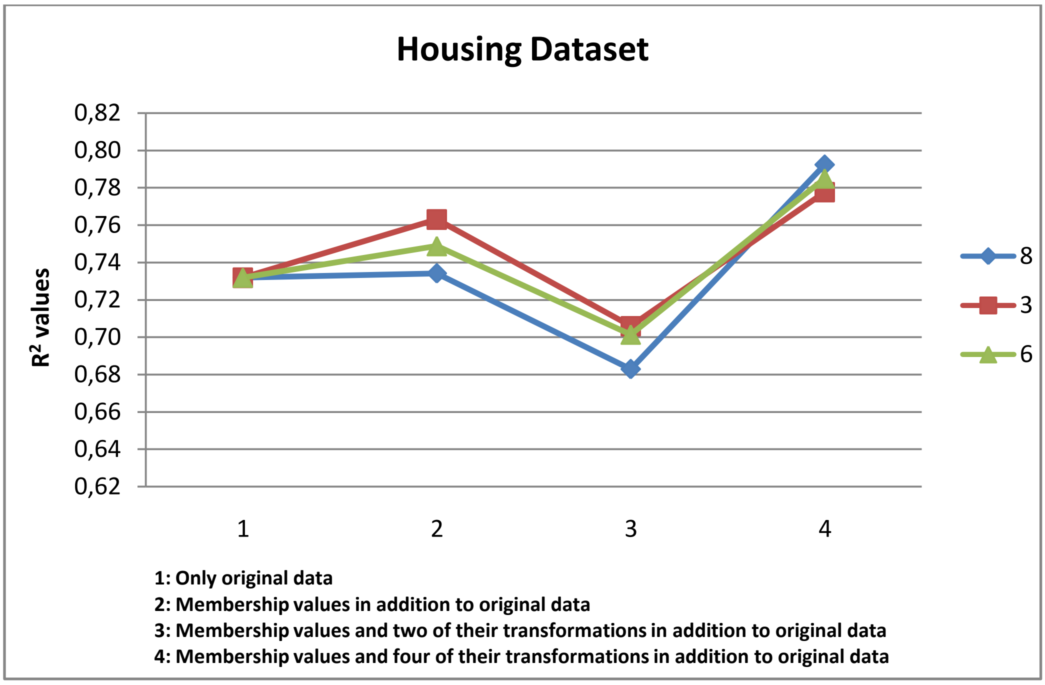


Figure 6.30 Graphical representation of R^2 values for fuzzy functions with genetic programmingfor housing dataset

For iris dataset, all cluster numbers has not shown a positive effect, as it could be seen in Table 6.31 and in Figure 6.31. While R-squarevalues are increasing atcluster number 3, it is decreasing atcluster number 2 and 4.

Table 6.31 R-square values of genetic fuzzy functions for iris dataset

		R^2 results of fuzzy functions with genetic programming for **iris** dataset			
		Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
Chosen optimum cluster number	3	0.9427	0.9576	0.9545	0.9482
	2		0.9418	0.9427	0.9436
	4		0.9390	0.9400	0.9413

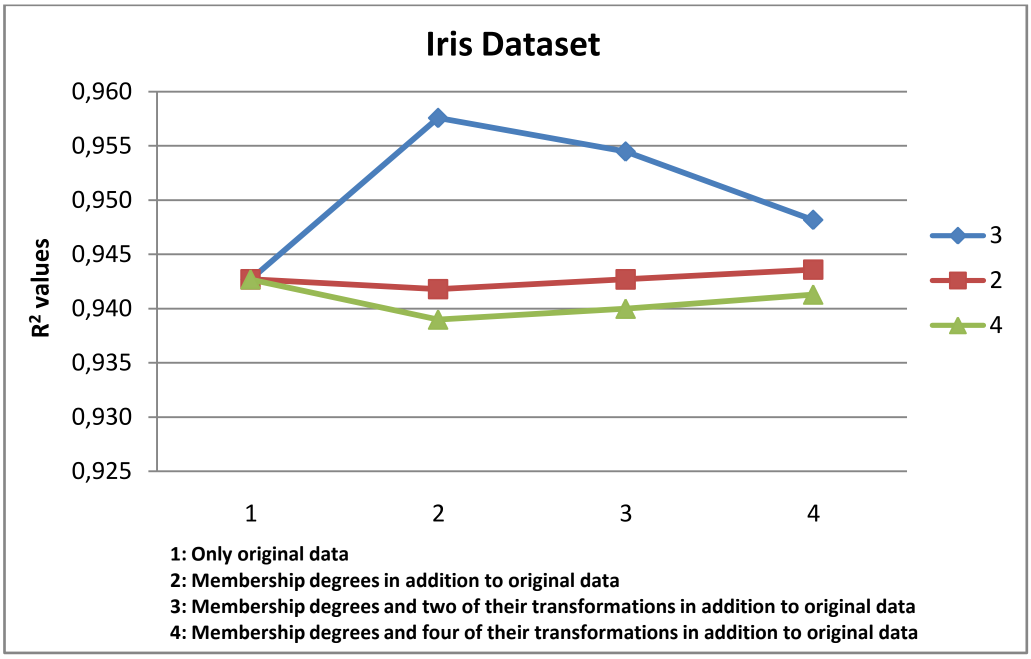


Figure 6.31 Graphical representation of R^2 values for fuzzy functions with genetic programming for iris dataset

As it could be seen in Table 6.32 and Figure 6.32, there is not a regular increase for wine dataset; while at some points, using fuzzy functions improved the R-square values, at some points R-square values are decreasing.

Table 6.32 R-square values of genetic fuzzy functions for wine dataset

		R^2 results of fuzzy functions with genetic programming for **wine** dataset			
		Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
Chosen optimum cluster number	3	0.7362	0.7076	0.7667	0.7736
	4		0.7223	0.781	0.7125
	6		0.7162	0.6963	0.7327

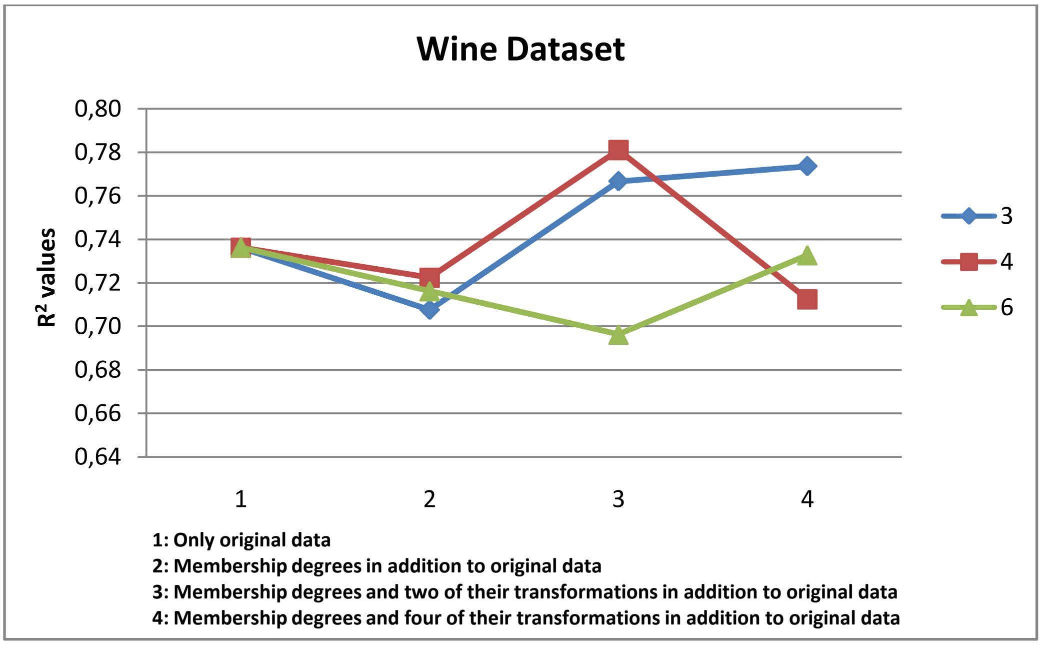


Figure 6.32 Graphical representation of R^2 values for fuzzy functions with genetic programming for wine dataset

If we interpret the all result, it could be said that, using membership degrees and their transformations generally improved the performance of genetic programming as in regression analysis. In the following section, the results of R-square values of fuzzy functions with LSE and fuzzy functions with GP are depicted in a table for all datasets in order to be able to be compared.

Table 6.33 Comparison of fuzzy functions with LSE and fuzzy functions with GP for abalone dataset

Abalone data	Cluster number	Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
R^2 results for fuzzy functions with LSE	3	0.8654	0.89022	0.89178	0.89478
	4		0.89038	0.89078	0.89358
	5		0.88950	0.89143	0.89405
R^2 results for fuzzy functions with GP	3	0.9171	0.9180	0.9109	0.9194
	4		0.9091	0.9275	0.9194
	5		0.9143	0.9172	0.9131

Table 6.34 Comparison of fuzzy functions with LSE and fuzzy functions with GP for auto-mpg dataset

Auto-mpg data	Cluster number	Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
R^2 results for fuzzy functions with LSE	5	0.8151	0.83942	0.85185	0.85715
	8		0.83973	0.85567	0.85770
	3		0.83932	0.83865	0.84108
R^2 results for fuzzy functions with GP	5	0.7623	0.8589	0.8432	0.7185
	8		0.8564	0.8034	0.8630
	3		0.8489	0.8392	0.8559

Table 6.35 Comparison of fuzzy functions with LSE and fuzzy functions with GP for concrete dataset

Concrete data	Cluster number	Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
R^2 results for fuzzy functions with LSE	4	0.6152	0.61610	0.62710	0.62543
	9		0.61842	0.62315	0.56758
	5		0.61763	0.62472	0.61372
R^2 results for fuzzy functions with GP	4	0.7988	0.7703	0.7060	0.7826
	9		0.8068	0.7223	0.7135
	5		0.7839	0.8175	0.8051

Table 6.36 Comparison of fuzzy functions with LSE and fuzzy functions with GP for ecoli dataset

Ecoli data	Cluster number	Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
R^2 results for fuzzy functions with LSE	4	0.7375	0.74832	0.75515	0.75718
	6		0.74653	0.75367	0.75302
	5		0.74377	0.75463	0.76320
R^2 results for fuzzy functions with GP	4	0.7855	0.7967	0.7503	0.7603
	6		0.7784	0.7823	0.7723
	5		0.7790	0.7720	0.7831

Table 6.37 Comparison of fuzzy functions with LSE and fuzzy functions with GP for glass dataset

Glass data	Cluster number	Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
R^2 results for fuzzy functions with LSE	7	0.6536	0.89443	0.89627	0.91170
	9		0.89520	0.87650	0.90033
	8		0.89608	0.89755	0.90277
R^2 results for fuzzy functions with GP	7	0.7552	0.8490	0.8659	0.8625
	9		0.7982	0.8726	0.8337
	8		0.8614	0.8461	0.8332

Table 6.38 Comparison of fuzzy functions with LSE and fuzzy functions with GP for housing dataset

Housing data	Cluster number	Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
R^2 results for fuzzy functions with LSE	8	0.7137	0.74058	0.74762	0.75862
	3		0.74605	0.74627	0.74957
	6		0.74132	0.74612	0.75065
R^2 results for fuzzy functions with GP	8	0.7319	0.7342	0.6829	0.7924
	3		0.7631	0.7060	0.7777
	6		0.7488	0.7014	0.7850

Table 6.39 Comparison of fuzzy functions with LSE and fuzzy functions with GP for iris dataset

Iris data	Cluster number	Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
R^2 results for fuzzy functions with LSE	3	0.9371	0.94307	0.94585	0.94792
	2		0.93852	0.94283	0.94342
	4		0.93890	0.94355	0.94030
R^2 results for fuzzy functions with GP	3	0.9427	0.9576	0.9545	0.9482
	2		0.9418	0.9427	0.9436
	4		0.9390	0.9400	0.9413

Table 6.40 Comparison of fuzzy functions with LSE and fuzzy functions with GP for wine dataset

Wine data	Cluster number	Only original data	Membership degrees	Membership degrees and two of their transformations	Membership degrees and four of their transformations
R^2 results for fuzzy functions with LSE	3	0.7322	0.73697	0.74020	0.74900
	4		0.73843	0.74262	0.74843
	6		0.73605	0.74452	0.75168
R^2 results for fuzzy functions with GP	3	0.7362	0.7076	0.7667	0.7736
	4		0.7223	0.7810	0.7125
	6		0.7162	0.6963	0.7327

6.6 Conclusion

In this part of the study, fuzzy functions with LSE and fuzzy functions with GPare applied to the datasetsand the effect of fuzzy functions concept on genetic programming is tried to be searched. According to the obtained results it could be said that fuzzy functions with LSE improved the prediction performance and gave better results with a few exceptions compared to regression analysis. When fuzzy functions are generated using genetic programming also improved the prediction performance in some cases.

In the following chapter, a briefsummary of the study is made and then a general assessment is made on fuzzy functions approach by comparing and evaluating the obtained results. Finally the study is terminated with future research part.

CHAPTER SEVEN
CONCLUSION AND FUTURE RESEARCH

7.1 Conclusion

In this part of the study, the purpose of the study is going to be overviewed and a general summary of the thesis is going to be made. Then finally future works are going to be represented.

As it was expressed before in previous chapters, the prime purpose of this study is to represent fuzzy functions with GP on the basis of fuzzy functions approach and its foundations. For this purpose a general review of the related topics which constitute the basis of fuzzy functions approach and form the starting point of fuzzy functions are represented.Firstly in chapter 2, fuzzy rule bases approach which is one of the most commonly known fuzzy inference system and applied in a variety of fieldsis introduced. The foundations of fuzzy rule bases, commonly used types of fuzzy rule bases and its main disadvantages are overviewedbriefly in chapter 2.

Due to play a crucial part in fuzzy functions conceptthe concept of fuzzy clustering, important types of clustering algorithms and commonly used clustering validity indexes are explained briefly in chapter 3.

The fundamental theory of fuzzy functions approach, which is proposed by Türkşen in order to eliminate the difficulties of fuzzy rule bases and constitutes the basis of this study is introduced in chapter 4. Then in the following sections the algorithm of fuzzy functions with LSE is represented and explained with a numerical example step by step.

In chapter 5, the proposed approach in which it is recommended to use genetic programming in generating fuzzy functions is introduced. For this purpose firstly the theory of genetic programming is overviewed andafterwards the proposed algorithm is introduced and then explained with a numerical example step by step.

In chapter 6, fuzzy functions with LSE and proposed model, fuzzy functions with GP, applied to datasets from the literature in order to be able to compare and present the performance of our approach.

If we evaluate the resultsit could be said that generating fuzzy functions by using different analyzing methods generally give better results and improve the prediction performance. But we can say that the effects of fuzzy functions are changing depending on the dataset. In the present thesis while using fuzzy functions approach improved the performance of some datasets significantly, for some of the datasets it showed just a small improvement and even decreased the prediction performance.

7.2 Future Works

Suggestions for future works based on the obtained results could be stated as follows;

- As it stated in previous section, the effects of type of problems (regression, classification, regression and classification i.e.) could be searched in detail by applying fuzzy functions to different types of problems.
- By applying different clustering validity indexes,more appropriate cluster numbers could be found out and thus the effects of more appropriate number of clusters could be compared.
- By using different clustering algorithms, the effect of the clustering algorithms could be searched in detail.
- For the present study Eureqa Formulize software program is used for the proposed model. As a future research, by choosing different types parameters and even using different genetic programming software the effects of them could be searched in details.

REFERENCES

Al-Rahamneh, Z., Reyalat, M., Sheta, A. F., Bani-Ahmad, S., & Al-Oqeili, S. (2011). A new software reliability growth model: Genetic programming based approach. *Journal of Software Engineering and Applications, 4,* 476-481.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Perez, J. M., & Perona I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46,* 243-256.

Balasko, B., Abonyi, J., & Feil, B. (2005). *Fuzzy clustering and data analysis toolbox for use with Matlab.* Retrieved September 14, 2012, from http://www.mathworks.com/matlabcentral/fileexchange/7473-clustering-and-data-analysis-toolbox

Başkır, M. B., & Türksen, I. B. (2013). Enhanced fuzzy clustering algorithm and cluster validity index for human perception. *Expert Systems with Applications, 40,* 929-937.

Baykasoğlu, A., Göçken, M., & Özbakır, L. (2010).Genetic programming based data mining approach to dispatching rule selection in a simulated job shop. *Simulation, 86*, 715-728.

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

Bezdek, J. C, & Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences, 10,* 191-203.

Brameier, M. F., & Banzhaf, W. (2007). *Linear Genetic Programming.*USA: Springer.

Butt, F., & Abhari, A. (2010). *Genetic algorithms: an approach to optimal web cache replacement.* In: Proceedings of the 2010 Spring Simulation Multiconference, Florida, USA.

Cannon, R. L., Dave, J. V., & Bezdek, J. C. (1986). Efficient implementation of the fuzzy c-means clustering algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 8*, 248-255.

Chaira, T. (2012). Intuitionistic fuzzy color clustering of human cell images on different color models. *Journal of Intelligent & Fuzzy Systems 23,* 43-51.

Chan, K.Y., Kwong, C. K., & Wong, T. C. (2011). Modeling customer satisfaction for product development using genetic programming. *Journal of Engineering Design, 22*(1), 55-68.

Chang, Y.-C. & Chen, S.-M. (2009). *Temperature prediction based on fuzzy clustering and fuzzy rules interpolation techniques*. In: Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan, 3444-3449.

Cominetti, O., Matzavinos, A., Samarasinghe, S., Kulasiri, D., Liu, S., Maini, P. K., & Erban, R. (2010). DifFUZZY: A fuzzy clustering algorithm for complex data sets. *International Journal of Computational Intelligence in Bioinformatics and Systems Biology, 1,* 402-417.

Cordon, O. (2011). A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. *International Journal of Approximate Reasoning 52,* 894–913.

Cordon, O., Herrera, F., Hoffmann, F., & Magdalena, L. (2001). *Genetic fuzzy systems evolutionary tuning and learning of fuzzy knowledge bases.* Singapore: World Scientific.

Cordon, O., Herrera F., & Zwir, I. (2001). Fuzzy modeling by hierarchically built fuzzy rule bases. *International Journal of Approximate Reasoning, 27,* 61-93.

Çelikyılmaz, F. A. (2005). *Fuzzy Functions with support vector machines.*University of Toronto. Department of Mechanical and Industrial Engineering. Msc Thesis, University of Toronto, Toronto, Canada.

Çelikyılmaz, A., & Türkşen, İ. B. (2007a). *Evolutionary fuzzy system models with improved fuzzy functions and its application to industrial process.* In: IEEE International Conference on Systems, Man and Cybernetics, Montreal, Canada, 541-546.

Çelikyılmaz, A., & Türkşen, İ. B. (2007b). Fuzzy functions with support vector machines. *Information Sciences, 17,* 5163-5177.

Çelikyılmaz, A., & Türkşen, İ. B. (2008a). Enhanced fuzzy system models with improved fuzzy clustering algorithm. *IEEE Transaction on Fuzzy Systems, 16,* 779-794.

Çelikyılmaz, A., & Türkşen, İ. B. (2008b). Uncertainty modeling of improved fuzzy functions with evolutionary systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 38,* 1098-1110.

Çelikyılmaz, A., & Türkşen, İ. B. (2009a). A genetic fuzzy system based on improved fuzzy functions. *Journal of Computers, 4*(2), 135-146.

Çelikyılmaz, A., & Türkşen, İ. B. (2009b). *Modeling uncertainty with fuzzy logic with recent theory and applications.* Berlin: Springer.

Çunkaş, M., & Taşkıran, U. (2011). Turkey's Electricity Consumption Forecasting Using Genetic Programming. *Energy Sources, Part B: Economics, Planning, and Policy, 6,* 406-416.

Dagher, I. (2012). Complex fuzzy c-means algorithm. *Artificial Intelligence Review, 38*(1), 25-39.

Demirci, M. (1999). Fuzzy functions and their fundamental properties. *Fuzzy Sets and Systems, 106*(2), 239-246.

Demirci, M. (2000). Fuzzy functions and their applications. *Journal of Mathematical Analysis and Applications, 252*, 495-517.

Demirci, M. (2001). Gradation of being fuzzy function. *Fuzzy Sets and Systems, 119*(3), 383-392.

Dulyakarn, P., & Rangsansei, Y. (2001). *Fuzzy c-means clustering using spatial information with application to remote sensing*. In: 22nd Asian Conference on Remote Sensing, Singapore, Singapore.

Fyfe, C., Marney, J. P., & Tarbert, H. F. E. (1999). Technical analysis versus market efficiency - a genetic programming approach. *Applied Financial Economics, 9*(2), 183-191.

Grupe, F. H., & Jooste, S. (2004). Genetic algorithms: A business perspective. *Information Management & Computer Security, 12*(3), 288-297.

Hamed, M., Keshavarz, A., Dehghani, H., & Pourghassem, H. (2012). *A clustering technique for remote sensing images using combination of watershed algorithm and Gustafson-Kessel clustering.*In: Fourth International Conference on Computational Intelligence and Communication Networks, Mathura, India, 222-226.

Hathaway, R. J., & Bezdek, J. C. (2006). Extending fuzzy and probabilistic clustering to very large data sets. *Computational Statistics & Data Analysis, 51*, 215-234.

Hewai, N. M. (2012). Genetic programming and genetic algorithms for propositions. *Journal of Applied Computer Science & Mathematics, 13*(6), 32-37.

Javadi, A. A., Farmani, R., & Tan, T.P. (2005). A hybrid intelligent genetic algorithm. *Advanced Engineering Informatics 19,* 255–262

Kamyab, S., & Bahrololoum, A. (2012). Designing of rule base for a TSK- fuzzy system using bacterial foraging optimization algorithm (BFOA). *Procedia - Social and Behavioral Sciences, 32,* 176 – 183.

Karthick, R., Saravanan, S., & Vetrisalvan, M. (2012). Optimization technique for maximization problem in evolutionary programming of genetic algorithm in data mining. *International Journal on Computer Science and Engineering (IJCSE), 4,* 1562-1569.

Kaur, A.,& Kaur, A. (2012). Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System. *International Journal of Soft Computing and Engineering (IJSCE), 2*(2), 323-325.

Kim, H.-S., Kim, J.-H., Ho, C.-H., & Chu, P.-S. (2011). Pattern classification of typhoon tracks using the fuzzy c-means clustering method. *Journal of Climate, 24,* 488-508.

Koza, R. J. (1995). *Survey of genetic algorithms and genetic programming*. In: Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies,San Francisco, USA, 589-594.

Kuo, C.-F. J., Jian, B.-L., Wu, H.-C., & Peng, K.-C. (2012). Automatic machine embroidery image color analysis system. Part I: Using Gustafson-Kessel clustering algorithm in embroidery fabric color separation. *Textile Research Journal, 82,* 571–583.

Kuo, C.-F. J., Shih, C.-Y., & Lee, J.-Y. (2004). Automatic recognition of fabric weave patterns by a fuzzy c-means clustering method. *Textile Research Journal, 74*(2), 107-111.

Langdon, W. B., Poli, R. McPhee, N. F., & Koza, J. R. (2008). Genetic programming: An introduction and tutorial, with a survey of techniques and applications. *Studies in Computational Intelligence (SCI), 115,* 927-1028.

Leondes, C. T. (Ed.). (1998). *Fuzzy logic and expert systems applications*. United States of America: Academic Press.

Mamdani, E. H. (1974). Applications of fuzzy algorithms for simple control of dynamic plant. *Proceedings of the Institution of Electrical Engineers, 121*, 1585-1588.

Matworks: Evaluating goodness of fit (n.d.). Retrieved February, 10, 2013, from http://www.mathworks.com/help/curvefit/evaluating-goodness-of-fit.html

Maulik, U., & Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern Recognition, 33,* 1455-1465.

Moallem, P., Mousevi, B. S., & Monadjemi, S. A. (2011). A novel fuzzy rule base system for pose independent faces detection. *Applied Soft Computing, 11*(2), 1801-1810.

Moreno-Torres, J. G., Llorà, X., Goldberg, D. E., & Bhargava, R. (2013). Repairing fractures between data using genetic programming-based feature extraction: A case study in cancer diagnosis. *Information Sciences, 222,* 805-823.

Niittymaki, J. (2001).General fuzzy rule base for isolated traffic signal control-rule formulation. *Transportation Planning and Technology, 24*(3), 227-247.

Nutonian: About Eureqa (n.d.). Retrieved July, 8, 2012, from http://www.nutonian.com/eureqa/

Palit, K. A., & Popovic, D. (2005). *Computational intelligence in time series forecasting: Theory and engineering applications*. London: Springer-Verlag.

Parker, J. K., Hall, L. O., & Bezdek, J. C. (2012). *Comparison of scalable fuzzy clustering methods.* In: 2012 IEEE International Conference on Fuzzy Systems, Brisbane, Australia, 1-9.

Ponce-Cruz, P., & Ramirez-Figueroa, F. D. (2010). *Intelligent control systems with LabVIEW^{TM}*. London: Springer-Verlag.

Rezaee, M. R., Lelieveldt, B. P. F., & Reiber, J. H. C. (1998). A new cluster validity index for the fuzzy c-mean. *Pattern Recognition Letters, 19*, 237-246.

Ross, J. T. (2004). *Fuzzy logic with engineering applications* (2nd ed.). England: John Wiley & Sons Ltd.

Sastry, K., Goldberg, D., & Kendall, G. (2005). Genetic algorithms. In E. K., Burke, & G. Kendall, (Ed.). *Search Methodologies:Introductory tutorials in optimization and decision support techniques* (97-125). USA: Springer.

Sato, T., Suzuki, T., & Mabuchi, K. (2007). *Fast automatic template matching for spike sorting based on Davies-Bouldin validation indices.* In: Proceedings of the 29th Annual International Conference of the IEEE EMBS, Lyon, France, 3200-3203.

Siarry, P., & Guely, F. (1998). A genetic algorithm for optimizing Takagi-Sugeno fuzzy rule bases. *Fuzzy Sets and Systems, 99*, 37-47.

Sivarao, Brevern, P., El-Tayeb, N. S. M., & Vengkatesh V. C. (2009). GUI based Mamdani fuzzy inference system modeling to predict surface roughness in laser machining. *International Journal of Electrical & Computer Sciences IJECS, 9*(9), 281-288.

Song, A., & Zhang, M. (2012). Genetic programming for detecting target motions. *Connection Science, 24*(2-3), 117-141.

Stavrakoudis, D. G., Galidaki, G. N., Gitas, I. Z., & Theocharis J. B. (2012). A genetic fuzzy-rule-based classifier for land cover classification from hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing, 50*(1), 130-148.

Surmann, H. & Selenschtschikow, A. (2002). *Automatic generation of fuzzy logic rule bases: Examples I.* In: Proc. of the NF2002: First International ICSC Conference on Neuro-Fuzzy Technologies, Havana, Cuba, 75-81.

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics, 15*(1), 116-132.

Tsoi, H.-P., & Gao, F. (1999). Control of injection velocity using a fuzzy logic rule-based controller for thermoplastics injection molding. *Polymer Engineering and Science, 39*(1), 3-17.

Türkşen, İ. B. (2011). A review of developments in fuzzy system models: Fuzzy rule bases to fuzzy functions. *Scientia Iranica, Transactions D: Electrical and Computer Engineering, 18*, 522-527.

Türkşen, İ. B. (2012). A review of developments from fuzzy rule bases to fuzzy functions. *Hacettepe Journal of Mathematics and Statistics, 41*, 347-359.

Türkşen, İ. B., & Çelikyılmaz, A. (2006). Comparison of fuzzy functions with fuzzy rule base approaches. *International Journal of Fuzzy Systems, 8*(3), 137-149.

Uci Learning Machine Repository: Datasets (n.d.). Retrieved May, 15, 2012, from http://archive.ics.uci.edu/ml/datasets.html

Wikipedia: Linear least squares (mathematics) (2010). Retrieved February, 10, 2013, from, http://en.wikipedia.org/wiki/Linear_least_squares_%28mathematics%29

Yang, M.-S., Hwang, P.-Y., & Chen, D.-H. (2004). Fuzzy clustering algorithms for mixed feature variables. *Fuzzy Sets and Systems, 141*, 301–317.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8,* 338-353.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics, 3*(1), 28-44.

Zhang, F., & Qian, X. (2012). A new validity index for fuzzy clustering. *Journal of Computational Information Systems, 8*, 5875–5883.

Zhou, J., Yu, L., Mabu, S., Shimada, K., Hirasawa, K., & Markon, S. (2008). A traffic-flow-adaptive controller of double-deck elevator systems using genetic network programming. *Transactions on Electrical and Electronic Engineering, 3,* 703-714.

APPENDIX

Appx.1

Least Square Estimation

In statistics and mathematics, linear least squares is an approach fitting a mathematical or statistical model to data in cases where the idealized value provided by the model for any data point is expressed linearly in terms of the unknown parameters of the model (Wikipedia, 2010).

In a regression model, the assumption is that the dependent variable is a linear function of one or more independent variables plus an error factor. Let the regression model be defined as a multi-input, single output (MISO) model as follows:In matrix notationthe general linear model is expressed as (Çelikyılmaz and Türkşen, 2009b, p. 340);

$$y = \beta_0 + \beta_1 x_1 + \beta_{nv} x_{nv} + \varepsilon \qquad (A.1)$$

Here;

- ‘y’ represents the output variable,
- $x_{nv's}$ are the input variables where nv is the number of variables,
- ε represent the error term.
- β_j’s are the coefficients parameters.

To represent the regression model in matrix notation;

$$y = X\beta + \varepsilon \qquad (A.2)$$

- y is output matrix that consist of n vectors,

- X is the inputs which is consist of $[n \times p]$ matrix of. Here n represents the number of vectors, nv is the number of variables.
- β is represent the coefficient parameters matrix that is consist of $[nv \times 1]$
- ε represents the error matrix which is consist of $[n \times 1]$.

The objective is to minimize the total residuals. Therefore the simplest linear regression, which tries to minimize the total squared error between the actual and estimated output, is called the least squares regression. (Çelikyılmaz and Türkşen, 2009b, p. 341);

$$minQ = \sum_{k=1}^{n} (y_i - \beta_0 + \beta_1 x_{1,k} + \beta_{nv} x_{nv,k})^2 \qquad (A.3)$$

In matrix notation the equation below is minimized;

$$\begin{aligned} &minQ = (y - X\beta)'(y - X\beta) \\ &\frac{\partial}{\partial\beta}[(y - X\beta)'(y - X\beta)] = 0 \qquad (A.4) \\ &2(X^{'}X)\beta = 2X'y \\ &\beta = (X^{'}X)^{-1}X'y \end{aligned}$$

Appx.2

Calculation of R-square value

Sum of squares due to error: This statistic measures the total deviation of the response values from the fit to the response values and also called as the summed square of residuals and represent as below (MathWorks, n.d.).

$$SSE = \sum_{i=1}^{n} w_i(y_i - \hat{y}_i)^2 \qquad (A.5)$$

Here;

- $y_i$represents the observed output value,
- $\hat{y}_i$represents the predicted output value.
- w_i is the weighting value and generally takes 1.

This statistic measures how successful the fit is in explaining the variation of the data. In other words R-square is the square of the correlation between the response values and the predicted response values. (MathWorks, n.d.).

R-square is defined as the ratio of the sum of squares of the regression (SSR) and the total sum of squares (SST). SSR is defined as;

$$SSR = \sum_{i=1}^{n} w_i(\hat{y}_i - \bar{y})^2 \qquad (A.6)$$

$$SST = \sum_{i=1}^{n} w_i(y_i - \bar{y})^2 \qquad (A.7)$$

Where $\bar{y}$ is the mean of the observed data y_i.

$$Rsquare = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} \qquad (A.8)$$