d`f"k esa Qt+h ,ao d`f=e ra=dh; usVodZ dk;Zfof/k;ksa dk v/;;u

Study of Fuzzy and Artificial Neural Network Methodologies in Agriculture

RAMA KRISHNA SINGH

INDIAN AGRICULTURAL STATISTICS RESEARCH INSTITUTE
INDIAN AGRICULTURAL RESEARCH INSTITUTE
NEW DELHI – 110 012

2007

Study of Fuzzy and Artificial Neural Network Methodologies in Agriculture

By

RAMA KRISHNA SINGH

A Thesis Submitted to the Faculty of the Post-Graduate School,
Indian Agricultural Research Institute, New Delhi,
in partial fulfillment of the requirements
for the award of the degree of

DOCTOR OF PHILOSOPHY

IN

AGRICULTURAL STATISTICS

Approved by:

Chairperson:

(Dr. Prajneshu)

Members:

(Dr. Anil Rai)

(Dr. D. R. Singh)

(Dr. D. K. Singh)

(Mr. H. S. Sikarwar)

INDIAN AGRICULTURAL STATISTICS RESEARCH INSTITUTE

LIBRARY AVENUE, PUSA, NEW DELHI-12

Dr. Prajneshu
Principal Scientist
Head Division of Biometrics
I.A.S.R.I. New Delhi-110012

CERTIFICATE

This is to certify that the work incorporated in the thesis entitled **Study of fuzzy and artificial neural network methodologies in agriculture** submitted in partial fulfillment of the requirements for the award of degree of **Doctor of Philosophy in Agricultural Statistics** of the Post-Graduate School, Indian Agricultural Research Institute, New Delhi, is a record of *bonafide* research carried out by **Mr. Rama Krishna Singh** under my guidance and supervision and no part of this dissertation has been submitted for any other degree or diploma.

All assistance and help received during the course of this investigation has been duly acknowledged.

New Delhi (**PRAJNESHU**)

Dated: Chairman Advisory Committee

ACKNOWLEDGEMENTS

I feel extremely happy while expressing my profound sense of respect, gratitude and thanks to Dr. Prajneshu, Principal Scientist, Head-Division of Biometrics, Indian Agricultural Statistics Research Institute (I.A.S.R.I.), New Delhi and Chairman of my Advisory Committee for his inspiring guidance, innovative ideas, constant encouragement and constructive criticism during the period of this investigation and also in the preparation of this thesis.

I am equally indebted to Dr. Himadri Ghosh, Scientist (SS), I.A.S.R.I., New Delhi for his constant help, counsel, valuable suggestions and punctilious criticism to help for accomplishment of my research work.

I accolade my respect to Dr. V. C. Sud, Principal Scientist, Division of Sample Survey, I.A.S.R.I., New Delhi for providing me relevant data for my research work.

I am extremely grateful to Dr. Anil Rai, Senior Scientist, Division of Computer Applications, IASRI, New Delhi and and Co-chairman of my Advisory Committee for his valuable guidance and advices during this investigation.

I am thankful to Mr. Sanjeev Kumar, Scientist, Division of Environmental Science, IASRI, New Delhi and Mr. Harnam Singh Sikarwar, Scientist, Division Computer Applications, IASRI, New Delhi and member of my advisory committee for their helpful attitude and encouragement throughout the course of this study.

I am highly grateful to Dr. S. D. Sharma, Director, IASRI for providing all facilities and constant encouragement throughout the study.

I avail this opportunity to express my profound gratitude to Dr. V. K. Sharma, Professor (Agril. Statistics), IASRI, New Delhi for providing the necessary research facilities my research work. I also appreciate the help rendered by the staff of TAC and CAS Lab, I.A.S.R.I.

It gives me great pleasure to express my heartfelt thanks to all my respected seniors, beloved juniors and batch mates for their constant help and enthusiasm, which helped me to accomplish this venture.

I extend my thanks to the Dean and Director, IARI, New Delhi and staff of PG school for their helpful attitude and cooperation, throughout the period of study. Last but not least I am thankful to the financial assistance that I got from I.A.S.R.I. in the form of SRF during the tenure of my study.

Diction is not enough to express unbountiful love, gratitude and regards to my parents and my elder sisters for their dedicated efforts. My affectionate thanks are always due to them for their love, affection, support and constant ushering of blessings on me. I want to express my sincere thanks to my younger brother & my wife. They were the inspiration and moral boosting who gave me sufficient energy to complete this thesis.

Above all, I bow my head to my Gurudev and Almighty who blessed me to retain good health and peace of mind.

Dated:
Place: IASRI, New Delhi *(Rama Krishna Singh)*

CONTENTS

CHAPTER I

INTRODUCTION

---

1. Various Categories of Mathematical Models

A mathematical model is an equation or set(s) of equations that represents the behavior of a system. With the availability of powerful computer systems and variety of software, modelling of complex and dynamic systems is now possible. Various categories of mathematical models are:

1.1 Empirical and mechanistic models

In an empirical model, the parameters do not have any biological interpretation where as in a mechanistic model these have specific biological interpretation and provide insight into the underlying biological phenomenon. Polynomial and regression models are examples of empirical models while logistic and Gompertz models are examples of mechanistic models.

1.2 Static and dynamic models

In a model, if time variable is not present, then it is called a static model. Such a formulation implicitly assumes that the quantified relationships remain the same across the design space of the data. The passage of time always brings changed circumstances, new situations, and fresh considerations, which can be handled by a dynamic model. So, a model is said to be dynamic if it contains time explicitly as a variable. Dynamical models are generally expressed in terms of differential or difference equations.

1.3 Deterministic and statistical models

A model is said to be deterministic if definite prediction of quantities (e.g. size of organism, rainfall, yield of crop) is made without any associated probability distribution. Examples of some of the deterministic models are:

$$Y(t) = A + B * t \tag{1.1}$$

$$Y(t) = A + B * t + C * t^2 \tag{1.2}$$

$$Y(t) = A * exp(-B * t) \tag{1.3}$$

Models given by above equations have been posed deterministically, as if the data never deviated from the model under consideration. It is obvious that this is unrealistic. Some statistical assumptions are needed to make the model workable. By adding an error term on the right hand side of the above equations one arrives at a statistical model. It is to be noted that the error term carries some appropriate assumptions, like that of independence and homoscedasticity and the distribution being normal.

1.4 Linear and nonlinear models

Linear Model: A linear model is one in which all the parameters appear linearly. Some examples of linear model are:

(a) Multiple linear regression

$$Y = a_0 + a_1 X_1 + ... + a_p X_p + \varepsilon ,$$

where Y is the dependent (predictand or response) variable, X_i are independent (predictor or stimulus) variables and ε is the error term.

(b) Polynomial models with one predictor variable

$Y = a + bX + \varepsilon$ (First-order model)

$Y = a + bX + cX^2 + \varepsilon$ (Second-order or Quadratic or Curvilinear model)

Above models are very widely used in Agriculture, Industry, Education, Medicine, etc. 'Method of least squares' is generally employed for estimation of parameters. However, if polynomial models of a certain order are fitted to data by applying this procedure and later it is decided to add an extra term of a higher order, then estimates of all the parameters in the model have to be computed afresh.

Nonlinear Models: It is well recognized that any type of statistical inquiry in which principles from some body of knowledge enter seriously into the analysis is likely to lead to a 'Nonlinear model'. Such models play a very important role in understanding the complex inter-relationships among variables. A 'nonlinear model' is one in which at least one of the parameters appears nonlinearly. More formally, in a 'nonlinear model', at least one derivative with respect to a parameter should involve that parameter. Examples of a nonlinear model are:

$$Y(t) = exp\left(at + bt^2\right) \tag{1.4}$$

$$Y(t) = at + exp(-bt) \tag{1.5}$$

Note: Some authors use the term ‘intrinsically linear’ to indicate a nonlinear model which can be transformed to a linear model by means of some transformation. For example, the model given by eq. (1.4) is ‘intrinsically linear’ in view of the transformation $X(t) = log_e Y(t)$.

2. Basics of Fuzzy Set Theory

Lotfi A. Zadeh introduced a theory whose objects- fuzzy sets-are sets with boundaries that are not precise. The membership in a fuzzy set is not a matter of affirmation or denial, but rather a matter of a degree. Zadeh not only challenged probability as sole agent for uncertainty but the very foundations upon which probability theory is based: two valued logic. When A is a fuzzy set and x is a relevant object, the proposition “x is a member of A” is not necessarily either true or false, as required by two-valued logic, but it may be true only to some degree, the degree to which x is actually a member of A. It is most common, but not required to express degrees of membership in fuzzy set as well as degrees of truth of the associated proposition by numbers in the closed unit interval [*0,1*]. The extreme values in this interval *0* and *1* then represent, respectively, the total denial and affirmation of the membership in a given fuzzy set as well as the falsity and truth of the associated proposition. The capability of fuzzy set to express gradual transitions from membership to non-membership and vice versa has a broad utility. It provides us not only with a meaningful and powerful representation of measurement uncertainties, but also with a meaningful representation of vague concepts expressed in natural language.

The fuzzy set A, in certain numerical universe of discourse X, is a set of pairs: $A = \{(\mu_A(x), x)\}, \ \forall x \in X$. where: μ_A is the membership function of the fuzzy set A, which assigns to each element ($x \in X$) the grade of its membership μ_A in the fuzzy set A, considering the fact $\mu_A(x) \in [0, 1]$. The membership function maps the numerical universe X of a given variable in the interval [*0, 1*]:

$$\mu_A(x): X \to [0, 1], \quad \forall x \in X$$

The concept of the fuzzy set enables us to formulate mathematically a notation of linguistic values and fuzzy numbers applied by people. The membership function can be represented in the form of a diagram (continuous or discrete diagram), mathematical formula, table, membership vector, sum or integral.

Normal fuzzy sets have membership function that assumes a value between *0* and *1* (including *1*). The range of the membership function need not be limited to the values between *0* and *1*. It is possible to be within this range when only set-type operations are also carried out on fuzzy sets. If arithmetical operations are also carried out, then values greater than *1* can be obtained as a result of them. If the maximal value of membership in a set will be denoted as $Sup_x \mu_A(x)$, then each non-empty fuzzy set A can be normalized to the set An in the interval of 0 and 1, by dividing the primary membership by its maximal value.

$$\mu_{A_n} = \frac{\mu_A(x)}{Sup_x \mu_A(x)}$$

Subnormal fuzzy sets are the sets whose maximal value of the membership function is less than 1. Subnormal fuzzy sets arise as a result of various operations (like set type, algebraic and arithmetical operations) executed on normal sets.

2.1 Characteristic parameters of a fuzzy set

(a) Height of a fuzzy set

This is the maximal value that the membership function has in the whole universe of discourse X of the set.

H(A)= *height* (A)= $Sup_x(\mu_A(x))$

Since generally the membership function can have many local maxima, the height of a set is defined with the aid of the supremum operation.

(b) Support of a fuzzy set A

This is the crisp subset of the set A all of whose elements have non-zero membership grades in the set A.

S(A)= Supp (A)= $\{x : \mu_A(x) > 0,\ \forall x \in X\}$

(c) Core of a Fuzzy set A

This is the crisp subset A in the universe of discourse X consisting of all elements with a membership grade equal to one.

$$\mathrm{C}(A) = \mathrm{Core}\,(A) = \{x : \mu_A(x) = 1,\ \ x \in X\}$$

(d) Convex and non-convex fuzzy sets

A convex fuzzy set has the property that all its α-cuts are compact (closed), one-part interval of the universe of discourse X . In the case of a non-convex fuzzy set there exists non-compact (unclosed) α-cuts consisting of many parts.

Non-convex sets come into existence as a result of performing set-type, algebraic and arithmetical operations on primary sets, which are convex as a rule. A convex fuzzy set satisfies the following conditions:

$$x_1 \le x_2 \le x_3 \Rightarrow \mu_A(x_2) \ge \min(\mu_A(x_1), \mu_A(x_3)),\ \forall x_1, x_2, x_3 \in X$$

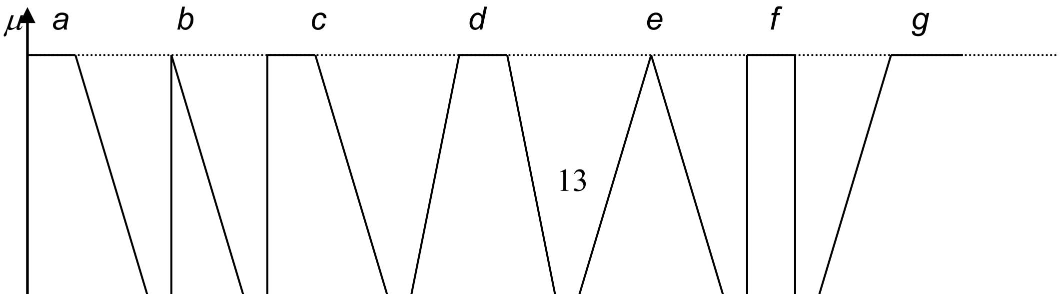
or, $$\mu_A(\lambda x_1 + (1-\lambda)x_3) \ge \min(\mu_A(x_1), \mu_A(x_3)),\ \forall \lambda \in [0,1]\, and\ \forall x_2, x_3 \in X$$

2.2 Types of membership functions of fuzzy sets

In practice of fuzzy set various types of membership functions are applied. Some of them are as follows: (a) Polynomial membership function (b) Gaussian membership function (symmetrical/asymmetrical) (c) Sigmoid membership function.

(a) Polynomial membership function

These functions are very often used in practice for their simplicity. The forms of the most of ten applied functions of the type of polygon are represented below as follows:



a- Left outside membership function; *b*-Triangular asymmetrical membership function; *c*-Trapezoidal asymmetrical membership function; *d* -Trapezoidal symmetrical membership function; *e*- Triangular symmetrical membership function; *f* -Rectangular membership function; *g* –Right outside membership function

In order to write down polygonal membership function mathematically logical variables $w_i : \{0, 1\}$ should be used. As an example in the case of the trapezoidal membership function Figure 1, the following logical variables are introduced:

$$w_1 = \begin{cases} 1 & for \;\; a \le x < b, \\ 0 & otherwise, \end{cases} \quad , \quad w_2 = \begin{cases} 1 & for \;\; b \le x < c, \\ 0 & otherwise, \end{cases} \quad , \quad w_3 = \begin{cases} 1 & for \;\; c \le x < d, \\ 0 & otherwise, \end{cases}$$

the membership function of the type of asymmetrical trapezoidal can be represented in the form:

$$\mu(x) = w_1(x - a/b - a) + w_2 + w_3(d - x/d - c)$$

In the case of the symmetrical triangular function, only one logical variable w must be introduced:

$$w = \begin{cases} 1 & for \;\; (e \text{ - } a) \le x < (e + a), \\ 0 & otherwise, \end{cases}$$

The membership function can be written down in the following form:

$$\mu(x) = w(a - |x - e|/a)$$

Both the case are depicted in figure 1 below.

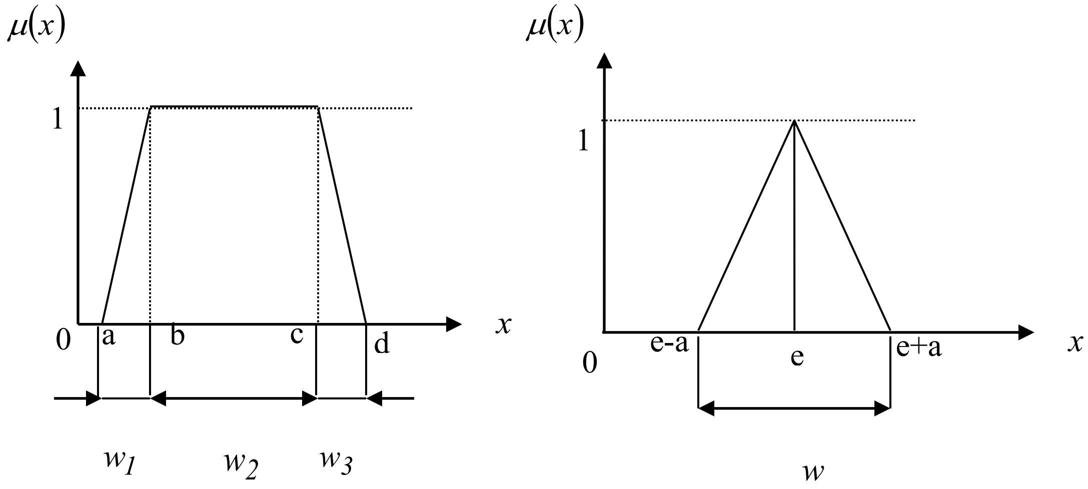


Figure1: Trapezoidal symmetrical and triangular symmetrical membership functions

(b) Gaussian symmetrical function

The Gaussian function is expressed by the formula

$$\mu(x) = exp\left[-(x - b/a)^2\right]$$

The Shape of the Gaussian function is determined by two parameters *a* and *b*, where *b* is the modal value of the function, and the parameter a determines its width.

(c) Sigmoid membership function

The Gauss functions are symmetrical functions and are suitable for representing internal fuzzy sets. For representing external sets the right and left sigmoid function can be used. The right sigmoid function is expressed by the formula:

$$\mu(x) = 1/1 + exp[-a.(x-b)]$$

3. Fuzzy Regression Methodology

Regression analysis is a very powerful technique for studying functional relationships between response and explanatory variables. In conventional regression analysis, deviations between the observed values and the estimates are assumed to be due to random errors. Thus, statistical techniques are applied to perform estimation and inference in regression analysis. However the deviations are sometimes due to the indefiniteness of the structure of the system or imprecise observations. Studies dealing with Fuzzy linear regression (FLR) model with fuzzy parameters can be broadly classified into two approaches, viz. (a) Linear programming (LP)-based methods, (b) Fuzzy least squares (FLS) methods.

3.1 Linear programming (LP)-based method

Tanaka *et al.* proposed this approach in 1982. FLR model along with parameters are expressed as:

$$Y = A_o + A_1.X_{1j} + ... + A_p.X_{pj}$$

where,

Parameters are denoted as $A_i = (a_{ic}, a_{iw})$ $i = 0, 1, ..., p$, here a_{ic}-denotes center, a_{iw}-denotes radius, Response variable as $Y = (y_c, y_w)$ and $X_{1j}, ..., X_{pj}$ are explanatory variables.

Parameters are estimated by minimizing “Total vagueness” of model-data combination i.e. sum of radii of predicted intervals, subject to constraints that each data point must lie within estimated value of response variable. This can be visualized as a LP problem and solved by using “Simplex procedure”.

$$Y_j = A_o + A_1.X_{1j} + ... + A_p.X_{pj}$$

Also,

$$y_j =< a_{oc}, a_{ow} > + < a_{1c}, a_{1w} > .x_{1j} + ... + < a_{pc}, a_{pw} > .x_{pj} =< y_{jc}, y_{jw} >, \text{ say}$$

Thus

$$y_{jc} = a_{oc} + a_{1c}.x_{1j} + ... + a_{pc}.x_{pj},$$

$$y_{jw} = a_{ow} + a_{1w}.\left|x_{1j}\right| + ... + a_{pw}.\left|x_{pj}\right|$$

Now, consider there are m data points, each comprising a (p+1)-row vector

Minimize

$$\sum_{j=1}^{m} \left(a_{ow} + a_{1w}.\left|x_{1j}\right| + ... + a_{pw}.\left|x_{pj}\right| \right)$$

Subject to

$$\sum_{j=1}^{m} \left\{ \left(a_{0c} + \sum_{i=1}^{p} a_{ic} x_{ij} \right) - \left(a_{0w} + \sum_{i=1}^{p} a_{iw} x_{ij} \right) \right\} \le Y_j$$

$$\sum_{j=1}^{m} \left\{ \left(a_{0c} + \sum_{i=1}^{p} a_{ic} x_{ij} \right) + \left(a_{0w} + \sum_{i=1}^{p} a_{iw} x_{ij} \right) \right\} \ge Y_j \text{ and } a_{iw} \ge 0$$

Several software packages, like SAS, LP88, and LINDO are available for solving LP problem. Any standard spreadsheet package, like Microsoft Excel may also be used to solve LP problem manually.

3.2 Fuzzy least squares (FLS) method

P. Diamond proposed the approach of FLS in 1988, which as its name suggests, is a fuzzy extension of Least squares method based on a new defined distance on the space of fuzzy numbers. In this approach, parameters are estimated by minimizing total vagueness, i.e. sum of widths of predicted intervals under the constraints that all data points fall within estimated interval of response variable. Predicted intervals computed using “Method of fuzzy least squares” have much shorter average widths as compared to that obtained using “Method of least squares”. This implies that former procedure is more efficient than latter.

4. Artificial Neural Network (ANN) Methodology

Computers are extremely fast at numerical computations, far exceeding human capabilities. However, the human brain has many abilities that would be desirable in computer. These include: ability to quickly identify features, even in the presence of noise; to understand, interpret, and act on probabilistic or fuzzy notations; to make inferences and judgments based on past experiences and relate them to situation that have never been encountered before; and to suffer localized damage without loosing complete functionality (fault tolerance). So even though the computer is faster than human brain in numeric computations, the brain outperforms the computer in other tasks. This is the underlying motivation for trying to understand and model the human brain using artificial neural network.

Artificial neural networks are parallel computational models comprised of densely interconnected adaptive processing units. These networks are fine-grained parallel implementations of nonlinear static or dynamic systems. A very important feature of these networks is their adaptive nature, where “learning by example” replaces “programming” in solving problems. This feature makes such computational models very appealing in application domains where one has little or incomplete understanding of the problem to be solved but where training data is readily available. In this sense, they can be treated as one of the *multivariate nonlinear nonparametric statistical methods.* This modeling approach with ability to learn from experience is very useful in many practical problems since it is often easier to have data than to have good theoretical guesses about underlying laws governing the systems from which data are generated.

Artificial neural networks are viable computational models for a wide variety of problems. They have been used in classification problems, such as identifying underwater sonar currents, recognizing speech, and predicting heart problems in patients. In time-series applications, ANNs have been used in predicting stock market performance. These are currently preferred tool in predicting protein secondary structures. As statisticians or users of statistics, these problems are normally solved through classical statistical methods, such as discriminant analysis, logistic regression, Bayes analysis, multiple regression, and ARIMA time-series models. It is, therefore, time to recognize ANN as a powerful tool for data analysis. An excellent overview of various aspects of ANN is provided by Warner and Misra (1996) and Cheng and Titterington (1994).

Neural networks have seen an explosion of interest over the last few years, and are being successfully applied across on extraordinary range of problem domains, in areas as diverse as finance, medicine, engineering, geology and physics. Indeed, anywhere that there are problems of prediction, classification or control, neural networks are being introduced. This sweeping success can be attributed to a few key factors:

- Power: Neural networks are very sophisticated modeling techniques capable of modeling extremely complex functions. In particular, neural networks are *nonlinear*. For many years linear modeling has been the commonly used technique in most modeling domains since linear models have well-known optimization strategies. Where the linear approximation was not valid (which was frequently the case) the models suffered accordingly. Neural networks also keep in check the *curse of dimensionality* problem that bedevils attempts to model nonlinear functions with large numbers of variables.
- Ease of use: Neural networks *learn by example*. The neural network user gathers representative data, and then invokes *training algorithms* to

automatically learn the structure of the data. Although the user does need to have some heuristic knowledge of how to select and prepare data, how to select an appropriate neural network, and how to interpret the results, the level of user knowledge needed to successfully apply neural networks is much lower than would be the case using (for example) some more traditional nonlinear statistical methods.

4.1 Basics of ANN

The artificial neuron is the basic building block/processing unit of an artificial neural network. These processing units are often referred as nodes, and loosely represent the biological neuron. It is necessary to understand the computational capabilities of this processing unit as a prerequisite for understanding the function of a network of such units.

To capture the essence of biological neural systems, an artificial neuron is defined as follows:

- It receives a number of inputs (either from original data, or from the output of other neurons in the neural network). Each input comes via a connection that has a strength (or *weight*); these weights correspond to synaptic efficacy in a biological neuron. Each neuron also has a single threshold value. The weighted sum of the inputs is formed, and the threshold subtracted, to compose the *activation* of the neuron (also known as the post-synaptic potential, or PSP, of the neuron).

- The activation signal is passed through an activation function (also known as a transfer function) to produce the output of the neuron.

(a) Model of Neuron

The McCulloch-Pitts model (McCulloch-Pitts 1943) of the neuron was one of the first attempts in this area. A neuron collects signals (weighted sum of input) at its synapses from connected units by summing all the excitatory and inhibitory influences acting on it. If the excitatory influences are dominant, then the neuron fires and sends this message to other neurons via the outgoing synapses. In this sense, the neuron function can be modeled as a simple threshold function f(.). As shown in figure 2., in both diagrams, the neuron fires if the combined signal strength exceeds a certain threshold, in the general case the neuron value is given by an activation function f(.).

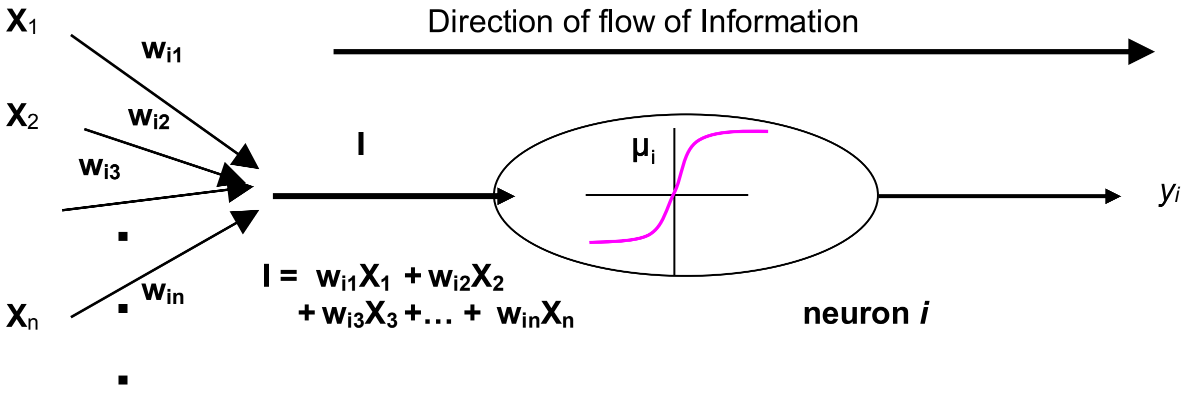


Figure 2: McCulloch-Pitts neuron model

Mathematically one can represent McCulloch-Pitts neuron model as

$$y_i = \Theta\left(\sum_j w_{ij} x_j - \mu_i\right) \tag{4.1}$$

where, y_i is output of neuron *i*, w_{ij} is weight from neuron j to neuron i, x_j is output of neuron j, μ_i is threshold for neuron i, and Θ is the activation/Transfer function.

(b) Activation/Transfer Function

The behaviour of an ANN depends on both the weights and the input-output function (transfer function) that is specified for the units. This function typically falls into one of three categories namely Linear, Threshold, and Sigmoid. For linear units, the output activity is proportional to the total weighted input. In threshold unit, the output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value. For sigmoid units, the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units, but all three must be considered rough approximations.

(c) Architecture of neural networks

(i) Feed-forward artificial neural network

Feed-forward artificial neural network allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward artificial neural network tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.

(ii) Feedback/Recurrent artificial neural networks

Feedback artificial neural networks can have signals traveling in both directions by introducing loops in the network. Feedback artificial neural networks are very powerful and can get extremely complicated. Feedback artificial neural networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations.

(d) Learning Algorithms

The weights in an ANN, similar to coefficients in a regression model, are adjusted to solve the problem presented to ANN. *Learning* or *training* is term used to describe process of finding values of these weights. Two types of learning with ANN are *supervised* and *unsupervised* learning.

(i) Supervised learning

Supervised learning incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be. During the learning process global information may be required. Paradigms of supervised learning include error-correction learning, reinforcement learning and stochastic learning. An important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of error between the desired and computed unit values. The aim is to determine a set of weights which minimizes the error. One well-known method, which is common to many learning paradigms, is the least mean square (LMS) convergence. Supervised learning occurs when there is a known target value associated with each input in training set. Output of ANN is compared with target value, and this difference is used to train ANN (alter the weights).

(ii) Unsupervised learning

Unsupervised learning uses no external teacher and is based upon only local information. It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their emergent collective properties. We say that a neural network learns off-line if the learning phase and the operation phase are distinct. A neural network learns on-line if it learns and operates at the same time. Usually, supervised learning is performed off-line, whereas unsupervised learning is performed on-line. Unsupervised learning is needed when training data lack target output values corresponding to input patterns. ANN must learn to group or cluster input patterns based on some common features, similar to factor analysis and principal component analysis.

5. Organization of Thesis

In chapter 2, theory of fuzzy sets and Possibilistic regression analysis is discussed. Three methods, viz. Minimization, Maximization, and Conjunction are considered. The approaches to solve these problems are outlined. The methodology is applied for modelling cotton crop yield data at block levels of Sirsa district, Haryana.

In chapter 3, Possibility and Necessity measures for obtaining reliable fuzzy estimates of crop yield have been thoroughly studied. Estimation of parameters is carried out using “Fuzzy least-squares” procedure. As an illustration, the methodology is applied for modelling of Pearl Millet crop yield data at block level of Bhiwani district, Haryana.

In chapter 4, a modified fuzzy least-squares approach for estimation of parameters is thoroughly studied. Relevant computer program has been developed in SAS/IML software package. Performance evaluation criterion based on “Difference in membership functions” is adopted for computation of error in estimation. As an illustration, the methodology is applied for modelling of Pearl Millet crop yield data at block level of Bhiwani district, Haryana.

In chapter 5, fitting of fuzzy von Bertalanffy growth model is done when response variable is reported in intervals corresponding to various values of explanatory variable. An efficient two-stage procedure proposed by Kao and Chyu, based on fuzzy least squares, is employed. The methodology is thoroughly discussed and, for its application, relevant computer programs are developed in “Nonlinear programming solver LINGO, Version 8” software package. Finally, as an illustration fuzzy von Bertalanffy growth model was fitted to pearl oyster age-length data.

In chapter 6, a particular type of ANN, viz. Multilayered feedforward artificial neural network (MLFANN) is thoroughly studied. In order to train such a network, two types of learning algorithms, namely Gradient descent algorithm (GDA) and Conjugate gradient descent algorithm (CGDA), are described. The methodology is illustrated by considering Maize crop yield data as response variable and Total human labour, Farm power, Fertilizer consumption, and Pesticide consumption as predictors. The data are taken from a recently concluded National Agricultural Technology Project of Division of Agricultural Economics, I.A.R.I., New Delhi. To train the neural network, relevant computer programs are written in MATLAB software package using its neural network toolbox.

In chapter 7, an important model from this class, viz. Adaptive Neuro-fuzzy inference system (ANFIS) is thoroughly studied. The model is implemented on Fuzzy Logic Toolbox of MATLAB using ANFIS. As an illustration, the methodology is applied for development of a forecasting model for secondary data of yield of banana plants on the basis of data at six different stages of growth using several biometrical characters, like plant height, plant girth and leaf length taken as predictors.

CHAPTER II

POSSIBILISTIC LINEAR REGRESSION ANALYSIS WITH FUZZY RESPONSE VARIABLE FOR CROP YIELD ESTIMATION

1. Background

Presently, more than five lakhs Crop-cutting experiments (CCE) in respect of principal crops of foodgrains, oilseeds and horticultural crops, etc. are conducted every year in our country by Directorate of Economics & Statistics, Ministry of Agriculture to arrive at crop yield estimates at district level. With the growing demand for micro-level planning, the need for building reliable estimates at small area level (Rao, 2003), say block or even gram panchayat is imperative. To achieve this, the number of CCE has to be increased many folds, which is not practicable.

Accordingly, Sud *et al.* (2006) carried out a detailed study for developing crop yield estimates at small area level using farmers' estimates. However, this would be meaningful only when inquiry-based farmers' estimates can be used successfully for modeling actual yield based on CCE. To this end, we consider cotton crop yield data at block levels of Sirsa district of Haryana State for Kharif season during 2003-04 given in Sud *et al.* (2006). For each of seven blocks, viz. Ellenabad, Rania, Baraguda, Sirsa, Nathusari, Odan, and Dabwali, farmers' estimate was described by a "*Crisp*" value. Further, the yield at block level based on CCE carried out in selected villages may be considered as fuzzy variable. The significance of fuzzy variables is that they facilitate gradual transitions of actual yield and possess a natural capability to express and deal with measurement uncertainties.

As response variable, viz. actual cotton yield based on CCE is a fuzzy number, which belongs in interval, linear regression analysis *per se* is not applicable. One way out is to convert interval values into crisp values by taking their mean values. However, in doing so a lot of vital information regarding variability is lost. Also in this method the uncertainties intrinsic to data would not be taken care off. The fuzzy paradigm allows us to express irreducible yield and measurement uncertainties by fuzzy numbers. Fortunately, the modeling problem can be handled through fuzzy linear regression analysis (Tanaka *et al.*, 1982). In this approach, either the phenomenon is considered fuzzy, or only response variable is considered fuzzy, or both response and explanatory variables are considered fuzzy.

First possibility when only the phenomenon is considered as fuzzy, was studied by Kandala and Prajneshu (2002) for crop yield forecasting based on remotely sensed data. The present study falls under the second category in which only the response variable is considered as fuzzy. Specifically, an attempt has been made in this chapter towards developing Possibilistic linear regression model of fuzzy data.

2. Fuzzy Sets and Possibilistic Linear Regression

2.1 Fuzzy sets

Classical set theory is based on the concept of *crisp set*, which assigns a value of either *1* or *0* to each individual in the universal set. The definition of a *crisp set* can be generalized such that the values assigned to elements of universal set fall within a specified range and indicate the membership grade of these elements in the set in question. Generally, the range of membership grade is taken to be *[0, 1]* and the membership function of a fuzzy set *A* is denoted by μ_A, i.e. a fuzzy set *A* is defined by

$$\mu_A(x): \quad X \to [0,1], \qquad (2.1)$$

where x is an element of *X*.

In fuzzy set theory, fuzzification of a *crisp function* $f : X \to Y$ is defined in the following way:

Let $\Im(X)$ and $\Im(Y)$ be the set of all fuzzy sets defined in the universal sets X and Y respectively. Then the *crisp function f* defined above induces the following functions,

$$f : \Im(X) \to \Im(Y),$$

$$f^{-1} : \Im(Y) \to \Im(X),$$

which are defined by

$$[f(A)](y) = \begin{cases} \displaystyle\sup_{x:y=f(x)} \mu_A(x) & \text{if } f^{-1}(y) \neq \phi \\ 0 & \text{if } f^{-1}(y) = \phi \end{cases} \qquad \text{for all } A \in \Im(X),$$

and $$\left[f^{-1}(B) \right](x) = \mu_B(f(x)) \qquad \text{for all } B \in \Im(Y)$$

Usually fuzzy sets, whose membership is triangular in nature, are used in applications due to its known properties and successful performance in linear fuzzy regression models (Tanaka *et al.*, 1982).

2.2 Possibilistic Regression Model

In the conventional regression model, deviations between observed and estimated values are due to several causes, like non-inclusion of all relevant explanatory variables, violation of assumption of linearity, and measurement errors. In possibility theory, these deviations are characterized as fluctuation of system parameters, which can be represented by a fuzzy number. Accordingly, it has become important to deal with fuzzy data originated from a fuzzy phenomenon. A model of such a fuzzy phenomenon might be represented as a fuzzy system equation, which can be described as possibility measures (Tanaka *et al.,* 1989). A possibility measure is different from assignment of membership grades of elements in fuzzy sets with ill-defined boundaries. In fact, it is a measure of a crisp set, which signifies the degree of evidence or belief that a particular element belongs in the set (Klir and Yuan, 2000). Possibility measure satisfies the equation

$$Pos(\phi) = 0, Pos(X) = 1 \text{ and } Pos(A \cup B) = max\{Pos(A), Pos(B)\} \tag{2.2}$$

It is known that every possibility measure is determined by the general equation

$$Pos(A) = \underset{x \in A}{Sup}\, r(x) \tag{2.3}$$

where, $r : X \rightarrow [0,1]$ is called possibility distribution. In our case, we shall see that regression parameters are taken as fuzzy numbers, which gives a fuzzy regression model. Thus data from fuzzy phenomenon is modeled through estimated fuzzy numbers and membership functions can be regarded as possibility distribution of the fuzzy system. Thus, a Possibilistic system is employed as a model of fuzzy phenomenon. The fitted fuzzy model is capable to describe uncertainty that the output belongs to two crisp intervals when it coincides with or is in close proximity to one of the boundaries $[a, a_1), [a_1, a_2), ..., [a_{n-1}, b]$, say. This uncertainty has all the characteristics of a possibility measure.

3. Formulation of Possibilistic Linear Regression Model

A linear system is defined by

$$Y_i = A_1 x_{i1} + \ldots + A_n x_{in} = \boldsymbol{A}\,\boldsymbol{x}_i, \qquad i = 1,\ldots,m \tag{3.1}$$

where x_{ij}, $i = 1,\ldots,m; j = 1,\ldots,n$ is real number. In fuzzy linear regression analysis, structure of the system is considered as fuzzy, which is represented as fuzzy function of input with parameters being triangular fuzzy numbers. A fuzzy number is a fuzzy set defined on real line where there is a unique $x \in R$, such that $\mu_A(x) = 1.$ A symmetric fuzzy number $\widetilde{A}$, denoted by $\widetilde{A} = (\alpha, c)$ is defined as $\mu_{\widetilde{A}}(x) = L((x-\alpha)/c), c > o$ where α is a center, c is a spread, and $L(x)$ has the following properties: (i) $L(x) = L(-x)$, (ii) $L(0) = 1$, (iii) $L(x)$ is strictly decreasing function for $x \geq 0$, and (iv) L is invertible on [0,1]. This fuzzy number can well explain a fuzzy set of the type “approximately α” due to decreasing nature of membership function $L(\,.\,)$ with respect to $|x-\alpha|$. This has been used for representation of fuzzy parameter in fuzzy linear regression. In modeling of output and input data, one of the objectives should be to find out estimated fuzzy number based on input data such that output has high membership values in estimated fuzzy number. This concept of belonging to a fuzzy set with high membership grade is called the concept of h-cut, denoted by ${}^{h}A = \{x : \mu_A(x) \geq h\}$.

Corresponding to (3.1), fuzzy linear model denoted as $\widetilde{Y}_i = f\left(\boldsymbol{x}_i, \widetilde{\boldsymbol{A}}\right)$, where $\widetilde{\boldsymbol{A}} = \left(\widetilde{A}_1, \ldots, \widetilde{A}_n\right)^t$, where superscript *t* denotes transpose and $\widetilde{A}_j = \left(\alpha_j, c_j\right)$ can be defined by following extension principle:

$$\mu_{\widetilde{Y}}(y) = \underset{\left\{(a_1,\ldots,a_n)^t \,:\, y = f(x, a_1,\ldots,a_n)^t\right\}}{Sup} \left(min\left(\mu_{\widetilde{A}_1}(a_1), \ldots, \mu_{\widetilde{A}_n}(a_n)\right)\right),$$

$$= 0, \qquad if \ \left\{ (a_1,\ldots,a_n)^t : y = f(x, a_1,\ldots,a_n) \right\} = \phi \tag{3.2}$$

Specifically, the membership function of the fuzzy set $\left(\widetilde{A}_1, \ldots, \widetilde{A}_n\right)^t$ is taken to be $\mu_{\left(\widetilde{A}_1,\ldots,\widetilde{A}_n\right)^t}(a_1,\ldots,a_n)^t = Min\left(\mu_{\widetilde{A}_1}(a_1), \ldots, \mu_{\widetilde{A}_n}(a_n)\right)$. Using (3.2), Tanaka *et al.* (1982) and Tanaka (1987) have shown that fuzzy output $\widetilde{Y}$ of system (3.1) can be represented as

$$\widetilde{Y} = \left(\alpha_1 x_1 + \ldots + \alpha_n x_n, c_1|x_1| + c_2|x_2| + \ldots + c_n|x_n|\right) = \left(\boldsymbol{\alpha}^t \boldsymbol{x}, \boldsymbol{c}^t |\, \boldsymbol{x} \,|\right) \tag{3.3}$$

In this work, an attempt has been made to apply three approaches for formulating Possibilistic linear systems and compare performance of these approaches in the light of reliability of farmers' estimate for crop yield estimation. After obtaining Possibilistic linear models, Kim and Bishu (1998) approach has been followed to obtain differences in membership function of the actual versus fitted fuzzy numbers for performance evaluation.

Definition 1: A symmetric fuzzy number $\tilde{A}_2 = (\alpha_2, c_2)$ is included in the fuzzy number $\tilde{A}_1 = (\alpha_1, c_1)$ with a degree $0 \le h < 1$, iff ${}^h\tilde{A}_2 \subseteq {}^h\tilde{A}_1$ denoted as $\tilde{A}_2 \subseteq_h \tilde{A}_1$. This is equivalent to

$$\begin{aligned} \alpha_1 &\le \alpha_2 + \left| L^{-1}(h) \right| (c_1 - c_2) \\ \alpha_1 &\ge \alpha_2 - \left| L^{-1}(h) \right| (c_1 - c_2) \end{aligned} \tag{3.4}$$

where $\left| L^{-1}(h) \right|$ is a negative valued inverse function of $L(h)$.

Let Y_i be an output (response variable) or an observation for *i-th* sample and x_{ij} be *j-th* input (explanatory variable) for *i*-th sample. The basic concept for formulating three Possibilistic linear regressions is to use mutual relation between observed and estimated intervals, which are obtained by inclusion of fuzzy numbers with a degree $h,\ 0 \le h < 1$. Our idea is to obtain fuzzy parameters for each of three Possibilistic linear regressions. The fuzzy parameters, denoted as $\overline{A}_i,\ \underline{A_i},\ and\ \hat{A}_i$ such that, for each of $i = 1,2,...,m$.

$$\tilde{Y}_i \subseteq_h \tilde{\overline{\tilde{Y}}}_i = \tilde{\overline{A}}_1 x_{i1} + \ldots + \tilde{\overline{A}}_n x_{in}\ , \tag{3.5}$$

$$\tilde{Y}_i \supseteq_h \underline{\tilde{Y}_i} = \underline{\tilde{A}_1} x_{i1} + \ldots + \underline{\tilde{A}_n} x_{in}\ , \tag{3.6}$$

$${}^h[Y_i] \cap {}^h\left[\hat{Y}_i = \hat{A}_1 x_{i1} + \ldots + \hat{A}_n x_{in}\right] \ne \phi \tag{3.7}$$

Equations (3.5) and (3.6) mean that h-level set of all given outputs $\left(\tilde{Y}_i\right)$ are covered and contained by corresponding estimated h-level set of fuzzy numbers $\left(\tilde{\hat{Y}}_i\right)$. Furthermore, (3.7) implies that all intersections of h-level set of given output $\tilde{Y}_i$ and estimated fuzzy number $\tilde{\hat{Y}}_i$ is not empty. The mutual relations given by (3.5)-(3.7) are illustrated in Figure 1. The formulation derived from (3.5)-(3.7) are respectively called Minimization problem, Maximization problem and Conjunction problem.

Minimization

Conjunction

Maximization

Fuzzy data

Figure 1: Concepts of Minimization, Maximization and Conjunction problems where shaded and un-shaded regions indicate *h*-level sets of observed and estimated fuzzy numbers

3.1 Minimization Problem

Let output $\tilde{Y}_i$ be denoted by fuzzy number $\tilde{Y}_i = \left(y_i, e_i\right)$, where y_i is centre and e_i is spread for $i = 1,...,m$. Consider following optimization problem

$$\underset{\overline{A}_j=(\overline{\alpha}_j,\overline{c}_j)}{Min} \overline{J}(\overline{\boldsymbol{c}}) = \sum \overline{\boldsymbol{c}}^{\,t} | \boldsymbol{x}_i |$$

where $\overline{\boldsymbol{c}}^t | \boldsymbol{x}_i |$ is spread of estimated fuzzy output $\overline{Y}_i$, subject to following three constraints:

$$y_i + e_i \left| L^{-1}(h) \right| \leq \overline{\alpha} x_i + \overline{\boldsymbol{c}}^{\,t} | x_i | \left| L^{-1}(h) \right| ,$$

$$y_i - e_i \left| L^{-1}(h) \right| \geq \overline{\alpha} x_i - \overline{\boldsymbol{c}}^{\,t} | x_i | \left| L^{-1}(h) \right| ,$$

$$\overline{\boldsymbol{c}} \geq \boldsymbol{0}, \quad i = 1,...,m \tag{3.8}$$

Vagueness of Possibilistic linear model $\tilde{\tilde{Y}}_i = \tilde{\tilde{A}}_1 x_1 + ... + \tilde{\tilde{A}}_n x_n$ is defined by $J = c_1 + ... + c_n$ (Tanaka *et al., 1982*). The quantity *J* is minimized subject to $\overline{h}_i \geq h$, $i = 1,...,m$ where, $\overline{h}_i$ is the maximum value such that $\tilde{Y}_i \subseteq_h \tilde{\tilde{Y}}_i$, and h is chosen by the decision-maker. The fitness level of fuzzy linear model to data $\tilde{Y}_1, ..., \tilde{Y}_m$ is defined by $\underset{i}{min}\ \overline{h}_i$. Under symmetric triangular fuzzy number set up of the parameters, the constraints in (3.8) reduce to that of Tanaka's approach. But Tanaka approach has not considered the effect of input on the estimated spread which is $c^{\,t} | x_i |$, $i = 1,...,m$ and hence it is more realistic to minimize $\sum \boldsymbol{c}^t | \boldsymbol{x}_i |$, in lines with Tanaka *et al.* (1989).

3.2 Maximization problem

Consider following optimization problem

$$\underset{\underline{A_j}=\left(\underline{\alpha_j},\underline{c_j}\right)_L}{Max} \quad \underline{J}(\underline{\boldsymbol{c}}) = \sum \underline{\boldsymbol{c}^t} \mid \boldsymbol{x}_i \mid$$

subject to following three constraints:

$$y_i + e_i \left| L^{-1}(h) \right| \geq \underline{\alpha} x_i + \underline{\boldsymbol{c}^t} \mid x_i \mid \left| L^{-1}(h) \right| ,$$

$$y_i - e_i \left| L^{-1}(h) \right| \leq \underline{\alpha} x_i - \underline{\boldsymbol{c}^t} \mid x_i \mid \left| L^{-1}(h) \right| ,$$

$$\underline{\boldsymbol{c}} \geq \boldsymbol{0}, \quad i = 1,\dots,m \tag{3.9}$$

Under triangular fuzzy number representation of parameters, this problem may be viewed as maximizing $\underline{J}(\underline{\boldsymbol{c}})$ subject to $\underline{h_i} \geq h$, $i = 1,\dots,m$, where $\underline{h_i}$ is maximum value such that ${}^{h_i}\widetilde{Y}_i \supseteq {}^{h_i}\underline{\widetilde{Y}_i}$ and is provided by the decision-maker.

3.3 Conjunction problem

Under symmetric triangular fuzzy number set up, the conjunction problem may be viewed as to minimize $\hat{J}(\hat{c})$ such that the degree of fitting $\hat{h}_i$ defined by the maximum value for which ${}^{h}[Y_i] \cap {}^{h}\left[\hat{Y}_i = \hat{A}_1 x_{i1} + \dots + \hat{A}_n x_{in}\right] \neq \phi$ is at least h, as shown in Figure 2.

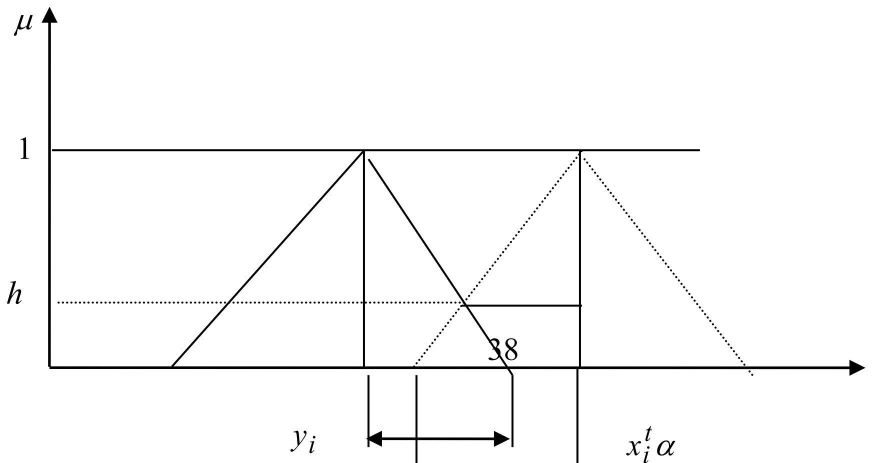


Figure 2: Degree of fitting of observed (in solid lines) and estimated (in dotted lines) fuzzy numbers for Conjunction problem

Evidently, from above figure it follows

$$1:\left(1-\hat{h}_i\right)=\left(\sum_j c_j\left|x_{ij}\right|\right):l \tag{3.10}$$

where

$$l=\left|y_i-x^t\alpha\right|-e_i\left(1-\hat{h}_i\right)$$

Substituting the value of l in (3.10), the expression for $\hat{h}_i$, $i=1,\dots,m$ can be obtained as

$$\hat{h}_i=1-\frac{\left|y_i-x_i^t\alpha\right|}{\sum_j c_j\left|x_{ij}\right|+e_i}, \tag{3.11}$$

Finally conjunction problem is to minimize the following:

$$\underset{\hat{A}_j=\left(\hat{\alpha}_j,\hat{c}_j\right)}{Min}\hat{J}(\hat{\boldsymbol{c}})=\sum\hat{\boldsymbol{c}}^{\,t}\mid\boldsymbol{x}_i\mid$$

subject to following three constraints:

$$y_i+e_i\left|\,L^{-1}(h)\right|\geq\hat{\alpha}x_i-\hat{\boldsymbol{c}}^{\,t}\mid x_i\mid\left|\,L^{-1}(h)\right|,$$

$$y_i - e_i \left| L^{-1}(h) \right| \le \hat{\alpha} x_i + \hat{\boldsymbol{c}}^{\,t} \mid x_i \mid \left| L^{-1}(h) \right| , \qquad (3.12)$$

$$\hat{\boldsymbol{c}} \ge \boldsymbol{0}, \quad i = 1, \ldots, m$$

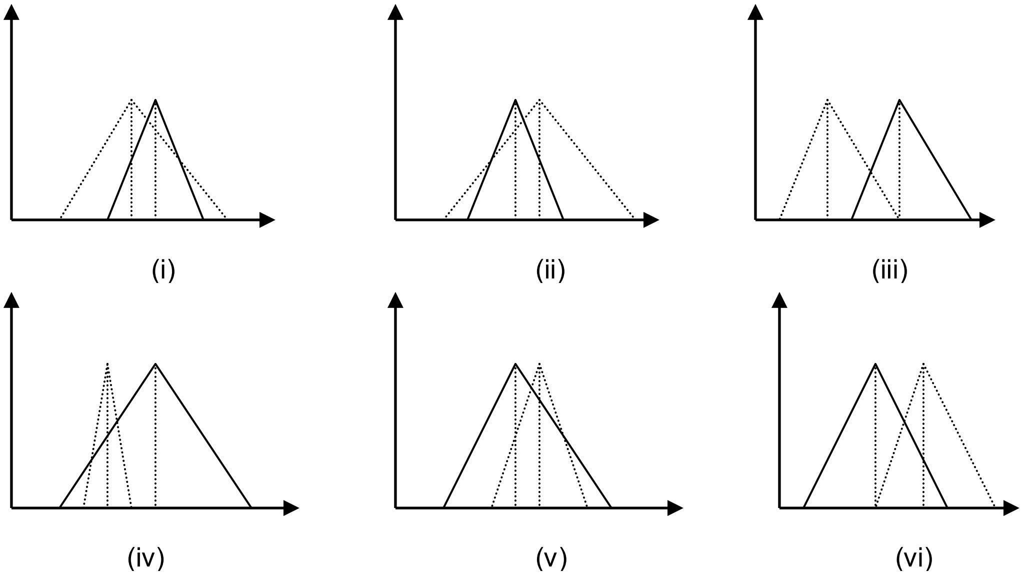
It is to be noted that, under symmetric triangular fuzzy numbers set up constraints in (3.12) reduce to constraints arising from the inequality $\hat{h}_i \ge h$, $i = 1,...,m$ where $\hat{h}_i$ is represented as in (3.11).

4. Performance Evaluation of Fuzzy Regression

In a fuzzy linear regression model, values of response variable are represented as fuzzy numbers with membership functions characterized by explanatory variable. In order to evaluate the closeness of observed and estimated fuzzy numbers, support of both fuzzy numbers should be close to each other, where support of a fuzzy set A is defined by $S_A = \{x : \mu_A(x) > 0\}$. Therefore, for performance evaluation of a fuzzy regression model, Kim and Bishu (1998) used ratio of difference between membership values to observed membership values as follows:

$$E_i = \frac{\displaystyle\int_{S_{\hat{\tilde{Y}}_i} \cup S_{\tilde{Y}_i}} \left| \hat{\tilde{Y}}_i(y) - \tilde{Y}_i(y) \right| dy}{\displaystyle\int_{S_{\tilde{Y}_i}} \tilde{Y}_i \; dy} = \frac{N_i}{D_i}, \text{ say} \qquad (4.1)$$

where $S_{\hat{\tilde{Y}}_i}$ and $S_{\tilde{Y}_i}$ are the support of $\hat{\tilde{Y}}_i$ and $\tilde{Y}_i$, respectively. It can easily be seen that D_i in (4.1) reduces to spread e_i in the case of observed triangular fuzzy numbers. The term N_i in equation (4.1) represents the difference of observed and estimated fuzzy responses. Also, various possibility of overlap between observed and estimated fuzzy values are depicted in Figure 3. Expression for N_i may be evaluated corresponding to each of the above possibilities by noticing the fact that N_i is equal to sum of areas of observed and estimated responses subtracting twice area of intersection (Kao and Chyu, 2002).


(i) (ii) (iii)

(iv) (v) (vi)

Figure 3: Different ways to depict the difference between observed and estimated membership function.

We now discuss only one of the above six possibilities, namely the last one in details. However, the algebra for remaining five possibilities follows along similar lines. The enlarged form of case (vi) of Figure 3 is depicted in Figure 4.

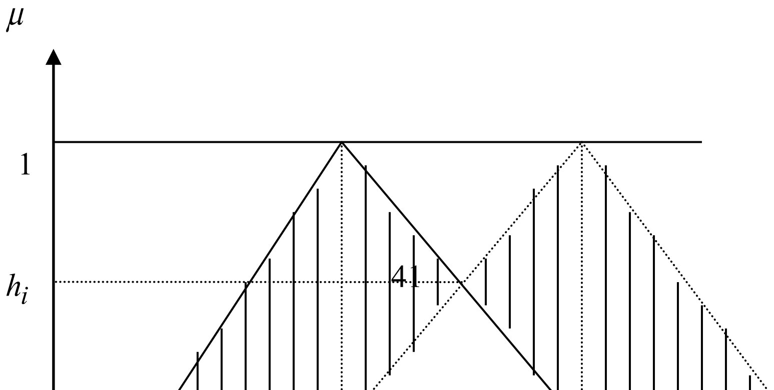


Figure 4: Difference of the membership functions between the observed and estimated responses represented as shaded region.

The value of N_i, which represents the shaded area in above figure, is calculated as

$$N_i = 0.5\left[y_{iu} - y_{il}\right] + 0.5\left[\hat{y}_{iu} - \hat{y}_{il}\right] - 2\ (0.5)\ \left[y_{iu} - \hat{y}_{il}\right]\ h_i \qquad (4.2)$$

where h_i is membership value of intersection of right shape function of $\tilde{Y}_i$ and left shape function $\hat{\tilde{Y}}_i$. In symbols, we have

$$h_i = height\left(\tilde{Y}_i \cap \hat{\tilde{Y}}_i\right),$$

where *"height"* of a fuzzy set A is value of x such that $\mu_A(x) = \underset{y \in X}{Max}\ \mu_A(y)$.

Algebraically, h_i can be expressed as

$$h_i = \frac{\left(y_{iu} - \hat{y}_{il}\right)}{\left(y_{iu} - y_{im} + \hat{y}_{im} - \hat{y}_{il}\right)} \qquad (4.3)$$

5. Results and Discussion

Sud *et al.* (2006) undertook a study on the important crops grown in Rabi season during agricultural year 2002-03 and Kharif season during 2003-04 in Bhiwani and Sirsa districts of Haryana State. Here, a part of data concerned with yield of cotton crop at block levels of Sirsa district was considered to develop a fuzzy estimate of cotton yield based on farmers' estimates. Seven blocks in Sirsa district are: Ellenabad, Rania, Baraguda, Sirsa, Nathusari, Odan and Dabwali. The number of farmers selected were as follows: Ellenabad (22 farmers), Rania (24 farmers), Baraguda (22 farmers), Sirsa (22 farmers), Nathusari (20 farmers), Odan (24 farmers) and Dabwali (26 farmers) The explanatory variable, at block level, is farmers' estimate while response variable at the same level is actual cotton crop yield based on Crop-cutting experiments, and are fuzzy numbers. Fuzzy linear regression models are constructed using Minimization, Maximization and Conjunction methods. The entire data analysis is carried out using LINGO, Version 8, software package (LINDO, 2002) available at I.A.S.R.I., New Delhi. The data for present investigation, culled from Sud *et al*. (2006), is reproduced in Table 1 for ready reference.

Table 1. Cotton yield (based on CCE) as triangular fuzzy numbers with farmers' estimates for Sirsa district

Blocks	*Farmers' Estimate (quintals/hectare)*	*Lower limit of yield (quintals /hectare)*	*Upper limit of yield (quintals /hectare)*
Ellenabad	14.25	15.26	19.44
Rania	16.80	13.67	16.92
Baraguda	16.97	15.83	20.31
Sirsa	17.56	15.91	17.59
Nathusari	16.67	13.53	16.57

Odan	14.40	12.53	15.60
Dabwali	9.12	10.72	13.35

Yield as function of farmers' estimates can be expressed as follows:

$$\widetilde{Y}_i = (\alpha_1, c_1) + (\alpha_2, c_2)\, x_i , \quad i = 1,\ldots,7 \tag{5.1}$$

As h lies in the interval [0, 1] and the two end-points, viz. h=0 and h=1 are only of academic interest, we perform computations for, say four equidistant values of h, viz. h=*0.2, 0.4, 0.6 and 0.8* in our subsequent analysis. Minimization, Maximization and Conjunction problems are formulated and solved below:

5.1 Solution of Minimization problem

The problem is formulated as follows:

$$Min\ \bar{J}(\bar{h}) = \sum_{i=1}^{7} (\bar{c}_1 * 1 + \bar{c}_2 * x_i), \quad i = 1,\ldots,7$$

subject to

$$17.35 + 2.09 \left| L^{-1}(h) \right| \le (\bar{\alpha}_1 * 1 + \bar{\alpha}_2 * 14.25) + (\bar{c}_1 * 1 + \bar{c}_2 * 14.25) \left| L^{-1}(h) \right| ,$$
$$17.35 - 2.09 \left| L^{-1}(h) \right| \ge (\bar{\alpha}_1 * 1 + \bar{\alpha}_2 * 14.25) - (\bar{c}_1 * 1 + \bar{c}_2 * 14.25) \left| L^{-1}(h) \right| ,$$

.

.

.

$$12.04 + 1.32 \left| L^{-1}(h) \right| \le (\bar{\alpha}_1 * 1 + \bar{\alpha}_2 * 9.12) + (\bar{c}_1 * 1 + \bar{c}_2 * 9.12) \left| L^{-1}(h) \right| ,$$
$$12.04 - 1.32 \left| L^{-1}(h) \right| \ge (\bar{\alpha}_1 * 1 + \bar{\alpha}_2 * 9.12) - (\bar{c}_1 * 1 + \bar{c}_2 * 9.12) \left| L^{-1}(h) \right| ,$$

$\bar{c}_1 , \bar{c}_2 \ge 0$.

Solving above equations using LINGO software package, the estimated models corresponding to h*=0.2, 0.4, 0.6 and 0.8*, are respectively obtained as:

$$
\begin{aligned}
\hat{\tilde{\tilde{Y}}}_i &= (11.26, 4) + (0.32, 0)x_i\,,\\
\hat{\tilde{\tilde{Y}}}_i &= (11.04, 4.71) + (0.33, 0)x_i\,,\\
\hat{\tilde{\tilde{Y}}}_i &= (10.83, 6.14) + (0.34, 0)x_i\,,\\
\hat{\tilde{\tilde{Y}}}_i &= (10.26, 10.42) + (0.36, 0)x_i \quad \text{for } i = 1,\ldots,7 \qquad (5.2)
\end{aligned}
$$

5.2 Solution of Maximization problem

The problem is formulated as follows:

$$\text{Max } \underline{\text{J}}(\underline{h}) = \sum_{i=1}^{7} \left(\underline{c_1} * 1 + \underline{c_2} * x_i\right)\,,\ i = 1,\ldots,7$$

subject to

$$17.35 + 2.09 \left| L^{-1}(h) \right| \geq \left(\underline{\alpha_1} * 1 + \underline{\alpha_2} * 14.25\right) + \left(\underline{c_1} * 1 + \underline{c_2} * 14.25\right) \left| L^{-1}(h) \right|,$$

$$17.35 - 2.09 \left| L^{-1}(h) \right| \leq \left(\underline{\alpha_1} * 1 + \underline{\alpha_2} * 14.25\right) - \left(\underline{c_1} * 1 + \underline{c_2} * 14.25\right) \left| L^{-1}(h) \right|,$$

.

.

.

$$12.04 + 1.32 \left| L^{-1}(h) \right| \geq \left(\underline{\alpha_1} * 1 + \underline{\alpha_2} * 9.12\right) + \left(\underline{c_1} * 1 + \underline{c_2} * 9.12\right) \left| L^{-1}(h) \right|,$$

$$12.04 - 1.32 \left| L^{-1}(h) \right| \leq \left(\underline{\alpha_1} * 1 + \underline{\alpha_2} * 9.12\right) - \left(\underline{c_1} * 1 + \underline{c_2} * 9.12\right) \left| L^{-1}(h) \right|,$$

$$\underline{c_1}, \underline{c_2} \geq 0$$

It is found that, for maximization problem, feasible solution does not exist for this data set. Tanaka and Watada (1989) also pointed out that optimum solution for maximization problem need not always exist.

5.3 Solution of Conjunction problem

The problem is formulated as follows:

$$Min\ \hat{J}(\hat{c}) = \sum_{i=1}^{7} (\hat{c}_1 * 1 + \hat{c}_2 * x_i),\ i = 1,\ldots,7$$

subject to

$$17.35 + 2.09\ \left|L^{-1}(h)\right| \le (\hat{\alpha}_1 * 1 + \hat{\alpha}_2 * 14.25) - (\hat{c}_1 * 1 + \hat{c}_2 * 14.25)\ \left|L^{-1}(h)\right|\ ,$$
$$17.35 - 2.09\ \left|L^{-1}(h)\right| \ge (\hat{\alpha}_1 * 1 + \hat{\alpha}_2 * 14.25) + (\hat{c}_1 * 1 + \hat{c}_2 * 14.25)\ \left|L^{-1}(h)\right|\ ,$$

.
.
.

$$12.04 + 1.32\ \left|L^{-1}(h)\right| \le (\hat{\alpha}_1 * 1 + \hat{\alpha}_2 * 9.12) - (\hat{c}_1 * 1 + \hat{c}_2 * 9.12)\ \left|L^{-1}(h)\right|\ ,$$
$$12.04 - 1.32\ \left|L^{-1}(h)\right| \ge (\hat{\alpha}_1 * 1 + \hat{\alpha}_2 * 9.12) + (\hat{c}_1 * 1 + \hat{c}_2 * 9.12)\ \left|L^{-1}(h)\right|\ ,$$

$$\hat{c}_1,\ \hat{c}_2 \ge 0$$

Solving above equations using LINGO software package, the estimated models corresponding to h=*0.2,0.4,0.6 and 0.8*, are respectively obtained as:

$$\hat{\tilde{Y}}_i = (9.54, 0.29) + (0.41, 0)x_i\ ,$$
$$\hat{\tilde{Y}}_i = (9.54, 1.00) + (0.40, 0)x_i\ ,$$
$$\hat{\tilde{Y}}_i = (9.97, 2.43) + (0.39, 0)x_i\ ,$$
$$\hat{\tilde{Y}}_i = (10.12, 6.70) + (0.38, 0)x_i \quad \text{for } i = 1,\ldots,7 \qquad (5.3)$$

Substituting the values of farmers’ estimates in (5.2) and (5.3), estimated fuzzy cotton yield corresponding to Minimization and Conjunction methods are obtained and reported in Table 2. Further, using (4.1), error in estimation E_i for each fitness value *h* is computed for both the methods and the same are also reported in Table 2.

Table 2. Estimated fuzzy yields along with error in estimation for Minimization and Conjunction methods

h	*Blocks*	*Estimated fuzzy yield (quintals /hectare)*			E_i
		$\hat{\bar{Y}}_{il}$	$\hat{\bar{Y}}_{im}$	$\hat{\bar{Y}}_{iu}$	
0.2	Ellenabad	11.82 (15.16)	15.82(15.45)	19.81(15.73)	1.32 (1.09)
	Raina	12.64(16.22)	16.63(16.50)	20.63(16.79)	1.73(1.03)
	Baraguda	12.69(16.29)	16.69(16.58)	20.69(16.86)	1.13(0.99)
	Sirsa	12.88(16.54)	16.88(16.82)	20.87(17.11)	3.75(0.67)
	Nathusa	12.60(16.17)	16.59(16.45)	20.59(16.74)	1.98(1.13)
	Odan	11.87(15.23)	15.86(15.51)	19.86(15.80)	2.00(1.13)
	Dabwali	10.18(13.04)	14.18(13.32)	18.17(13.61)	2.68(1.17)
0.4	Ellenabad	11.03(14.24)	15.74(15.24)	20.45(16.24)	1.54(1.33)
	Raina	11.87(15.26)	16.58(16.26)	21.29(17.26)	2.07(0.97)
	Baraguda	11.93(15.33)	16.64(16.33)	21.35(17.33)	1.34(1.14)
	Sirsa	12.12(15.56)	16.83(16.56)	21.54(17.56)	4.60(0.42)
	Nathusa	11.83(15.21)	16.54(16.21)	21.25(17.21)	2.32(1.17)
	Odan	11.08(14.30)	15.79(15.30)	20.50(16.30)	2.24(1.22)
	Dabwali	9.34(12.19)	14.05(13.19)	18.76(14.19)	2.97(1.31)
0.6	Ellenabad	9.53(13.10)	15.67(15.53)	21.81(17.96)	2.10(1.39)
	Raina	10.40(14.09)	16.54(16.52)	22.68(18.95)	2.87(1.28)
	Baraguda	10.46(14.16)	16.60(16.59)	22.74(19.02)	1.88(1.11)
	Sirsa	10.66(14.39)	16.80(16.82)	22.94(19.25)	6.30(1.89)
	Nathusa	10.36(14.04)	16.50(16.47)	22.64(18.90)	3.16(1.53)
	Odan	9.58(13.15)	15.72(15.58)	21.86(18.01)	2.95(1.59)
	Dabwali	7.79(11.10)	13.93(13.53)	20.07(15.96)	3.86(1.81)
	Ellenabad	5.33(8.83)	15.75(15.53)	26.17(22.23)	4.03(2.37)

0.8	Raina	6.25(9.80)	16.67(16.50)	27.09(23.20)	5.45(3.19)
	Baraguda	6.31(9.87)	16.73(16.57)	27.15(23.27)	3.69(2.11)
	Sirsa	6.52(10.09)	16.94(16.79)	27.36(23.49)	11.39(6.97)
	Nathusa	6.20(9.76)	16.62(16.46)	27.04(23.06)	5.90(3.50)
	Odan	5.38(8.89)	15.80(15.59)	26.22(22.29)	5.48(3.25)
	Dabwali	3.48(6.89)	13.90(13.59)	24.32(20.29)	6.98(4.20)

Note: The values within brackets ( ) denote yield and error in estimation for Conjunction method.

The sum of errors in estimation for both minimization and conjunction methods, are depicted in Table 3. It may be noticed that it increases as *h* increases.

Table 3. Sum of errors in estimation for various formulations

h Formulation	0.2	0.4	0.6	0.8
Minimization	14.58	17.09	23.12	42.92
Conjunction	7.20	7.56	10.60	25.58

A perusal shows that values of sums of errors for Conjunction method are throughout lower than those for Minimization method. In other words, Conjunction approach gives more reliable estimates for crop yield vis-à-vis Minimization approach. It may also be noted that, for Conjunction method, sums of errors increase only slightly up to h as high as *0.6*. To get a visual idea, observed cotton yield and estimated yield from Conjunction method for h=0.6 are depicted in Figure 5. Evidently, the observed yields (in solid lines) and estimated yields (in dotted lines) are found to be quite close to each other, thereby indicating that farmers' estimates are able to explain actual crop yield with high fitness levels.

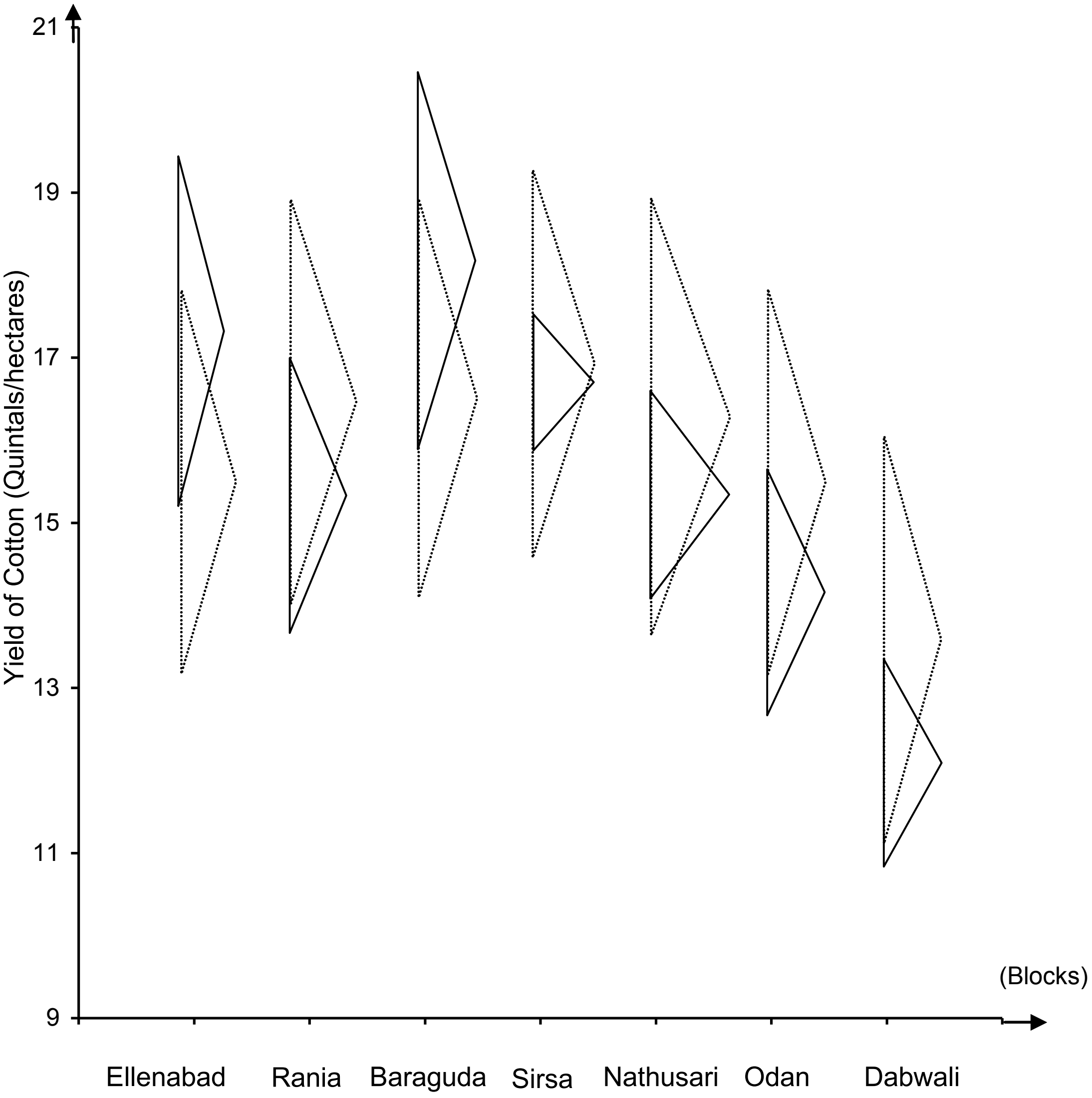


Figure 5: Observed (in solid lines) and Estimated (in dotted lines) cotton crop yields based on possibilistic regression model for Conjunction method

Concluding Remark: The extension of above work when explanatory variable is also fuzzy is in progress and shall be reported separately in due course of time.

CHAPTER III

POSSIBILITY AND NECESSITY MEASURES FOR FUZZY LINEAR REGRESSION ANALYSIS

1. Background

There is a growing demand in our country for estimation of crop yield at lower level, such as Community development block and in some cases even Gram panchayat. One alternative for meeting requirement of building up precise estimators at small area level is to increase number of crop-cutting experiments (CCE). However, this is usually not possible due to cost constraints. Therefore, a new cost effective technique needs to be developed, which may be adopted for implementation for estimation of yield of various crops at small area level (Rao, 2003). To this end, Sud *et al.* (2006) carried out a detailed study with the objective to develop precise estimates at block level using farmers' estimates. However, this would be meaningful only when inquiry based farmers' estimates are not very far off actual yield based on CCE. Ghosh *et al.* (2007) applied "Possibilistic linear regression model" considering yield data as fuzzy while farmers' estimate as 'crisp' value. Linear programming (LP) approach of Tanaka *et al.* (1982) for modelling of fuzzy variable was used for its inherent simplicity in terms of computation.

The main shortcoming of Tanaka's LP approach is that it is not based on sound statistical concepts (Chang and Ayyub, 2001). Therefore, another method for handling fuzzy data was developed by Diamond (1988) in which Fuzzy least-squares (FLS) criterion was considered. Kandala and Prajneshu (2004b) studied FLS method thoroughly and also applied it to some data. From a viewpoint of risk, Modarres *et al.* (2004) developed fuzzy linear regression model by considering Necessity measure of inclusion of observed fuzzy numbers in estimated fuzzy numbers. Further, Modarres *et al.* (2005) also extended a FLS with high fitness level of estimated model by incorporating the concept of Possibility theory. In this chapter, an attempt has been made to study FLS model using both the approaches of Possibility and Necessity measures. As an illustration, the above-mentioned methodology is applied for estimation of yield of Pearl Millet crop at block level for Bhiwani district in Haryana State using data given in Sud *et al.* (2006). Further, comparison of Possibility, Necessity, and Minimization approaches has been carried out for performance evaluation of FLS with optimal fitness level.

2. Possibility and Necessity of Events

2.1 Possibility measure

Let U be a universal set of elementary events. Any subset of U is called an event. An event $A \subseteq U$ is said to occur when some elementary event in A occurs. A possibility measure (Zadeh, 1965) on U is a set function Π from $\wp(U)$, the set of crisp subsets of U, to the unit interval [0, 1] , such that

$$\Pi(\phi) = 0,\ \Pi(U) = 1, \text{ and } \forall\, A, B \in \wp(U),\quad \Pi(A \cup B) = max(\Pi(A), \Pi(B)) \qquad (2.1)$$

Let F be a normalized fuzzy set with membership function $\mu_F(u)$ such that $\mu_F(u) = 1$ for some $u \in U$. Then, quantity $\Pi_F(A)$ derived from membership function $\mu_F(u)$ by

$$\Pi_F(A) = \underset{u \in A}{Sup}\ \mu_F(u) \qquad \forall\ A \subseteq U \qquad (2.2)$$

defines a possibility measure. Eq. (2.2) is interpreted as possibility of realizing event A when possibility of elementary events is expressed by fuzzy set F. Now, if suppose Π_F is crisp (i.e. $\Pi_F(u) \in \{0, 1\}$), then $\Pi_F(A) = 1 \Leftrightarrow A \cap F \neq \phi$. When both A and F are fuzzy, (2.2) can be readily extended, using fuzzy set intersection, to

$$\Pi_F(A) = \underset{u}{Sup}\ min(\mu_F(u), \mu_A(u)) \qquad (2.3)$$

Eq. (2.2) is a special case of eq. (2.3) and such an extension can be interpreted in terms of intersection of level cuts of F and A.

2.2 Necessity measure

A necessity measure (Dubois and Prade, 1980) is a set function $N : \wp(U) \rightarrow [0, 1]$ such that

$$N(\phi) = 0, N(U) = 1, \text{ and } N(A \cap B) = min(N(A), N(B))\ \ \forall\ A, B \subseteq U \qquad (2.4)$$

Let $\overline{A}$ be the complementary set of A, and Π be a possibility measure. Then it is easy to check that the set function N defined by

$$N(A) = 1 - \Pi(\overline{A}) \quad \forall \ A \subseteq U, \tag{2.5}$$

is a necessity measure. If Π derives from a normalized membership function μ_F, then it is obvious that $\forall \ A$,

$$N_F(A) = 1 - \Pi_F(\overline{A}) = \inf_{u \in \overline{A}} \left(1 - \mu_F(u)\right) \tag{2.6}$$

When A and F are crisp, then

$$N_F(A) = \begin{cases} 1, & \text{if } F \subset A \\ 0, & \text{otherwise} \end{cases} \tag{2.7}$$

Hence, while possibility is related to intersection, necessity refers to set inclusion. Eq. (2.5) can be extended, consistently with eq. (2.3), by defining

$$N_F(A) = 1 - \operatorname{Sup}_{u} \ \min\left(\mu_F(u), 1 - \mu_A(u)\right) = \inf_{u} \ \max\left(1 - \mu_F(u), \mu_A(u)\right) \tag{2.8}$$

which is a measure of fuzzy set F contained in fuzzy set A.

2.3 Possibilistic regression model

In the conventional regression model, deviations between observed and estimated values are due to several causes, like non-inclusion of all relevant explanatory variables, violation of assumption of linearity, and measurement errors. However in many real world situations, response/explanatory variables can not be taken as crisp values. For example, in modelling yield of Pearl Millet crop at bock level, it is always meaningful to consider yield as a fuzzy variable because there are many representative values of yield for a particular block from several villages obtained by CCE. Also, significance of yield to be expressed as fuzzy variables is that it facilitates gradual transitions of actual yield and possesses a natural capability to express and deal with measurement uncertainties. In possibility theory, these deviations are characterized as fluctuations of system parameters, which can be represented by a fuzzy number. Accordingly, it has become important to deal with fuzzy data originated from a fuzzy phenomenon. The formulation of Possibilistic linear regression model has been introduced by Tanaka *et al.* (1982). There are *m* explanatory non-fuzzy variables, $\boldsymbol{x}_i$, *i* =1, 2, . . . , *m*, while the response variables are symmetric fuzzy number, $\widetilde{Y}_i = (y_i, e_i)$. The objective is to estimate a fuzzy linear regression model, expressed as follows:

$$\hat{\widetilde{Y}}_i = \widetilde{A}_0\, x_{i0} + \widetilde{A}_1\, x_{i1} + \ldots + \widetilde{A}_n\, x_{in} = \widetilde{\boldsymbol{A}}\, \boldsymbol{x}_i\,,\ i = 1,\ldots,m \tag{2.9}$$

In model (2.9), $\widetilde{\boldsymbol{A}} = (\widetilde{A}_0, \widetilde{A}_1, \ldots, \widetilde{A}_n)$ is a vector of fuzzy parameters where $\widetilde{\mathrm{A}}_\mathrm{j} = (\alpha_\mathrm{j}, \mathrm{c}_\mathrm{j})$ is a symmetric fuzzy number with α_j as center and c_j as spread. Fuzzy parameters of model are estimated for certain fitness level *h*, $0 \leq h \leq 1$ such that *h*-level cut of estimated fuzzy number contains *h*-level cut of observed values. Problem is formulated as Minimization problem, which is given as follows:

$$\underset{\tilde{A}_j=\left(\alpha_j,\; c_j\right)}{Min} J(\boldsymbol{c})=\sum \boldsymbol{c}^{\,t}\,|\,\boldsymbol{x}_i\,| \tag{2.10}$$

where $\boldsymbol{c}^t|\,\boldsymbol{x}_i\,|$ is spread of estimated fuzzy output $\hat{\tilde{Y}}_i$, subject to following three constraints:

$$\begin{aligned}
&y_i+e_i\,\left|L^{-1}(h)\right| \le \boldsymbol{\alpha}^t x_i+\boldsymbol{c}^{\,t}\,|\,x_i\,|\left|L^{-1}(h)\right|\;,\\
&y_i-e_i\,\left|L^{-1}(h)\right| \ge \boldsymbol{\alpha}^t x_i-\boldsymbol{c}^{\,t}\,|\,x_i\,|\,\left|L^{-1}(h)\right|\;,\\
&\boldsymbol{c}\ge \boldsymbol{0},\quad i=1,\ldots,m
\end{aligned} \tag{2.11}$$

The main shortcoming of Tanaka’s linear programming approach as noticed by Chang and Ayyub (2001), is that number of constraints increases proportionately as number of data points increases leading to computational difficulties. The other drawback is that it is not statistically sound. Accordingly, an improvement by using fuzzy least-squares criterion is described in next section.

3. Fuzzy Least-Squares (FLS) Models

Diamond (1988) proposed FLS method to determine fuzzy parameters by adopting concept of minimum fuzziness between observed and estimated values. Along similar lines, a model is constructed in this paper based on following concepts:

(i) Objective function is to minimize sum of squares of difference between estimated regression spread and observed spread of given data.

(ii) Degree of fitness of FLS model, based on each of Possibility and Necessity measures, is greater than or equal to a threshold h, $0 \le h < 1$.

3.1 Fitness of FLS model based on Possibility measure

In fuzzy linear regression model $\hat{\tilde{Y}}_i = \tilde{\boldsymbol{A}}\,\boldsymbol{x}_i$, let $\hat{\tilde{Y}}_i$ and $\tilde{Y}_i$ be estimated and observed data for a vector of independent variables $\boldsymbol{x}_i$, respectively. For $i = 1,...,m$ we define possibility of degree of fitness of estimated $\hat{\tilde{Y}}_i$ for given observed data $\tilde{Y}_i$ as

$$f_i = Pos\left(\tilde{Y}_i = \hat{\tilde{Y}}_i\right) \tag{3.1}$$

The degree of fitness of estimated FLS model to data $X_1, X_2, ..., X_m$ is defined by

$$f = min\,\{\, f_i\,,\; i = 1,2,...,m\} \tag{3.2}$$

A relation for possibility of equality of two fuzzy numbers as obtained by Modarres *et al.* (2005) states that if $\tilde{A} = (\alpha, c)$ and $\tilde{B} = (\beta, d)$, then

$$Pos\left(\tilde{A} = \tilde{B}\right) = L\left(\frac{\alpha - \beta}{c + d}\right) \tag{3.3}$$

By applying Extension principle for fuzzy linear regression model $\hat{\tilde{Y}}_i = \tilde{\boldsymbol{A}}\,\boldsymbol{x}_i$ and for a vector of independent variables $\boldsymbol{x}_i$, the centre and spread of estimated symmetry fuzzy output is $\boldsymbol{\alpha}^t \boldsymbol{x}_i$ and $\boldsymbol{c}^t |\boldsymbol{x}_i|$, respectively. Therefore, membership function of $\hat{\tilde{Y}}_i$ is

$$\hat{\tilde{Y}}_i(\boldsymbol{y}_i) = \begin{cases} L\left\{\dfrac{(y_i - \boldsymbol{\alpha}^t\ \boldsymbol{x}_i\ )}{\boldsymbol{c}^t\ |\boldsymbol{x}_i|}\right\}, & \text{if } \boldsymbol{x}_i \neq 0, \\ 1, & \text{if } \boldsymbol{x}_i = 0,\ \boldsymbol{y}_i = 0, \\ 0, & \text{otherwise}. \end{cases} \tag{3.4}$$

On the other hand, $\tilde{Y}_i = (\boldsymbol{y}_i, \boldsymbol{e}_i)$. By substituting centre and spread of $\hat{\tilde{Y}}_i = \tilde{\boldsymbol{A}}\,\boldsymbol{x}_i$ and $\tilde{Y}_i$ in eq. (3.4), degree of fitness of estimated FLR model, f_i, is calculated as follows:

$$f_i = Pos\left(\hat{\tilde{Y}}_i = \tilde{Y}_i\right) = L\left(\frac{\boldsymbol{\alpha}^t\ \boldsymbol{x}_i - \boldsymbol{y}_i}{\boldsymbol{c}^t\ |\ \boldsymbol{x}_i\ |\ +\ \boldsymbol{e}_i}\right), \qquad \boldsymbol{x}_i \neq 0 \tag{3.5}$$

where

$$\boldsymbol{\alpha}^t = (\alpha_0, \alpha_1, ..., \alpha_n)\ ,\ \boldsymbol{c}^t = (c_0, c_1, ..., c_n) \text{ and } |\boldsymbol{x}_i| = \left(|\boldsymbol{x}_{i0}|, |\boldsymbol{x}_{i1}|, ..., |\boldsymbol{x}_{in}|\right)'$$

The objective function of FLS model is to minimize square of total difference between observed spread, e_i, and estimated spread, $\boldsymbol{c}^t\ |\ \boldsymbol{x}_i\ |$. This can be achieved by minimizing following objective function

$$\text{Minimize } Z\ (h) = \sum_{i=1}^{N}\left(c\ |\ x_i\ | - e_i\right)^2 \tag{3.6}$$

The problem in FLS regression model is to determine fuzzy parameters $\widetilde{A}$ such that $f_i \geq h, \ \forall i, \ i = 1,...,m$. On substituting the value of f_i from eq. (3.2) and solving above inequality constraints of FLS regression model are as follows:

$$\begin{aligned}
&\sum_{j=0}^{n} \alpha_j \ x_{ij} \ + \left| L^{-1}(h) \right| \ \sum_{j=0}^{n} c_j \left| x_{ij} \right| \ \geq \ y_i \ - \left| L^{-1}(h) \right| \ e_i, \\
&\sum_{j=0}^{n} \alpha_j \, x_{ij} \ - \ \left| L^{-1}(h) \right| \ \sum_{j=0}^{n} c_j \left| x_{ij} \right| \ \leq \ y_i \ + \left| L^{-1}(h) \right| \ e_i, \\
&\boldsymbol{c}^t \left| \boldsymbol{x}_i \right| \ \geq \ 0, \qquad\qquad i = 1,...,m
\end{aligned} \tag{3.7}$$

Decision maker selects a threshold $0 \leq h < 1$, as least value for fitness of FLS regression model. Therefore, optimal solution depends on threshold value h. The model involves quadratic programming and can be solved by using any nonlinear optimization solver.

3.2 Fitness of FLS model based on Necessity measure

Let *h*-level set of two fuzzy numbers say $\widetilde{A}$ and $\widetilde{F}$ be $L_h(\widetilde{A})$ and $L_h(\widetilde{F})$ respectively for which degree of its membership function exceeds level *h*:

$$\begin{aligned}
&L_h(\widetilde{A}) = \left\{ u \in R^1 \ / \ \mu_{\widetilde{A}}(u) \geq h \right\} = \left[A_h^L \ , A_h^R \right], \\
&L_h(\widetilde{F}) = \left\{ u \in R^1 \ / \ \mu_{\widetilde{F}}(u) \geq h \right\} = \left[F_h^L \ , F_h^R \right],
\end{aligned} \tag{3.8}$$

where $A_h^L \left(F_h^L \right)$ and $A_h^R \left(F_h^R \right)$ are left and right side extreme points of *h*-level set of $\widetilde{A}(\widetilde{F})$ respectively. As has already been pointed out in Sections 2.1 and 2.2, Possibility is related to intersection and Necessity refers to set inclusion. Using notations for *h*-level sets $L_h(\widetilde{A})$ and $L_h(\widetilde{F})$, and eq. (2.8), following results are obtained:

$$Nes\left(\widetilde{F} \subset \widetilde{A}\right) \geq h \quad \text{iff} \quad A_h^L \leq F_{1-h}^L \text{ and } A_h^R \leq F_{1-h}^R \tag{3.9}$$

Considering fuzzy linear regression model $\widetilde{Y}_i = \widetilde{\boldsymbol{A}}X = \left(\sum_{j=0}^{n} \boldsymbol{a}_j x_{ij}, \sum_{j=0}^{n} \boldsymbol{c}_j \left| x_{ij} \right| \right)$, and using the results obtained in eq. (3.9) for $Nes\left(\widetilde{Y}_i \subset \hat{\widetilde{Y}}_i \right) \geq h$, following inequality is obtained:

$$\begin{aligned} &\sum_{j=0}^{n} \alpha_j x_{ij} - \left| L^{-1}(h) \right| \sum_{j=0}^{n} c_j \left| x_{ij} \right| \leq y_i - \left| L^{-1}(1-h) \right| e_i, \\ &\sum_{j=0}^{n} \alpha_j x_{ij} + \left| L^{-1}(h) \right| \sum_{j=0}^{n} c_j \left| x_{ij} \right| \geq y_i + \left| L^{-1}(1-h) \right| e_i, \end{aligned} \tag{3.10}$$

Now, based on concepts of fuzzy least-squares, objective function (3.6) is minimized with respect to Necessity conditions of (3.10) to yield fuzzy least-square results under Necessity measure.

3.3 Performance evaluation

To determine fuzzy parameters such that estimation error is minimized, following bisection algorithm is suggested.

(i) Set $h = 0,\ h_L = 0$ and $h_U = 1$, where h_L and h_U are lower and upper bound for *h*, respectively.

(ii) Solve (3.6) and denote value of optimal objective function by z^0.

(iii) Set $h=(h_L+h_U)/2$ and solve problem (3.6) again. Denote value of optimal objective function by z^*. Update values of h_L and h_U as $h_L=h$, if $z^*=z^0$ and $h_U=h$, otherwise.

(iv) If difference between two consecutive value of *h* is less than acceptable tolerance, ε, procedure is completed and fuzzy parameters are determined, otherwise go to (iii) above.

In a fuzzy linear regression model, values of response variable are represented as fuzzy numbers with membership functions characterized by explanatory variable. In order to evaluate closeness of observed and estimated fuzzy numbers, support of both fuzzy numbers should be close to each other, where support of a fuzzy set A is defined by $S_A=\{u:\mu_A(u)>0\}$. Therefore, for performance evaluation of a fuzzy regression model, Kim and Bishu (1998) used ratio of difference between membership values to observed membership values as follows:

$$E_i=\frac{\displaystyle\int_{S_{\hat{\tilde{Y}}_i}\cup S_{\tilde{Y}_i}}\left|\,\hat{\tilde{Y}}_i(y)-\tilde{Y}_i(y)\,\right|dy}{\displaystyle\int_{S_{\tilde{Y}_i}}\tilde{Y}_i\;dy} \tag{3.11}$$

where $S_{\hat{\tilde{Y}}_i}$ and $S_{\tilde{Y}_i}$ are the support of $\hat{\tilde{Y}}_i$ and $\tilde{Y}_i$, respectively.

4. Results and Discussion

As an illustration, a part of data given in Sud *et al.* (2006) concerned with yield of Pearl Millet crop at block levels for Bhiwani district of Haryana State is considered here to develop a fuzzy estimate of Pearl Millet crop yield. Nine blocks in the district are: B. Khera, Bhiwani, Kairu, Tosham, Siwani, Loharu, Badhra, Dadri-I and Dadri-II. The number of farmers selected are as follows: B. Khera (16 farmers), Bhiwani (24 farmers), Kairu (18 farmers), Tosham (22 farmers), Siwani (26 farmers), Loharu (26 farmers), Badhra (20 farmers), Dadri-I (24 farmers) and Dadri-II (22 farmers). The explanatory variable, at block level, is farmers' estimate while response variable at the same level is actual Pearl Millet crop yield based on Crop-cutting experiments, and are fuzzy numbers. Entire data analysis is carried out using LINGO, Version 8, software package (LINDO, 2002) available at I.A.S.R.I., New Delhi. The data for present investigation, culled from Sud *et al*. (2006), is reproduced in Table 1 for ready reference.

Yield as function of farmers' estimates can be expressed as

$$\hat{\tilde{Y}}_i = (\alpha_1 , c_1) + (\alpha_2 , c_2)\, x_i \qquad i = 1,\ldots,9 \qquad (4.1)$$

Table 1. Pearl Millet crop yield (based on CCE) as triangular fuzzy numbers with farmers' estimates for Bhiwani district

Block Number	*Blocks*	*Farmers' Estimate (quintals/hectare)*	*Lower limit of yield (quintals /hectare)*	*Upper limit of yield (quintals /hectare)*
1	B. Khera	13.36	10.00	15.00
2	Bhiwani	19.69	12.50	20.00
3	Kairu	10.01	6.00	12.42
4	Tosham	10.66	5.00	10.80
5	Siwani	9.98	6.25	12.01
6	Loharu	11.93	9.09	14.51
7	Badhra	11.96	7.33	15.01

8	Dadri-I	10.08	8.75	13.75
9	Dadri-II	9.75	11.43	15.01

4.1 Possibility approach

When $Pos\left(\hat{\tilde{Y}}_i = \tilde{Y}_i\right)$, fuzzy linear regression model with least-squares error can be formulated with following objective function to be minimized:

$$Min\ Z(h) = \{(\alpha_1 + \alpha_2) - 2.50\}^2 + \{(\alpha_1 + \alpha_2) - 3.75\}^2 + \ldots + \{(\alpha_1 + \alpha_2) - 1.79\}^2 \tag{4.2}$$

subject to

$$\left.\begin{aligned} 12.50 + 2.50\left|L^{-1}(h)\right| &\geq (\alpha_1 * 1 + \alpha_2 * 13.36) - (c_1 * 1 + c_2 * 13.36)\left|L^{-1}(h)\right| \\ 12.50 - 2.50\left|L^{-1}(h)\right| &\leq (\alpha_1 * 1 + \alpha_2 * 13.36) + (c_1 * 1 + c_2 * 13.36)\left|L^{-1}(h)\right| \end{aligned}\right. ,$$

$$\vdots$$

$$\left.\begin{aligned} 13.22 + 1.79\left|L^{-1}(h)\right| &\geq (\alpha_1 * 1 + \alpha_2 * 9.75) - (c_1 * 1 + c_2 * 9.75)\left|L^{-1}(h)\right| \\ 13.22 - 1.79\left|L^{-1}(h)\right| &\leq (\alpha_1 * 1 + \alpha_2 * 9.75) + (c_1 * 1 + c_2 * 9.75)\left|L^{-1}(h)\right| \end{aligned}\right. ,$$

$$c_1 * 1 + c_2 * 13.36 \geq 0, \ldots, c_1 * 1 + c_2 * 9.75 \geq 0 \tag{4.3}$$

The above nonlinear quadratic optimization problem is solved to obtain FLS regression model. The optimal value of fitness level is obtained using bisection algorithm as discussed in Section 3.3. A computer program is written in LINGO and objective function value z^0 is obtained by taking $h = 0$. Then, the value of h is updated according to bisection algorithm to obtain subsequent values of objective function. At last iteration, optimal value of fitness level h is determined. The optimum value of fitness level at tolerance level, $\varepsilon = 0.001$, is computed as h = 0.451. Using the above fitness level and solving quadratic optimization problem, model constructed is as follows:

$$\hat{\tilde{Y}}_i = (9.66, 1.60) + (0.13, 0.11)x_i \qquad i = 1,2,\ldots,9 \tag{4.4}$$

4.2 Necessity approach

For $Nes\left(\tilde{Y}_i \subset \hat{\tilde{Y}}_i\right)$, fuzzy linear regression model with least-squares error can be formulated with following objective function to be minimized:

$$Min\ Z(h) = \{(\alpha_1 + \alpha_2) - 2.50\}^2 + \{(\alpha_1 + \alpha_2) - 3.75\}^2 + \ldots + \{(\alpha_1 + \alpha_2) - 1.79\}^2 \tag{4.5}$$

subject to

$$\begin{array}{l} 12.50 - 2.50\left|L^{-1}(1-h)\right| \geq (\alpha_1 * 1 + \alpha_2 * 13.36) - (c_1 * 1 + c_2 * 13.36)\left|L^{-1}(h)\right| \\ 12.50 + 2.50\left|L^{-1}(1-h)\right| \leq (\alpha_1 * 1 + \alpha_2 * 13.36) + (c_1 * 1 + c_2 * 13.36)\left|L^{-1}(h)\right| \end{array},$$

$$\vdots$$

$$\begin{array}{l} 13.22 - 1.79\left|L^{-1}(1-h)\right| \geq (\alpha_1 * 1 + \alpha_2 * 9.75) - (c_1 * 1 + c_2 * 9.75)\left|L^{-1}(h)\right| \\ 13.22 + 1.79\left|L^{-1}(1-h)\right| \leq (\alpha_1 * 1 + \alpha_2 * 9.75) + (c_1 * 1 + c_2 * 9.75)\left|L^{-1}(h)\right| \end{array},$$

$$c_1 * 1 + c_2 * 13.36 \geq 0, \ldots, c_1 * 1 + c_2 * 9.75 \geq 0 \tag{4.6}$$

The above problem is solved in a similar manner as above and optimal value for h is computed as *0.003* at tolerance level of $\varepsilon = 0.001$. The model constructed is as follows:

$$\hat{\tilde{Y}}_i = (8.53\ ,\ 1.73) + (0.20\ ,\ 0.10)x_i\ , \quad i = 1,2,\ldots,9 \tag{4.7}$$

4.3 Minimization approach

The problem is formulated as follows:

$$Min\ \ J(h) = \sum_{i=1}^{9} (c_1 * 1 + c_2 * x_i), \qquad i = 1,2,\ldots,9 \tag{4.8}$$

subject to

$$\begin{array}{l} 12.50\ +\ 2.50\ \left|L^{-1}(h)\right|\ \le (\alpha_1 * 1 + \alpha_2 * 13.36) +\ (c_1 * 1 + c_2 * 13.36)\left|L^{-1}(h)\right| \\ 12.50\ -\ 2.50\ \left|L^{-1}(h)\right|\ \ge (\alpha_1 * 1 + \alpha_2 * 13.36) -\ (c_1 * 1 + c_2 * 13.36)\left|L^{-1}(h)\right| \end{array},$$

$$\begin{array}{l} . \\ . \\ . \end{array}$$

$$\begin{array}{l} 13.22\ +\ 1.79\ \left|L^{-1}(h)\right|\ \le\ (\alpha_1 * 1 + \alpha_2 * 9.75) + (c_1 * 1 + c_2 * 9.75)\ \left|L^{-1}(h)\right| \\ 13.22\ -\ 1.79\ \left|L^{-1}(h)\right|\ \ge\ (\alpha_1 * 1 + \alpha_2 * 9.75) -\ (c_1 * 1 + c_2 * 9.75)\left|L^{-1}(h)\right| \end{array},$$

$$c_1\ ,\ c_2 \ge 0 \tag{4.9}$$

As discussed on Page 56, the optimum value of h is obtained as 0.451. Therefore, solving above linear programming problem (4.9) for *h = 0.451,* fuzzy linear regression model constructed is as follows:

$$\hat{\tilde{Y}}_i = (\ 6.04\ ,\ 7.53\ ) + (0.41\ ,\ 0\ )x_i \qquad i = 1,2,\ldots,9 \tag{4.10}$$

Substituting values of farmers' estimates as given in Table 1 in eqs. (4.4), (4.7) and (4.10), estimated fuzzy Pearl Millet crop yield corresponding to Possibility, Necessity, and Minimization methods are computed and reported in Table 2. Further, using eq. (3.12), errors in estimation E_i for optimal fitness level are computed for all the three approaches and the same are reported in Table 3.

Table 2. Estimated fuzzy yields

Blocks	Estimated Yields								
	Possibility			Necessity			Minimization		
	$\hat{Y}_{il}$	$\hat{Y}_{im}$	$\hat{Y}_{iu}$	$\hat{Y}_{il}$	$\hat{Y}_{im}$	$\hat{Y}_{iu}$	$\hat{Y}_{il}$	$\hat{Y}_{im}$	$\hat{Y}_{iu}$
B. Khera	8.29	11.34	14.40	8.14	11.20	14.27	4.02	11.56	19.09
Bhiwani	8.40	12.14	15.88	8.77	12.47	16.17	6.64	14.17	21.71
Kairu	8.23	10.92	13.61	7.80	10.53	13.26	2.64	10.17	17.71
Tosham	8.25	11.00	13.76	7.87	10.66	13.46	2.91	10.44	17.98
Siwani	8.23	10.92	13.60	7.80	10.53	13.26	2.63	10.16	17.70
Loharu	8.27	11.16	14.06	7.99	10.92	13.84	3.43	10.97	18.50
Badhra	8.27	11.17	14.07	8.00	10.92	13.85	3.44	10.98	18.52
Dadri-I	8.24	10.93	13.63	7.81	10.55	13.28	2.67	10.20	17.74
Dadri-II	8.23	10.89	13.55	7.78	10.48	13.19	2.53	10.07	17.60

Table 3. Errors in estimation

Blocks	Errors in estimation of crop yield		
	Possibility	Necessity	Minimization
B. Khera	0.83	1.76	2.01
Bhiwani	1.59	1.81	1.01
Kairu	0.91	0.73	1.35
Tosham	1.56	1.45	1.60
Siwani	1.05	0.85	1.62
Loharu	0.44	1.51	1.79
Badhra	0.24	0.25	0.97
Dadri-I	0.25	1.51	2.01
Dadri-II	1.92	2.41	3.22
Total	**8.79**	**12.28**	**15.84**

A perusal shows that values of sums of errors for Possibility and Necessity methods are lower than those for Minimization method for all the blocks. In other words, least-squares approach gives more reliable estimates for crop yield vis-à-vis linear programming approach. Further, Possibility approach is found to be superior to Necessity approach for data under consideration. To get a visual idea, observed and estimated Pearl Millet crop yields obtained from using Possibility approach with h=*0.451,* are depicted in Figure 1. Evidently, observed yields (in solid lines) and estimated yields (in dotted lines) are found to be quite close to each other, thereby indicating that farmers' estimates are able to explain actual crop yield with high fitness levels.

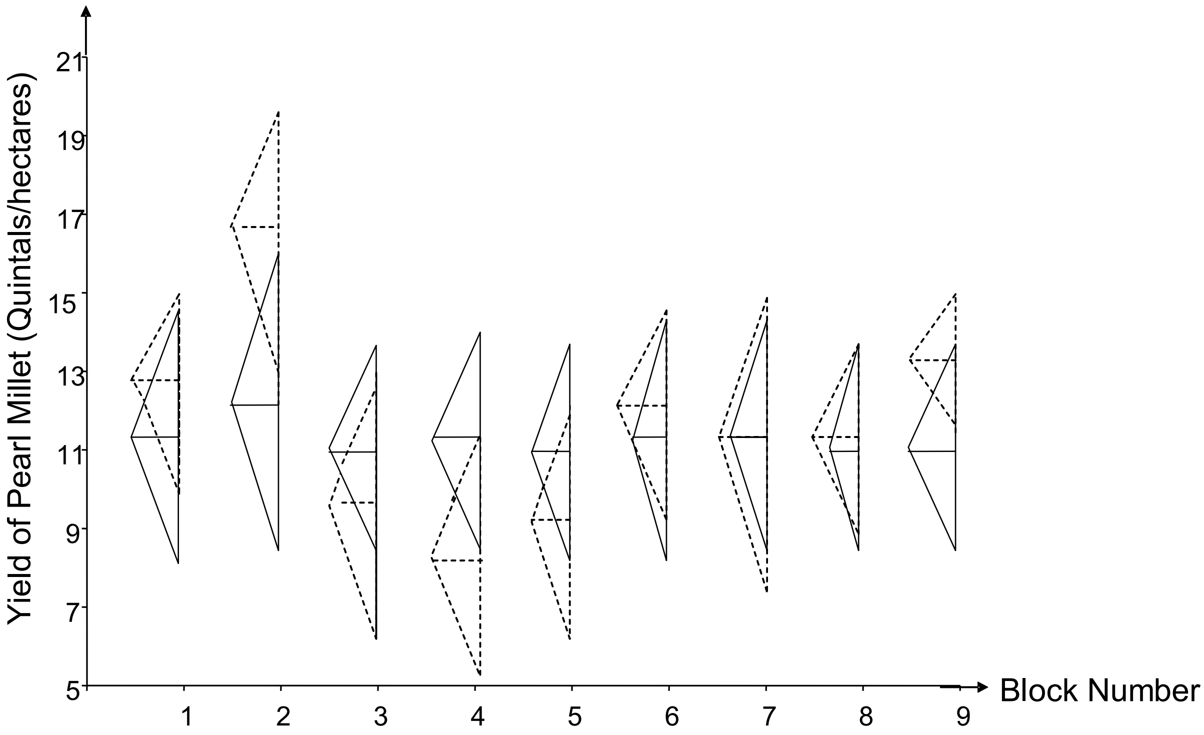


Figure 1: Observed (in solid lines) and estimated (in doted lines) Pearl Millet crop yield based on Possibility approach at optimal fitness level

CHAPTER IV

A MODIFIED FUZZY LEAST-SQUARES APPROACH FOR FUZZY LINEAR REGRESSION ANALYSIS

---

1. Background

Multiple linear regression model is widely employed to study relationship between response and explanatory variables. The underlying phenomenon as well as both the above mentioned variables are generally assumed to be "crisp". However, in reality, there is always some amount of "impreciseness" or "vagueness" or "fuzziness" in one or more of the above. Therefore, for a more realistic modelling, it is desirable to study "Fuzzy" versions of linear regression models.

Broadly two approaches exist in the literature for studying a Fuzzy linear regression (FLR) model. In the former, proposed by Tanaka *et al.* (1982), parameters of FLR model:

$$Y = A_0 + A_1 X_1 + \ldots + A_m X_m \tag{1.1}$$

where

$$A_i = (a_{ic}, a_{iw}), \qquad Y = (y_c, y_w),$$

subscripts "*c*" and *"w"* denoting respectively centres and widths, are estimated by minimizing "Total vagueness" of model-data combination, subject to constraints that each data point must lie within estimated value of response variable. This can be visualized as a Linear programming (LP) problem and solved by using "Simplex procedure". To this end, several software packages, like SAS, LP88, and LINDO are available. However, any standard spreadsheet package, like Microsoft Excel may also be used to solve LP problem manually.

Kandala and Prajneshu (2003) demonstrated applicability of above methodology when the two explanatory variables, viz. Plant height and Leaf area index and response variable, viz. Dry-matter accumulation are all crisp but underlying phenomenon is assumed to be fuzzy in nature. It was shown that widths of prediction intervals in respect of Fuzzy linear regression model were much less than those for Multiple linear regression model. Kandala and Prajneshu (2002) had earlier obtained similar results in the situation when the two explanatory variables, viz. Normalized difference vegetation index and Ratio vegetation index are highly correlated.

Further, for determining age-length relationship in a fish species, response variable (length) generally lies in an interval for different fish of same age. Kandala and Prajneshu (2004a) applied FLR methodology for fitting fuzzy von Bertalanffy growth model with a view to determining age-length relationship in pearl oyster. It may be pointed out that traditional statistical methods are not capable of handling such a situation in which response variable is in intervals. The only way out there is to get rid of interval values for response variable by converting these to crisp values either by taking mean or mode, thereby losing a lot of vital information about spread. Further, uncertainties intrinsic to data can not be taken care off. However, a criticism of Tanaka's approach is that it is not based on sound statistical principles. Another drawback, as pointed out by Chang and Ayyub (2001), is that as the number of data points increases, the number of constraints in LP increases proportionally, thereby resulting in computational difficulties.

The second approach based on Fuzzy least squares (FLS) method, was pioneered by Diamond (1988), which as its name suggests, is a fuzzy extension of Least squares method based on a new defined distance on the space of fuzzy numbers. Kandala and Prajneshu (2004b) have applied this methodology for fitting well-known "Allometric model" to length-weight data of some fish species. Recently, Singh *et al.* (2007) have thoroughly studied Possibility and Necessity approaches for obtaining reliable fuzzy estimates using "Fuzzy least-squares" procedure. It was shown that, for data under consideration, "Possibility" measure is better than "Necessity" measure.

D'Urso (2003) initiated a new approach based on Modified fuzzy least-squares approach, to deal with Fuzzy linear regression analysis. A doubly linear adaptive fuzzy regression model was proposed based on two linear models: (i) Core regression model and (ii) Spread (width) regression model. First one explains “centres” of fuzzy observations, while second one is for their “spreads”. In this work, this approach is followed. Further, performance evaluation criterion based on “Difference in membership functions” is adopted for computation of error in estimation (Kim and Bishu, 1998). As an illustration, the methodology is applied for crop yield estimation of Pearl Millet at block level using data of Sud *et al.* (2006). Superiority of present approach over Possibility fuzzy least-squares approach is demonstrated for data under consideration.

2. Modified Fuzzy Least Squares Method

Suppose we wish to study relationship between a set of crisp explanatory variables, X_i, *i* =*1*, *2*, . . . , *m* and a fuzzy response variable, $\widetilde{Y}$ = *(y, e)*, where *y* is centre, and *e* is spread (width). Following along similar lines as D’Urso (2003), a modified fuzzy regression model (MFRM), in terms of a doubly linear adaptive fuzzy regression model, i.e. a core and spread regression model, is represented as follows:

$$y = y^* + \boldsymbol{\varepsilon}_y, \qquad y^* = \boldsymbol{X}\,\boldsymbol{a} \tag{2.1}$$

$$e = e^* + \boldsymbol{\varepsilon}_e, \qquad e^* = y^* b + \mathbf{1}\, d \tag{2.2}$$

where $\boldsymbol{X}$ is a $n \times (m+1)$ matrix containing input variables (data matrix), $\boldsymbol{a}$ is a column $(m+1)$ vector containing parameters of the first regression model, y and y^* are respectively vectors of observed and interpolated centres, both having dimension $n \times 1$, e and e^* are respectively vectors of assigned and interpolated spreads, both having dimension $n \times 1$, **1** is a $(n \times 1)$ vector of all 1's, b and d are parameters for second regression model, and $\boldsymbol{\varepsilon}_y$ and $\boldsymbol{\varepsilon}_e$ are error terms of the two regression models respectively assumed to follow independent standard normal distributions.

Observe that predictive variables ***X*** are taken into consideration in (2.2) through observed centres, in fact it can be written explicitly as $e = \boldsymbol{X}\,\boldsymbol{a}\,b + \mathbf{1}\,d + \boldsymbol{\varepsilon}_e$. The model is hence capable to take into account possible linear relations between size of spreads and magnitude of estimated centres. Euclidean distance between two symmetrical fuzzy numbers $Y_i = (\,y_i,\, e_i\,)$ and $Y_i^* = \left(\,y_i^*,\, e_i^*\,\right)$ is defined as follows:

$$\delta_i = \delta\left(\,Y_i,\, Y_i^*\,\right) = \sqrt{\left(y_i - y_i^*\,\right)^2 + \left(e_i - e_i^*\,\right)^2} \tag{2.3}$$

Drawing analogy from Linear regression analysis (based on crisp data), parameters here are estimated by minimizing following sum of squared errors:

$$\varphi(\boldsymbol{a}, b, d) = \sum_{i=1}^{n} \delta_i^2 = \left(y - y^*\right)^T \left(y - y^*\right) + \left(e - e^*\right)^T \left(e - e^*\right) \tag{2.4}$$

where superscript "*T*" denotes transpose. Equating to zeros partial derivatives with respect to (vector) parameters ***a***, *b*, *d* and solving resultant equations, recursive solution for problem of least-squares estimation with fuzzy data is given by

$$\boldsymbol{a} = \left(1 + b^2\right)^{-1} \left[\left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \left(y + eb - \mathbf{1}bd\right)\right],$$

$$b = \left(\boldsymbol{a}^T \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{a}\right)^{-1}\left(e^T \boldsymbol{X}\boldsymbol{a} - \boldsymbol{a}^T \boldsymbol{X}^T \mathbf{1} d\right),$$

$$d = n^{-1}\left(e^T \mathbf{1} - \boldsymbol{a}^T \boldsymbol{X}^T \mathbf{1} b\right) \tag{2.5}$$

Relevant computer program for solving above equations recursively is written in SAS/IML software package and appended as Annexure-I at the end of this chapter.

3. Performance Evaluation

Performance evaluation of a fuzzy regression model is based on closeness of observed and estimated fuzzy numbers, such that support of both are close to each other, where support of a fuzzy set A is defined by $S_A = \{x : \mu_A(x) > 0\}$. Kim and Bishu (1998) used ratio of difference between membership values to observed membership values for performance evaluation as follows:

$$E_i = \frac{\displaystyle\int_{S_{\hat{\tilde{Y}}_i} \cup S_{\tilde{Y}_i}} \left| \hat{\tilde{Y}}_i(y) - \tilde{Y}_i(y) \right| dy}{\displaystyle\int_{S_{\tilde{Y}_i}} \tilde{Y}_i \, dy} = \frac{N_i}{D_i}, \text{say} \tag{3.1}$$

where $S_{\hat{\tilde{Y}}_i}$ and $S_{\tilde{Y}_i}$ are the support of $\hat{\tilde{Y}}_i$ and $\tilde{Y}_i$, respectively. It can easily be seen that D_i in (3.1) reduces to spread e_i in the case of observed triangular fuzzy numbers. The term N_i in equation (3.1) represents difference of observed and estimated fuzzy responses. Smaller values of N_i indicate that fuzzy regression model fits the data better. Also, various types of overlap between observed and estimated fuzzy values are possible. Expression for N_i may be evaluated by noticing the fact that N_i is equal to sum of areas of observed and estimated responses subtracting twice area of intersection (Kao and Chyu, 2002). One possibility of overlap between observed and estimated fuzzy value is depicted in Figure 1.

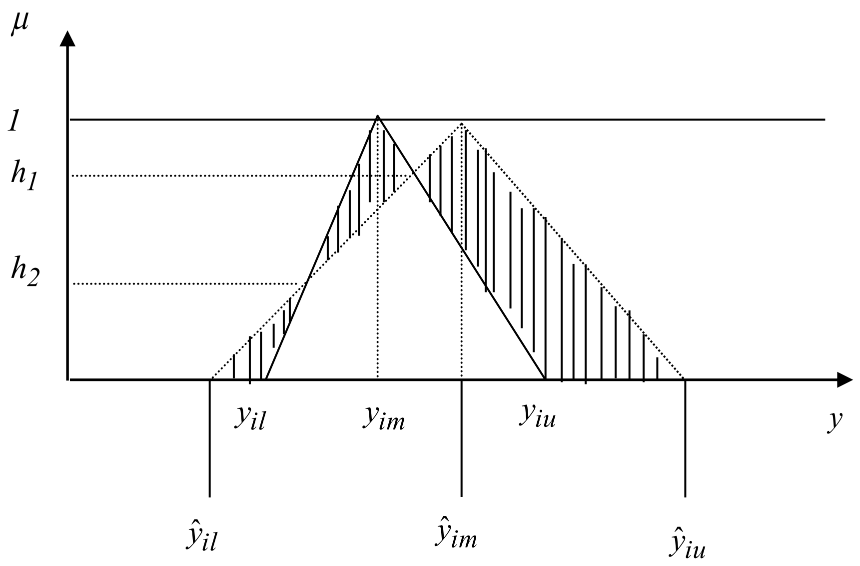


Figure 1: Difference of membership functions between observed and estimated responses represented as shaded region

The value of N_i, which represents shaded area in above figure, is calculated as

$$N_i = 0.5\left[y_{iu} - y_{il}\right] + 0.5\left[\hat{y}_{iu} - \hat{y}_{il}\right] - 2\,(0.5)\,\left[y_{iu} - \hat{y}_{il}\right]\,h_1 + 2(0.5)\left[y_{il} - \hat{y}_{il}\right]\,h_2$$

where $h_i = (h_1, h_2)$ is membership value of intersection of right shape function of $\tilde{Y}_i$ and left shape function $\hat{\tilde{Y}}_i$. In symbols, we have

$$h_i = height\left(\tilde{Y}_i \cap \hat{\tilde{Y}}_i\right), \tag{3.2}$$

where *"height"* of a fuzzy set A is value of x such that $\mu_A(x) = \underset{y \in X}{Max}\, \mu_A(y)$.

Algebraically, $h_i = (h_1, h_2)$ can be expressed as

$$h_1 = \frac{(y_{iu} - \hat{y}_{il})}{(y_{iu} - y_{im} + \hat{y}_{im} - \hat{y}_{il})}, \qquad h_2 = \frac{(y_{il} - \hat{y}_{il})}{(y_{il} - y_{im} + \hat{y}_{im} - \hat{y}_{il})} \tag{3.3}$$

The algebra for other possibilities follows along similar lines but is omitted here to save space.

4. An Illustration

Entire data analysis is carried out using Statistical Analysis System, Version 9.1.3, software package (SAS, 2006) available at I.A.S.R.I., New Delhi. Sud *et al.* (2006) carried out a detailed study for developing yield estimates of Pearl Millet crop at block levels for Bhiwani district of Haryana. Nine blocks in Bhiwani district are: B. Khera, Bhiwani, Kairu, Tosham, Siwani, Loharu, Badhra, Dadri-I and Dadri-II. The number of farmer selected is as follows: B. Khera (16 farmers), Bhiwani (24 farmers), Kairu (18 farmers), Tosham (22 farmers), Siwani (26 farmers), Loharu (26 farmers), Badhra (20 farmers), Dadri-I (24 farmers) and Dadri-II (22 farmers). The explanatory variable, at block level, is farmers' estimate while response variable at the same level is actual Pearl Millet crop yield based on Crop-cutting experiments, and are fuzzy numbers. The data for present investigation, culled from Sud *et al*. (2006), is reproduced in Table 1 for ready reference. Parameter estimates of MFRM model are computed as follows:

$$\boldsymbol{a} = (4.09,\ 0.61)^T, \quad b = 0.18, \quad d = 0.87$$

Further, estimated centres and spreads are reported in Table 2. The numbers of iteration required for parameter estimates are *521* with objective function value as *23.14*. Estimated Pearl Millet crop yields along with errors in estimation at block levels in respect of Possibilistic least-squares model (PLSM) and Modified fuzzy regression model (MFRM) approaches are reported in Table 3.

Table 1. Pearl Millet crop yield (based on CCE) as triangular fuzzy numbers with farmers' estimates for Bhiwani district

Block Number	*Blocks*	*Farmers' Estimate (quintals/hectare)*	*Lower limit of yield (quintals /hectare)*	*Upper limit of yield (quintals /hectare)*
1	B. Khera	13.36	10.00	15.00
2	Bhiwani	19.69	12.50	20.00
3	Kairu	10.01	6.00	12.42
4	Tosham	10.66	5.00	10.80
5	Siwani	9.98	6.25	12.01
6	Loharu	11.93	9.09	14.51
7	Badhra	11.96	7.33	15.01
8	Dadri-I	10.08	8.75	13.75
9	Dadri-II	9.75	11.43	15.01

Table 2. Estimated centres and spreads

Block Nos.	1	2	3	4	5	6	7	8	9
y_i^*	12.25	16.11	10.20	10.60	10.18	11.37	11.39	10.21	10.04
e_i^*	3.05	3.74	2.69	2.76	2.68	2.89	2.90	2.69	2.65

Table 3. Estimated crop yields along with errors in estimation

Blocks	Estimated crop yields						Errors in estimation	
	PLSM			MFRM			PLSM	MFRM
	$\hat{Y}_{il}$	$\hat{Y}_{im}$	$\hat{Y}_{iu}$	$\hat{Y}_{il}$	$\hat{Y}_{im}$	$\hat{Y}_{iu}$		
B. Khera	8.29	11.34	14.40	9.20	12.25	15.30	0.83	0.26
Bhiwani	8.40	12.14	15.88	12.37	16.11	19.85	1.59	0.07

Kairu	8.23	10.92	13.61	7.51	10.20	12.89	0.91	0.57
Tosham	8.25	11.00	13.76	7.84	10.60	13.36	1.56	1.42
Siwani	8.23	10.92	13.60	7.50	10.18	12.86	1.05	0.66
Loharu	8.27	11.16	14.06	8.48	11.37	14.26	0.44	0.31
Badhra	8.27	11.17	14.07	8.49	11.39	14.29	0.24	0.26
Dadri-I	8.24	10.93	13.63	7.52	10.21	12.90	0.25	0.75
Dadri-II	8.23	10.89	13.55	7.39	10.04	12.69	1.92	2.28

Total sums of errors in estimation for MFRM method, viz. 6.58, is found to be lower than that for PLS method, which is 8.79. In other words, MFRM approach gives more reliable estimates vis-à-vis PLS approach for data under consideration. To get a visual idea, observed and estimated Pearl Millet crop yields, obtained through MFRM approach, are depicted in Figure 2. Evidently, observed yields (in solid lines) and estimated yields (in dotted lines) are found to be quite close to each other, thereby indicating that farmers' estimates are able to explain actual Pearl Millet crop yields at Block level. It is hoped that this type of work would go a long way in obtaining crop yield estimates at small area level, which is of utmost importance for micro-level planning.

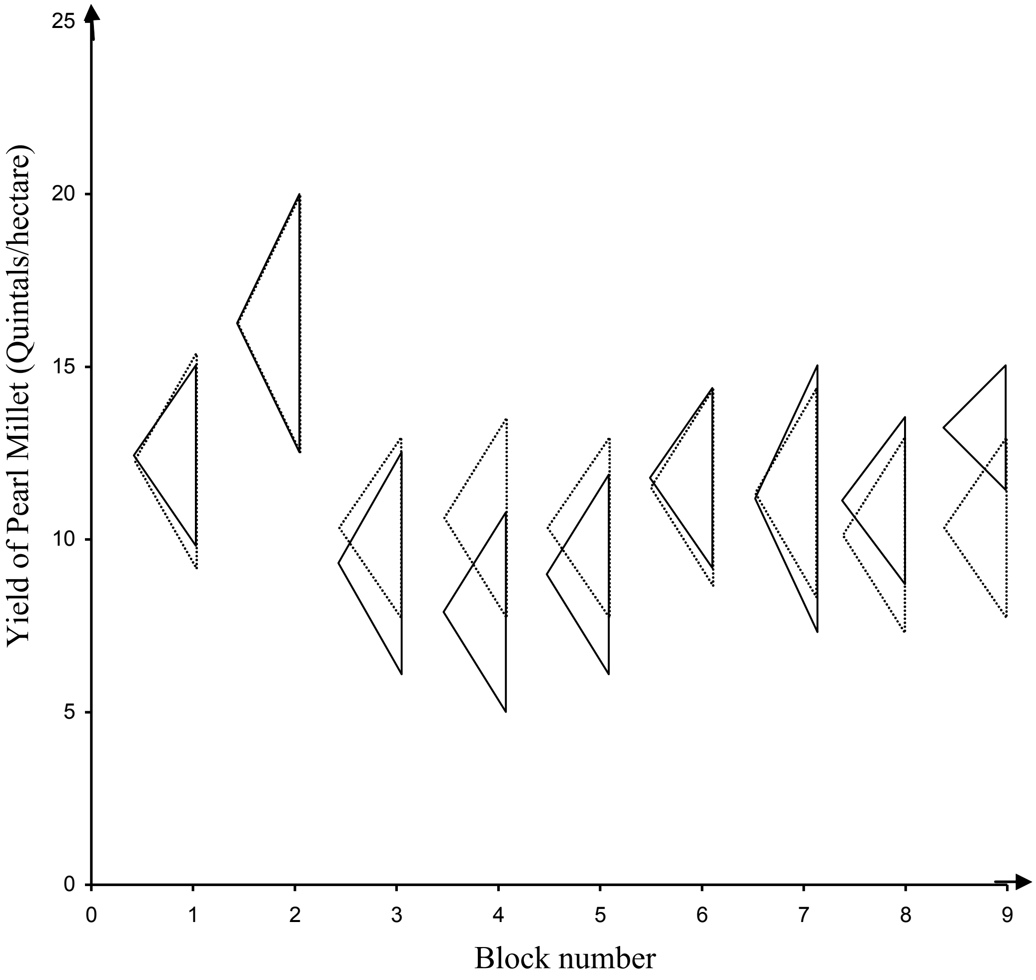


Figure 2: Observed (in solid lines) and estimated (in dotted lines) Pearl Millet crop yields at block level based on MFRM approach

ANNEXURE – I

```
/* Program for Bajra crop of Bhiwani district Haryana*/
proc iml;
start line(coeff, intercept, minX, maxX, step);
x = do(minX, maxX, step); y = coeff # x + intercept;
finish;
do;
        /* Input data */
        y = {12.50,16.25,9.21,7.90,9.13,11.80,11.17,11.25,13.22};
        ei = {2.50,3.75,3.21,2.90,2.88,2.71,3.84,2.50,1.79};
        Data = {13.36,19.69,10.01,10.66,9.98,11.93,11.96,10.08,9.75};

                /* Modified fuzzy Regression analysis program */
                Num = nrow(Data);  Independent = ncol(Data); One = J(Num, 1);
                X = One || Data; N = X` * X; M = inv(N); P = X` * ei;

                    /* Modified fuzzy regression model: iterative solution */
                    threshold = 1e-8;
                    IterNum = 0;
                    a = (1:(Independent+1))`; * arbitrary starting values;
                    b = 1; * arbitrary starting values;
                    d = 1; * arbitrary starting values;
                    do until (abs((a - aOld) // (b - bOld) // (d - dOld))< threshold);
                            IterNum = IterNum + 1;
                            aOld = a; bOld = b; dOld = d;
                            a = M * (X` * y + P # b - X` [, +] # b # d) / (1 + b ## 2);
                              b = (P` * a - a` * X` [, +] # d) / (a` * N * a);
                              d = (ei[+] - a` * X` [, +] # b) / Num;
                      end;
                                yStar = X * a;
                                eiStar = yStar # b + One # d;
                                Ey = (y - yStar);
                                Eei = (ei - eiStar);
                                Of = Ey[##] + Eei[##];

                 /* Output */
                   print
                 "Modified fuzzy regression model: Iterative solution",,
                 "Data", Data,, "a:" a,, "b:" b, "d:" d, "y:" y, "yStar:" yStar,,
                  "Sum Ey" (Ey[+]),, "ei:" ei "Sum Eei" (Eei[+]),, "eiStar:" eiStar,,
                  "Objective function:" Of,, "Number of iterations:" IterNum;
end;
quit;
```

CHAPTER V

A TWO-STAGE FUZZY LEAST SQUARES PROCEDURE FOR FITTING VON BERTALANFFY GROWTH MODEL

---

1. Background

von Bertalanffy growth model is generally employed for determining age-length relationship in fisheries (Quinn and Deriso, 1999). The usual procedure is to convert the original nonlinear model to a linear model by expressing length at time-epoch (t+1), i.e. L_{t+1} as a function of L_t, thereby arriving at a Multiple linear regression (MLR) model. Then, method of least squares is used for estimation of parameters. In this procedure, response variable (length) is assumed as precise or crisp. However, data collected generally comprise length measurements for a number of even-aged fish at various time epochs. For each fixed time point, a random sample of individuals is selected, and measurements are recorded. So, in reality, for each time epoch, length of a fish is not described by just one number but by a set of observations lying in an interval. In other words, the response variable is fuzzy. To handle such a situation, it is imperative to employ “Fuzzy linear regression (FLR)” methodology (Kacprzyk and Fedrizzi, 1992). In traditional MLR methodology, the situation of interval response variable is handled by taking mean-values, thereby losing a lot of vital information.

There are basically two approaches to deal with FLR models. The first one, proposed by Tanaka et al. (1982), is based on Linear programming (LP) approach. Here, parameters of FLR model:

$$Y = A_1 X_1 + \ldots + A_p X_p \tag{1.1}$$

where

$$A_i = (a_{ic}, a_{iw}), \qquad Y = (y_c, y_w)$$

are estimated by minimizing “Total vagueness” of model-data combination, subject to constraints that each data point must lie within estimated value of response variable. This can be visualized as an LP problem and solved by using “Simplex procedure”. Kandala and Prajneshu (2004) applied this methodology for fitting fuzzy von Bertalanffy growth model with a view to determining age-length relationship in pearl oyster.

The second approach based on Fuzzy least squares (FLS) methods, was pioneered by Diamond (1988), which as its name suggests, is a fuzzy extension of Least squares method based on a new defined distance on the space of fuzzy numbers. A drawback of this procedure is that the spread of estimated responses increases as magnitude of explanatory variable increases, even though the spread of observed responses are roughly constant or decreasing. To overcome this, Kao and Chyu (2002) proposed a “two-stage” approach for fitting FLR model through FLS approach and showed its superiority over Diamond’s procedure.

In this work, our purpose is to study Kao and Chyu's (2002) approach thoroughly and then, as an illustration, apply it to some fisheries data.

2. Materials and Methods

Fuzzy von Bertalanffy growth model

In this model, length of fish at time *t*, viz. L_t is expressed as

$$L_t = L_\infty \left[1 - \exp\{-k(t - t_o)\}\right] \tag{2.1}$$

where L_∞ , k and t_o are parameters of the model. Parameter L_∞ is interpreted as maximum mean length and *k* is a curvature parameter that determines how fast a fish approaches L_∞. Devaraj (1983) pointed out that the simplest method of fitting eq. (2.1) with additive error term is Bagenal's method. In this method, this equation is rewritten as

$$L_{t+1} = a + bL_t + \varepsilon \tag{2.2}$$

where

$$a = L_\infty(1 - e^{-k}) \text{ and } b = e^{-k}$$

Without loss of generality, assume all observations are fuzzy numbers, since crisp values can be represented by degenerated fuzzy numbers. Eq. (2.2) can be rewritten as

$$\widetilde{L}_{t+1} = a + b\widetilde{L}_t + \widetilde{\varepsilon}_t \qquad t=1,2,\ldots,n \tag{2.3}$$

where $\widetilde{L}_{t+1}$, $\widetilde{L}_t$, and $\widetilde{\varepsilon}_t$ are fuzzy numbers with membership functions $\mu_{\widetilde{L}_{t+1}}$, $\mu_{\widetilde{L}_t}$, $\mu_{\widetilde{\varepsilon}_t}$ respectively. The problem is to obtain estimates for a, b and $\widetilde{\varepsilon}_t$ which provides best explanation for the relationship between explanatory and response variables.

Several types of membership functions, like Triangular, Trapezoidal, and Gaussian have been developed in the literature (Klir and Yuan, 2000). From computational point of view, triangular membership functions are easiest to handle and, hence, these are most commonly used. In this paper, we shall confine ourselves to such membership functions. Accordingly, observations are assumed to be in the form of triangular fuzzy numbers defined as

$$\widetilde{L}_t = [X_{tl}, X_{tm}, X_{tu}] \text{ and } \widetilde{L}_{t+1} = [Y_{tl}, Y_{tm}, Y_{tu}] \tag{2.4}$$

with membership functions:

$$\mu_{\widetilde{L}_t}(x) = \begin{cases} (x - X_{tl})/(X_{tm} - X_{tl}), & \text{if } X_{tl} \le x \le X_{tm} \\ (X_{tu} - x)/(X_{tu} - X_{tm}), & \text{if } X_{tm} \le x \le X_{tu} \end{cases}$$

$$\mu_{\widetilde{L}_t}(y) = \begin{cases} (y - Y_{tl})/(Y_{tm} - Y_{tl}), & \text{if } Y_{tl} \le y \le Y_{tm} \\ (Y_{tu} - y)/(Y_{tu} - Y_{tm}), & \text{if } Y_{tm} \le y \le Y_{tu} \end{cases}$$

3. Kao and Chyu's Two–Stage Approach

Kao and Chyu (2002) proposed two–stage approach where, in first stage, defuzzification of the fuzzy observations to crisp values is carried out and classical least squares method is applied to estimate the regression coefficient. In second stage, fuzzy error form is determined via a mathematical program by minimizing the errors in estimation. In order to estimate parameters *a* and *b* appearing in eq. (2.2), fuzzy observations $\widetilde{L}_{t+1}$, and $\widetilde{L}_t$ are defuzzified to crisp values and conventional least squares employed. There are several defuzzification methods, like Centroid or Centre of gravity, Centre of sums, Middle of maxima, and Height methods available in literature (Piegat, 2001). We consider here Centroid method of defuzzification because only in this method. all membership functions take part in defuzzification process.

Let $L_{(t+1)c}$ and L_{tc} be defuzzified values of $\widetilde{L}_{t+1}$ and $\widetilde{L}_t$ respectively. Then

$$L_{(t+1)c} = \int_{-\infty}^{+\infty} y\, \mu_{\widetilde{L}_{t+1}}(y)dy \Bigg/ \int_{-\infty}^{+\infty} \mu_{\widetilde{L}_{t+1}}(y)dy = [Y_{tl} + Y_{tm} + Y_{tu}]/3$$

$$L_{tc} = \int_{-\infty}^{+\infty} x\, \mu_{\widetilde{L}_t}(x)dx \Bigg/ \int_{-\infty}^{+\infty} \mu_{\widetilde{L}_t}(x)dx = [X_{tl} + X_{tm} + X_{tu}]/3 \qquad (3.1)$$

Based on crisp values of $L_{(t+1)c}$ and L_{tc}, the classical method of least squares is applied to obtain $\hat{a}$ and $\hat{b}$ as

$$\hat{a} = \bar{L}_{(t+1)c} - b_1 \bar{L}_{tc} \ , \quad \hat{b} = \left\{ \sum_{t=1}^{n} L_{tc} L_{(t+1)c} - n \bar{L}_{tc} \bar{L}_{(t+1)c} \right\} \Big/ \left\{ \sum_{t=1}^{n} L_{tc}^2 - n \bar{L}_{tc}^2 \right\} \qquad (3.2)$$

where

$$\bar{L}_{tc} = \sum_{t=1}^{n} L_{tc} / n \ , \qquad \bar{L}_{(t+1)c} = \sum_{t=1}^{n} L_{(t+1)c} / n$$

3.1 Estimation of fuzzy error term

It may be noted from eq. (2.3) that when $\tilde{L}_{t+1}$ and $\tilde{L}_t$ are triangular fuzzy numbers, then the error term $\tilde{\varepsilon}_t$ will be triangular as well. Let $\tilde{E}_t = [-l, 0, r]$ be an estimate for $\tilde{\varepsilon}_t$. Then from eq. (2.3), estimated response is

$$\hat{\tilde{L}}_{t+1} = \hat{a} + \hat{b}\,\tilde{L}_t + \tilde{E}_t \qquad (3.3)$$

Different choices for *l* and *r* may be made. One extreme possibility is to take *l* and *r* so large that all observed responses may be covered. However, this would result in error term becoming too fuzzy to ensure precise estimation. The other extreme possibility is to take *l* and *r* as the smallest left and right spreads of the observed responses. However, in this case, estimated intervals for various responses would be too narrow to cover any observation.

For evaluation of validity of a fuzzy regression model, Kim and Bishu (1998) used a measure involving differences between membership values of observed and estimated fuzzy responses. When $\tilde{L}_t$ and $\tilde{\varepsilon}_t$ are triangular fuzzy numbers, $\hat{\tilde{L}}_{t+1}$ is also triangular. Difference between membership values of $\tilde{L}_{t+1}$ and $\hat{\tilde{L}}_{t+1}$ is

$$D_t = \int_{S_{\tilde{L}_{t+1}} \cup S_{\hat{\tilde{L}}_{t+1}}} \left| \mu_{\tilde{L}_{t+1}}(y) - \mu_{\hat{\tilde{L}}_{t+1}}(y) \right| \tag{3.4}$$

where $S_{\tilde{L}_{t+1}}$ and $S_{\hat{\tilde{L}}_{t+1}}$ are respectively the supports of $\mu_{\tilde{L}_{t+1}}(y)$ and $\mu_{\hat{\tilde{L}}_{t+1}}(y)$. Smaller the values of $D_{t,}$, better is the fit of fuzzy regression model to data. Therefore, our objective is to find the estimates for *l* and *r* that would minimize total difference between observed and estimated responses.

Let $l_{\min}$ and $r_{\min}$ be respectively smallest left and right spreads of observed responses. Determination of best values for *l* and *r* can be formulated as the following mathematical programming problem:

Minimize $\sum_{t=1}^{n} D_t$ subject to the constraints $\left(\hat{Y}_{tm} - \hat{Y}_{tl}\right) \ge l_{\min}, \left(\hat{Y}_{tu} - \hat{Y}_{tm}\right) \ge r_{\min}$ (3.5)

From eq. (3.3), estimated response $\hat{\tilde{L}}_{t+1}$ can be represented as

$$\hat{\tilde{L}}_{t+1} = [\hat{Y}_{tl},\ \hat{Y}_{tm},\ \hat{Y}_{tu}] = \left[\hat{a} + \hat{b}\, X_{tl} - l,\quad \hat{a} + \hat{b}\, X_{tm},\quad \hat{a} + \hat{b}\, X_{tu} - r\right] \tag{3.6}$$

As discussed by Kao and Chyu (2002), the expression for D*t* can be written as

$$D_t = 0.5\left[Y_{tu} - Y_{tl}\right] + 0.5\left[r + l + \hat{b}\left(X_{tu} - X_{tl}\right)\right] - 2\,(0.5)\left[Y_{tu} - \left(\hat{a} + \hat{b}X_{tl} - l\right)\right]\alpha_t \tag{3.7}$$

where

$$\alpha_t = height\left(\tilde{L}_{t+1} \cap \hat{\tilde{L}}_{t+1}\right) = \left(Y_{tu} - \hat{Y}_{tl}\right) / \left(Y_{tu} - Y_{tm} + \hat{Y}_{tm} - \hat{Y}_{tc}\right) \tag{3.8}$$

It may be noted that eq. (3.5) involves a nonlinear objective function and two bounded constraints and can be solved using a nonlinear programming software package.

4. Results and discussion

Nonlinear programming solver LINGO, Version 8, available at I.A.S.R.I., New Delhi is used for subsequent data analysis (LINDO, 2002). As an illustration, we consider the data of pearl oyster, *Pinctuda fucata* in Tuticorin Harbour farm, Gulf of Mannar, as given in Chellam (1988). For various monthly age-groups up to 36 months, intervals containing minimum and maximum length measurements of random samples of around 30 pearl oysters are reported. However, usual practice to analyze such a data set is to get rid of the situation of response variable being in intervals is by considering only the mean or mode, thereby losing a lot of vital information.

We now apply the Two-stage procedure discussed in last section to fit fuzzy von Bertalanffy growth model to above mentioned data set. In order to fit eq. (4), we first rearrange the data in the form given below:

Table 1. Triangular fuzzy intervals for lengths of pearl oyster for various age-groups

Age (t) (in months)	Length $\widetilde{L}_{t+1}$ (in mm.)				Length $\widetilde{L}_t$ (in mm.)		
	Y_{tl}	Y_{tm}	Y_{tu}		X_{tl}	X_{tm}	X_{tu}
1	9.0	13.4	17.8		3.0	7.1	11.2
2	12.0	18.0	24.0		9.0	13.4	17.8
3	19.0	23.6	28.3		12.0	18.0	24.0
4	22.0	28.6	35.2		19.0	23.6	28.3
5	27.0	31.5	36.0		22.0	28.6	35.2
6	31.7	39.4	47.0		27.0	31.5	36.0
8	35.0	40.0	45.0		31.7	39.4	47.0
9	41.5	47.2	52.8		35.0	40.0	45.0
10	42.0	48.8	55.6		41.5	47.2	52.8
11	43.2	49.6	56.0		42.0	48.8	55.6
12	46.0	51.3	56.5		43.2	49.6	56.0
14	47.0	52.0	57.0		46.0	51.3	56.5
15	44.0	51.0	58.0		47.0	52.0	57.0
16	47.0	53.0	59.0		44.0	51.0	58.0
17	48.0	53.5	59.0		47.0	53.0	59.0
18	51.0	56.1	61.2		48.0	53.5	59.0
20	49.0	55.5	62.0		51.0	56.1	61.2
21	52.4	57.4	62.5		49.0	55.5	62.0
22	54.0	59.8	65.5		52.4	57.4	62.5
23	54.0	60.0	66.0		54.0	59.8	65.5
24	54.0	60.2	66.5		54.0	60.0	66.0
25	56.0	61.5	67.0		54.0	60.2	66.5
26	56.0	61.8	67.5		56.0	61.5	67.0
27	56.5	63.0	69.5		56.0	61.8	67.5
29	56.2	63.4	70.5		56.5	63.0	69.5

The length values ($\widetilde{L}_t$) corresponding to first 25 values of age (*t*) would be used for fitting of eq. (2.3) while those for the last 5 values corresponding to age (*t*), viz. t=30, 31,…, 34 would be used for validation purposes. Using eq. (3.2), we get $\hat{a}$ = 7.05 and $\hat{b}$ = 0.90. Then eq. (3.6) gives

$$\hat{\tilde{L}}_{t+1} = \left[7.05 + 0.90X_{tl} - l,\ \ 7.05 + 0.90X_{tm},\ \ 7.05 + 0.90X_{tu} - r\right] \tag{4.1}$$

By substituting values of $\hat{a}$ and $\hat{b}$ in above equation and also values of X_{tl}, X_{tm},, and X_{tu} for first 25 values of t, we get $\hat{Y}_{tl}$, $\hat{Y}_{tm}$, $\hat{Y}_{tu}$ in terms of variables l and r; their values are, however, omitted here to save space. Using these values and the ones reported in Table 1, eq. (4.1) yields the values of α_t in terms of variables l and r. Subsequently, using eq. (3.7), we get the values of D_t and the same are reported in Table 2.

Table 2. Values of t and D_t

t	D_t
1	8.09+0.5*(r+l)-(8.06+l)*(8.06+l)/(8.09+l)+(0.69-r)*(0.69-r)/(0.71-r)
2	9.96+0.5*(r+l)-(8.87+l)*(8.87+l)/(9.96+l)
3	10.05+0.5*(r+l)-(9.61+r)*(9.61+r)/(10.05+r)+(r-0.31)*(r-0.31)/(0.69+r)
4	10.79+0.5*(r+l)-(10.48+r)*(10.48+r)/(10.79+r)+(2.12-l)*(2.12-l)/(2.47-l)
5	10.44+0.5*(r+l)-(9.19+l)*(9.19+l)/(10.44+l)+(0.19+l)*(0.19+l)/(1.44+l)
6	11.70+0.5*(r+l)-(7.7+r)*(7.7+r)/(11.70+r)
7	11.89+0.5*(r+l)-(9.47+l)*(9.47+l)/(11.89+l)+(l-0.53)*(l-0.53)/(1.92+l)
8	10.15+0.5*(r+l)-(5.98+r)*(5.98+r)/(10.15+r)
9	11.89+0.5*(r+l)-(11.26+l)*(11.26+l)/(11.89+l)+(1.11-r)*(1.11-r)/(1.77-r)
10	12.52+0.5*(r+l)-(11.20+l)*(11.20+l)/(12.52+l)
11	11.01+0.5*(r+l)-(10.63+l)*(10.63+l)/(11.01+l)+(0.13+l)*(0.13+l)/(0.45+l)
12	9.72+0.5*(r+l)-(8.61+l)*(8.61+l)/(9.72+l)
13	11.50+0.5*(r+l)-(8.71+l)*(8.71+l)/(11.50+l)
14	12.30+0.5*(r+l)-(12.17+r)*(12.17+r)/(12.30+r)+(r-1.83)*(r-1.83)/(r+0.29)
15	10.90+0.5*(r+l)-(9.71+l)*(9.71+l)/(10.90+l)
16	10.05+0.5*(r+l)-(9.07+r)*(9.07+r)/(10.05+r)
17	11.09+0.5*(r+l)-(9.12+l)*(9.12+l)/(11.09+l)
18	10.90+0.5*(r+l)-(10.36+r)*(10.36+r)/(10.90+r)+(r-0.26)*(r-0.26)/(r+0.74);
19	10.29+0.5*(r+l)-(9.21+r)*(9.21+r)/(10.29+r)
20	11.12+0.5*(r+l)-(10.81+r)*(10.81+r)/(11.12+r)

21	$11.65+0.5*(r+l)-(10.92+l)*(10.92+l)/(11.65+l)$
22	$11.12+0.5*(r+l)-(10.81+r)*(10.81+r)/(11.12+r)$
23	$10.70+0.5*(r+l)-(10.12+l)*(10.12+l)/(10.70+l)$
24	$11.68+0.5*(r+l)-(11.20+r)*(11.20+r)/(11.68+r)+(0.87-l)*(0.87-l)/(1.29-l)$
25	$13.00+0.5*(r+l)-(12.68+l)*(12.68+l)/(13.00+l)+(1-r)*(1-r)/(1.26-r)$

The constraints of eq. (3.5), after substituting the values of estimated $\hat{Y}_{tl}, \hat{Y}_{tm}, \hat{Y}_{tu}$ for each of first 25 values of *t*, reduce to just two inequalities $0.42 \le l < 1.29,\ 0 \le r < 0.75$. Therefore, using eq. (3.5), we minimize D_t for each value of *t.* To this end, a computer program was written in LINGO and the same is appended as Annexure-I. Optimal solution was obtained at the 26th iteration, with minimum values of objective function, as 54.59. Further, optimum values of left spread (l) and right spread (r) are respectively computed as 0.93 and 0.39; thus the estimate of fuzzy error term ($\tilde{\varepsilon}_t$) is the interval (-0.93, 0, 0.39). Using eq. (2.3), the estimated fuzzy model is then obtained as

$$\tilde{L}_{t+1} = 7.05 + 0.90\,\tilde{L}_t + (-0.93,\ 0,\ 0.39)$$

Further, from eq. (2.2),

$$\hat{k} = -\log_e \hat{b} = 0.105\,,\ \hat{L}_\infty = \hat{a}/\left(1 - e^{-k}\right) = 70.500$$

In order to validate the fitted fuzzy regression model, the estimated values for *t =30,31,…,34* are computed and reported in last column of Table 3. The corresponding observed values are reproduced in second column of this table.

Table 3. Validation of fitted model for out-of-sample data

t	**Observed**	**Estimated**
30	**(57.3, 71.5)**	**(56.2, 70.01)**
31	**(59.0, 72.0)**	**(57.61, 70.91)**
32	**(57.0, 73.0)**	**(59.14, 71.36)**
33	**(61.0, 74.0)**	**(57.34, 72.26)**
34	**(63.4, 76.0)**	**(60.93, 73.16)**

Evidently, the observed and estimated values are quite close for all values of $t =30,31,\ldots,34,$ indicating thereby that the fitted model performs satisfactorily. To get a visual idea, graph of estimated fitted model along with data is depicted in Figure 1.

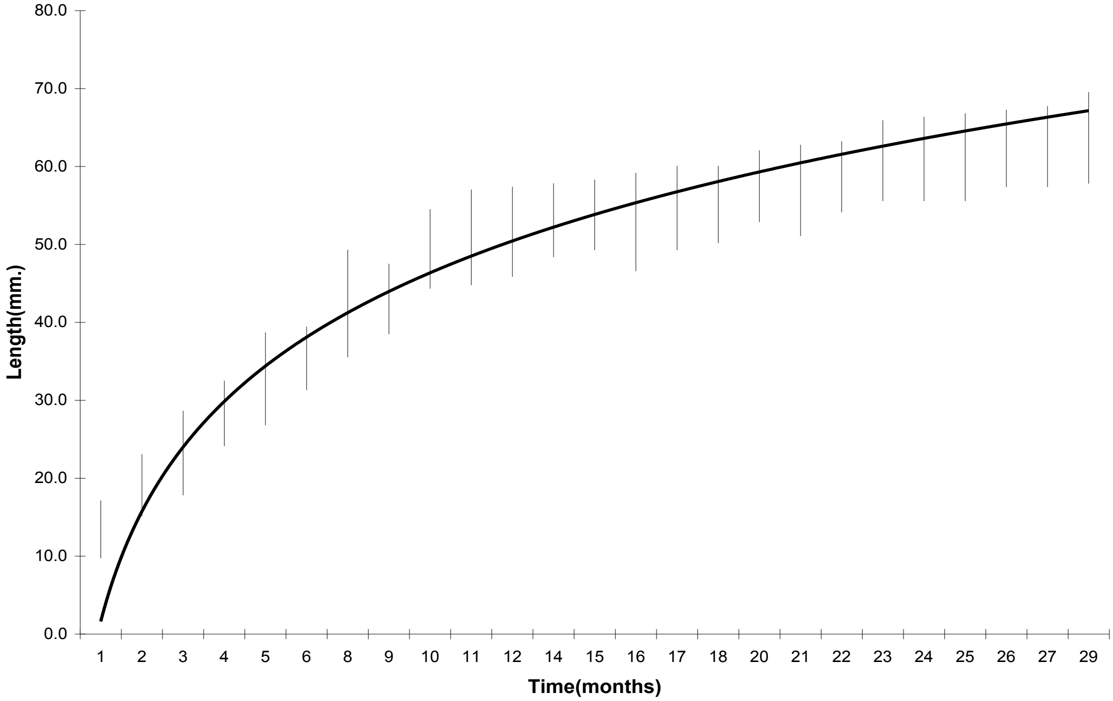


Figure1: Fitting of fuzzy von Bertalanffy growth model using Kao and Chyu's approach

It may be pointed out that the fitted model is able to provide a good fit to the data despite the fact that the observed data comprise overlapping class-intervals. Further, as only the minimum and maximum values of lengths of pearl oyster corresponding to various age-groups was reported in Chellam's (1988) paper, we were able to consider the form of membership functions of fuzzy lengths as triangular.

Had all the data values were known, it would have been possible to employ that particular membership function which could have been best for the data. To this end, possibility of fuzzy numbers to be described by some other membership function forms, like Trapezoidal, Gaussian, and harmonic could be explored. However, this would involve quite complicated mathematical expressions.

ANNEXURE-I

Program for optimum values of spreads of fuzzy errors

```
!Lingo Von Bertanlanffy;

Min=ERR_A1+ERR_A2+ERR_A3+ERR_A4+ERR_A5+ERR_A6+ERR_A7+ERR_A8+E
RR_A9+ERR_A10+ERR_A11+ERR_A12+ERR_A13+ERR_A14+ERR_A15+ERR_A16
+ERR_A17+ERR_A18+ERR_A19+ERR_A20+ERR_A21+ERR_A22+ERR_A23+ERR_
A24+ERR_A25;

ERR_A1=8.09+0.5*(R+L)-(8.06+L)*(8.06+L)/(8.09+L)+(0.69-R)*(0.69-R)/(0.71-R);
ERR_A2=9.96+0.5*(R+L)-(8.87+L)*(8.87+L)/(9.96+L);
ERR_A3=10.05+0.5*(R+L)-(9.61+R)*(9.61+R)/(10.05+R)+(R-0.31)*(R-0.31)/(0.69+R);
ERR_A4=10.79+0.5*(R+L)-(10.48+R)*(10.48+R)/(10.79+R)+(2.12-L)*(2.12-L)/(2.47-L);
ERR_A5=10.44+0.5*(R+L)-(9.19+L)*(9.19+L)/(10.44+L)+(0.19+L)*(0.19+L)/(1.44+L);
ERR_A6=11.70+0.5*(R+L)-(7.7+R)*(7.7+R)/(11.70+R);
ERR_A7=11.89+0.5*(R+L)-(9.47+L)*(9.47+L)/(11.89+L)+(L-0.53)*(L-0.53)/(1.92+L);
ERR_A8=10.15+0.5*(R+L)-(5.98+R)*(5.98+R)/(10.15+R);
ERR_A9=11.89+0.5*(R+L)-(11.26+L)*(11.26+L)/(11.89+L)+(1.11-R)*(1.11-R)/(1.77-R);
ERR_A10=12.52+0.5*(R+L)-(11.20+L)*(11.20+L)/(12.52+L);
```

```
ERR_A11=11.01+0.5*(R+L)-
(10.63+L)*(10.63+L)/(11.01+L)+(0.13+L)*(0.13+L)/(0.45+L);
ERR_A12=9.73+0.5*(R+L)-(8.61+L)*(8.61+L)/(9.73+L);
ERR_A13=11.50+0.5*(R+L)-(8.71+L)*(8.71+L)/(11.50+L);
ERR_A14=12.30+0.5*(R+L)-(12.17+R)*(12.17+R)/(12.30+R)+(R-0.17)*(R-
0.17)/(R+0.29);
ERR_A15=10.90+0.5*(R+L)-(9.71+L)*(9.71+L)/(10.90+L);
ERR_A16=10.05+0.5*(R+L)-(9.07+R)*(9.07+R)/(10.05+R);
ERR_A17=11.09+0.5*(R+L)-(9.12+L)*(9.12+L)/(11.09+L);
ERR_A18=10.90+0.5*(R+L)-(10.36+R)*(10.36+R)/(10.90+R)+(R-0.26)*(R-
0.26)/(R+0.74);
ERR_A19=10.30+0.5*(R+L)-(9.21+R)*(9.21+R)/(10.30+R);
ERR_A20=11.18+0.5*(R+L)-(10.81+R)*(10.81+R)/(11.18+R);
ERR_A21=11.65+0.5*(R+L)-(10.92+L)*(10.92+L)/(11.65+L);
ERR_A22=11.13+0.5*(R+L)-(10.81+R)*(10.81+R)/(11.13+R);
ERR_A23=10.70+0.5*(R+L)-(10.12+L)*(10.12+L)/(10.70+L);
ERR_A24=11.68+0.5*(R+L)-(11.20+R)*(11.20+R)/(11.68+R)+(0.87-L)*(0.87-L)/(1.29-
L);
ERR_A25=13.00+0.5*(R+L)-(12.68+L)*(12.68+L)/(13.00+L)+(1-R)*(1-R)/(1.26-R);

L>=0.42;
L<1.29;
R>=0;
R<0.75;
```

CHAPTER VI

ARTIFICIAL NEURAL NETWORK METHODOLOGY FOR MODELLING AND FORECASTING MAIZE CROP YIELD

---

1. Background

Multiple linear regression (MLR) modelling is a very powerful technique and is widely used to estimate linear relationship between response variable and predictors. Its main limitation is that it is useful only when underlying relation between response and predictor variables is assumed to be “linear”. However, in a realistic situation, this assumption is rarely satisfied. Also, if there are several predictors, it is not possible to have an idea of underlying nonlinear functional relationship between response and predictor variables. Fortunately, to handle such a situation, an extremely versatile approach of “Artificial neural networks” (ANNs) is rapidly developing. Cheng and Titterington (1994) have reviewed the ANN methodology from a statistical perspective, while Warner and Misra (1996) have laid emphasis on understanding ANN as a statistical tool.

A distinguishing feature of ANNs that makes them valuable and attractive for a statistical task is that, as opposed to traditional model-based methods, ANNs are data driven self-adaptive methods in that there are a few apriori assumptions about the models for problems under study. This modelling approach with ability to learn from experience is very useful in many practical problems since it is often easier to have data than to have good theoretical guesses about underlying laws governing the systems from which data are generated. Recently, in an excellent article, Zhang (2007) has discussed various pitfalls in ANN modelling work, which must be avoided.

Most widely used ANN is Multilayered feedforward artificial neural network (MLFANN). Purpose of this work is to thoroughly discuss its various aspects. As an illustration, the methodology is applied for modelling and forecasting maize crop yield on the basis of four predictor variables, viz. Total human labour, Farm power, Fertilizer consumption, and Pesticide consumption by taking part of data culled from Singh *et al.* (2004). MLFANN with zero, one, and two hidden layers are considered. Optimum numbers of hidden layers as well as optimum numbers of units in each hidden layer are found by computing mean square errors.

2. Methodology

2.1 Preliminaries of ANN

ANN can be defined as interconnected assembly of simple processing elements (or units/node/neurons). The processing ability of network is stored in inter-unit connection strengths or weights obtained by a process of learning from a set of training patterns. A typical ANN consists of one input layer, one output layer and hidden layers. Each layer can have several units whose output is a function of weighted sum of their inputs. Input into a node is a weighted sum of outputs from nodes connected to it. Thus net input into a node *i* is (For details, see Warner and Misra, pp. 286, 1996)

$$Netinput_i = \sum \left(w_{ij} * output_j \right) + u_i \tag{2.1}$$

where w_{ij} are weights connecting neuron j to neuron i, $output_j$ is output from unit j and u_i is a threshold for neuron i. Threshold term is baseline input to a node in absence of any other inputs. If a weight w_{ij} is negative, it is termed *inhibitory* because it decreases net input, otherwise it is called *excitatory*.

Each unit takes its net input and applies an *activation function* to it. For example, output of *j* th unit, also called *activation value* of the unit, is

$g\left(\sum w_{ji} x_i\right)$, where $g(\,.\,)$ is activation function and x_i is output of *i* th unit connected to unit *j*. Two important activation functions commonly used are :

(i) Pureline:

$$g(netinput) = constant\,.\,(netinput) \tag{2.2}$$

(ii) Sigmoidal:

$$g(netinput) = 1/[1 + exp(-netinput)] \tag{2.3}$$

With no hidden units, an ANN can classify only linearly separable problems (ones for which possible output values can be separated by global hyperplanes). However, it has been shown by Cybenko (1989) that with one hidden layer, an ANN can describe any continuous function (if there are enough hidden units), and that with two hidden layers, it can describe any function.

The weights in an ANN, similar to coefficients in a regression model, are adjusted to solve the problem presented to ANN. *Learning* or *training* is used to describe process of finding values of these weights. Two types of learning with ANN are *supervised* and *unsupervised* learning. Supervised learning occurs when there is a known target value associated with each input in training set. Output of ANN is compared with target value, and this difference is used to train ANN (alter the weights). Unsupervised learning is needed when training data lack target output values corresponding to input patterns. ANNs discussed so far are constructed with layers of units, and thus are termed *Multilayered* ANNs. A layer of units in such an ANN is composed of units that perform similar tasks.

2.2 Multilayered feedforward artificial neural network (MLFANN)

A MLFANN is one where units in one layer are connected only to units in the next layer, and not to units in a preceding layer or units in the same layer. MLFANN can have a number of hidden layers with a variable number of hidden units per layer. When counting layers, it is common practice not to count input layer because it does not perform any computation, but simply passes data onto next layer. So an MLFANN with an input layer, one hidden layer, and an output layer is termed a two-layered MLFANN.

MLFANN is perhaps the most popular network architecture. This is the type of network in which units are arranged in a layered feedforward topology as shown in Figure 1. The network thus has a simple interpretation as a form of input-output model, with weights and thresholds (biases) as free parameters of the model. Such networks can model functions of almost arbitrary complexity, with the number of layers, and the number of units in each layer, determining the function complexity.

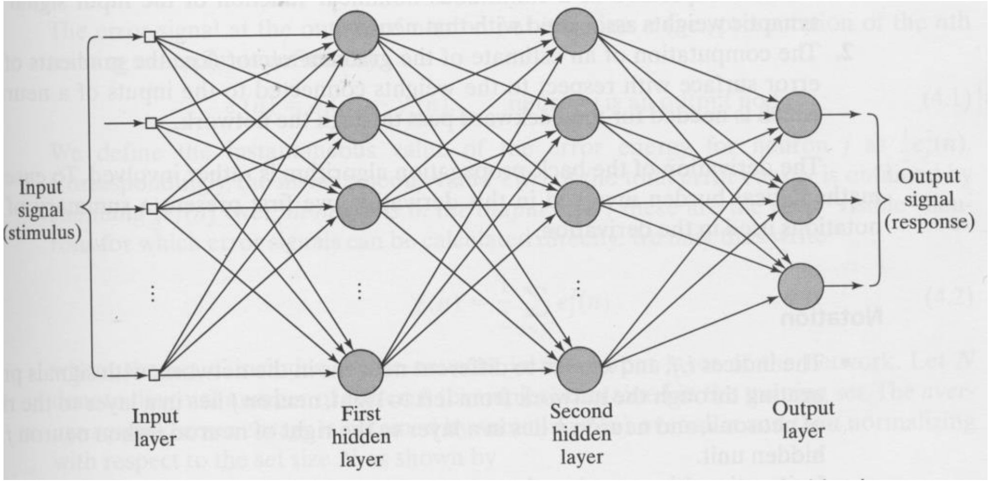


Figure 1: A Multilayered feedforward artificial neural network

Neural networks are constructed by learning from repeated presentation of inputs (the x's) and outputs (the y's) and adjusting internal parameters so as to minimize error between fitted and desired y. Neural network can be seen as a general way to parameterize data through arbitrary nonlinear functions from space of predictor variables to the space of response variables. The great utility and flexibility of neural network arises from application of learning algorithms that allow the network to construct correct weights, and hence the desired function, for a given set of observations.

2.3 Learning algorithms

As the input–output vectors are presented to the network, a learning algorithm adjusts connection weights until the system converges on a function that correctly reproduces the output. Optimal connection weights may be obtained by using Gradient descent algorithm or Conjugate gradient descent algorithm with a view to minimizing sum of squared error function of the network output.

2.3.1 Gradient descent algorithm (GDA)

In order to optimize weights, objective function to be minimized is generally taken as sum of squared errors defined as

$$E = 0.5 \sum \sum \left(y_{pk} - Y_{pk} \right)^2 \tag{2.4}$$

where subscript *p* refers to patterns (observations) with a total of *n* patterns, subscript *k* to output unit with a total of O output units, *y* and *Y* are respectively observed and estimated responses.

As input units simply pass information to hidden units, input into *j* th hidden unit is

$$h_{pj} = \sum w_{ji} x_{pi} \tag{2.5}$$

Here w_{ji} is weight from input unit *i* to hidden unit *j*, and x_{pi} is value of *i*th input for pattern *p*. The *j*th unit applies an activation function say, sigmoid function given by eq.(2.3), to its net input and outputs:

$$v_{pj} = g\left(h_{pj}\right) = 1 / \left(1 + exp\left(- h_{pj}\right)\right) \tag{2.6}$$

Similarly, output unit *k* receives a net input of

$$f_{pk} = \sum W_{kj} v_{pj} \tag{2.7}$$

Here W_{kj} represent weight from hidden unit *j* to output *k*. The unit then outputs quantity

$$Y_{pk} = g(f_{pk}) = 1/(1 + exp(-f_{pk})) \tag{2.8}$$

Eqs. (2.5) to (2.8) demonstrate that objective function given by eq. (2.4) is a function of unknown weights w_{ji} and W_{kj}. So we evaluate partial derivative of objective function with respect to weights, and then move weights in a direction down the slope, continuing until error function no longer decreases. Mathematically, this can be expressed as

$$\Delta W_{kj} = -\eta \, \partial E / \partial W_{kj} \tag{2.9}$$

The η term is known as *learning rate* and simply scales step size. Substituting eqs. (2.5) to (2.8) in eq. (2.4) and expanding eq. (2.9) using chain rule, we get

$$\partial Y_{pk} / \partial f_{pk} = g'(f_{pk}) = Y_{pk}(1 - Y_{pk}) \tag{2.10}$$

and

$$\partial f_{pk} / \partial W_{kj} = v_{pj} \tag{2.11}$$

Substituting these results back in eq.(2.9), change in weights from hidden units to output units is given by

$$\Delta W_{kj} = -\eta [(-1)(y_{pk} - Y_{pk})] Y_{pk}(1 - Y_{pk}) v_{pj} \tag{2.12}$$

Weights are updated as

$$W_{kj}(t+1) = W_{kj}(t) + \Delta W_{kj} \tag{2.13}$$

Similarly, calculations for weights from inputs to hidden units can be carried out as given in Warner and Misra (1996). Finally, the algorithms, following along similar lines as Hertz *et al.* (1991) are as follows:

(i) Initialize the weights to small random values. This puts the output of each unit around 0.5.

(ii) Choose a pattern *p* and propagate it forward. This yields values for v_{pj} and Y_{pk}, the outputs from the hidden layer and output layer.

(iii) Compute the output errors: $\delta_{pk} = (y_{pk} - Y_{pk}) g'(f_{pk})$

(iv) Compute the hidden layer errors: $\psi_{pj} = \sum \delta_{pk} W_{kj} v_{pi} (1 - v_{pj})$

(v) To update the weights, compute: $\Delta W_{kj} = \eta \delta_{pk} v_{pj}$ and $\Delta w_{ji} = \eta \psi_{pj} i_{pi}$

Repeat the steps for each pattern.

2.3.2 Conjugate gradient descent algorithm (CGDA)

The basic GDA adjusts weights in steepest descent direction (negative of gradient). This is the direction in which performance function is decreasing most rapidly. It turns out that, although function decreases most rapidly along negative of gradient, this does not necessarily produce fastest convergence. In CGDA, search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions.

In most of training algorithms a learning rate is used to determine length of weight update (step size). In CGDA, step size is adjusted at every iteration. A search is made along conjugate gradient direction to determine step size, which minimizes performance function along that line. As CGDA requires only a little more storage than GDA, this is often a good choice for networks with a large number of weights (greater than 100).

3. An Illustration

Singh *et al.* (2004) carried out a study dealing with various aspects of Maize crop. In present illustration, part of the data from State of Uttar Pradesh covering 170 farmers, for whom complete data were available, is considered. Specifically, response variable taken is: Maize crop yield, while four predictors are: Total human labour (T.H.L.) (Rs/hectare), Farm Power ☐ (F.P.) (Rs/hectare), Fertilizer consumption (F.C.) (Kg/hectare) and Pesticide consumption (P.C.) (Rs/hectare). Graphical representation of training data on semi-logarithmic scale is exhibited in Figure 2.

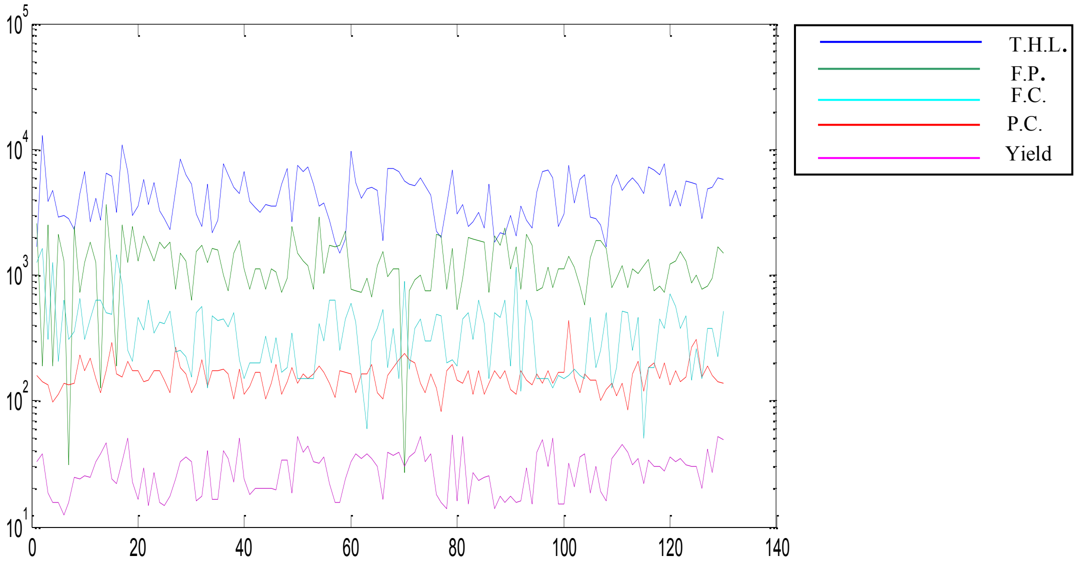


Figure 2: MATLAB plot of variables under training data set

Neural Network Toolbox in MATLAB® (2006) available at IASRI, New Delhi is employed to train the MLFANN. Before training, input and target values were preprocessed using suitable scaling, so that they fall within a specified range. Available 170 observations are divided into three subsets: (i)

First subset is training set comprising 130 observations, which is used for computing gradient and updating the network weight and biases, (ii) Second set of 30 observations comprise validation set, and (iii) Test set comprise remaining 10 observations.

MLFANN was trained using both GDA and CGDA. Several possibilities were tried. When no layer was taken as hidden layer, only input and output layer were used. Activation function employed was “Pureline”. When one hidden layer was considered, activation function between input layer and hidden layer was taken as “Sigmoidal” while that between hidden layer and output layer, “Pureline” activation function was used. Further, with two hidden layers, Sigmoidal activation functions in the hidden layer and a linear transfer function in the output layer is used. The purpose of doing so is that if the last layer of a MLFANN has sigmoid neurons, then the outputs of the network are limited to a small range because of the “squashing” property of sigmoid function. If linear output neuron are used the network output can take on any value.

Performance of the trained network can be measured by mean square on the training, validation and test sets, but it is often useful to investigate the network response in more detail. One option is to perform a regression analysis between network response and the corresponding targets. A large number of networks were trained, correlation coefficient values obtained between network output and target values for training data is reported in Table 1. Figure 3 depicts graph between output and target values for MLFANN (11-16-1) model using conjugate gradient descent algorithm. Evidently, the two sets of values are seen to be extremely close to each other.

Table 1. Correlation coefficients between output and target yield values

Training methods Number of neurons on hidden layers	GDA	CGDA
(i) No hidden layer	0.437	0.639
(ii) One hidden layer		
8	0.705	0.811
12	0.691	0.847
15	0.698	0.874
(ii) Two hidden layers		
(5, 10)	0.738	0.793
(8,13)	0.773	0.916
(11,16)	0.808	0.933

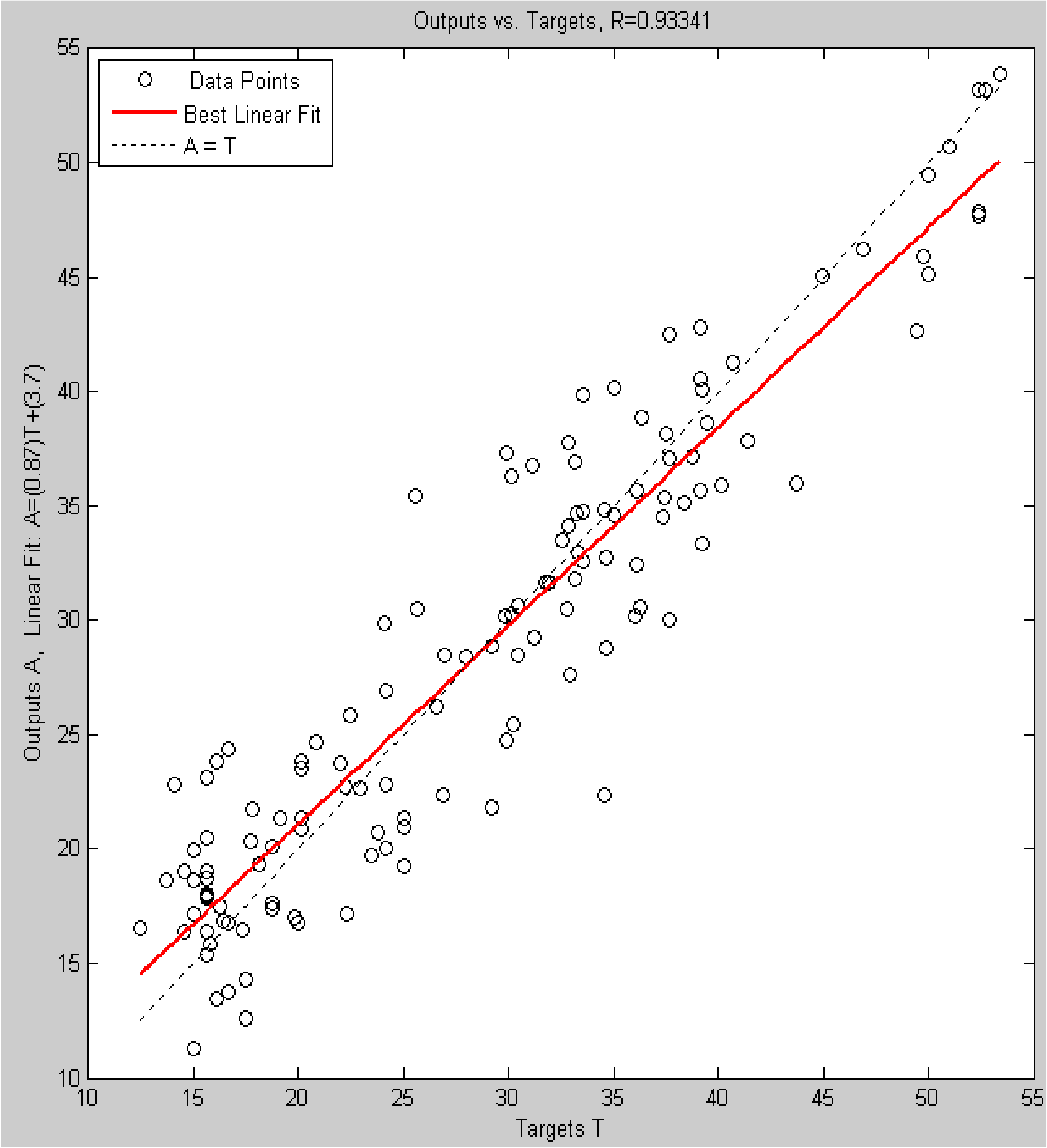


Figure 3: Graph between output and target values for MLFANN(11-16-1) model using conjugate gradient descent algorithm

We use MSE for network performance evaluation. In Table 2, MSEs for training as well as validation sets are summarized for both GDA and CGDA. Figure 4 shows how training goal was met by best network 11-16-1 using CGDA.

***Table 2.** MSE for training and validation data using both learning algorithms*

Training methods Number of neurons in hidden layers		GDA	CGDA
(i) No hidden layer	Training	69.01	69.01
	Validation	41.40	41.38
(ii) One hidden layer			
8	Training	55.61	33.00
	Validation	29.98	0.013
12	Training	54.54	32.46
	Validation	35.68	0.003
15	Training	54.78	22.86
	Validation	14.73	0.003
(ii) Two hidden layers			
(5, 10)	Training	52.88	43.05
	Validation	14.35	0.22
(8,13)	Training	46.72	18.68
	Validation	17.92	0.004
(11,16)	Training	40.32	12.94
	Validation	20.39	0.003

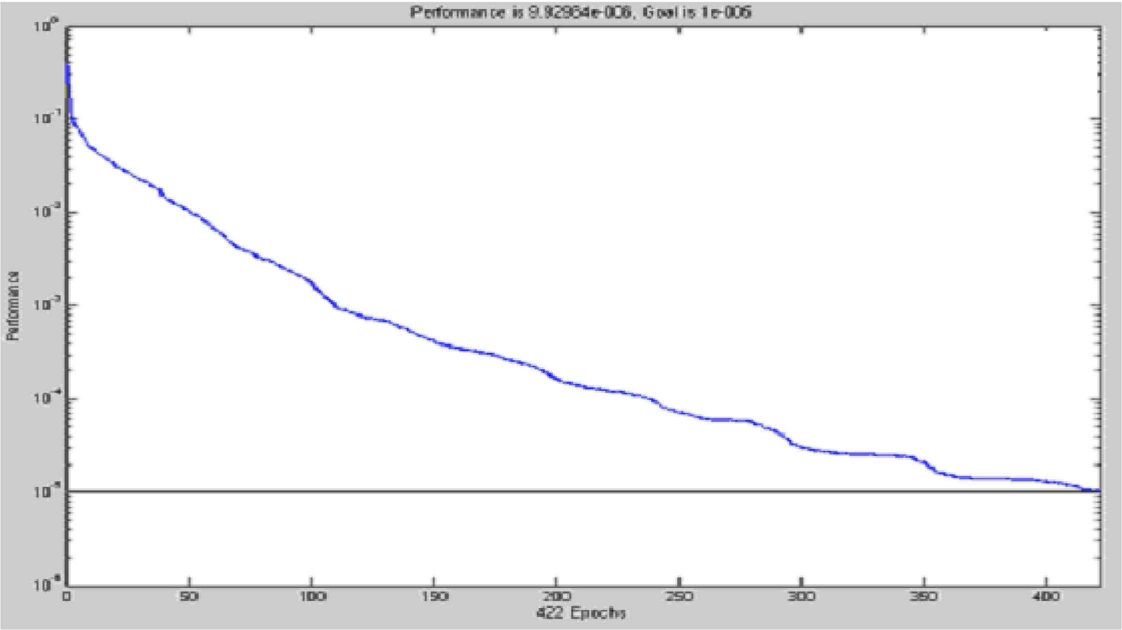


Figure 4: Performance of MLFANN (11-16-1) model for training data using conjugate gradient descent algorithm

The computer program written in MATLAB to train MLFANN using the two training algorithms is given in Annexure I. As CGDA is a faster learning method than GDBP, therefore, number of epochs used to train the MLFANN using CGDA is fewer than that for GDA. The MSEs for best trained MLFANN (11-16-1) using CGDA and for traditionally used MLR are respectively computed as 12.94 and 69.01, thereby clearly demonstrating superiority of MLFANN (11-16-1) over MLR for data under consideration. Finally, for test data comprising 10 observations, predicted values of response variable using MLFANN (11-16-1) model along with actual values are reported in Table 3. Evidently, predicted and actual values are quite close. Thus, artificial neural network methodology is successful in describing given data.

Table 3. Performance of MLFANN (11-16-1) model for test data

Observation number	Predicted values	Actual values
161	15.23	17.75
162	39.23	38.21
163	27.49	28.75
164	34.96	32.02
165	40.38	39.94
166	39.48	38.97
167	21.46	26.46
168	18.50	13.75
169	37.56	39.34
170	48.46	52.50

4. Conclusions

In this work, potential of Artificial neural network methodology is highlighted for successfully tackling the realistic situation in which exact nonlinear functional relationship between response variable and a set of predictors is not known. Although ANNs may not be able to provide same level of insight as many statistical models, it is not correct to treat them as “black boxes”. In fact, one active area of research area in ANN is in the understanding of effect of predictors on response variable. It is hoped that, in future, research workers would start applying not only MLFANN but also some of the other more advanced ANN models, like Radial basis function neural network, and Generalized regression neural network.

ANNEXURE I

MATLAB Computer Program to Train MLFANN using GDA and CGDA

```
   load 'book11.txt';                  % First load the data set;
   t=book11(1:130, 1)';                % Response data of training set;
   p=book11(1:130,2:5)';               % Explanatory data of training set;
   P=book11(131:160,2:5)';             % Explanatory data of validation set;
   T=book11(131:160, 1)';              % Response data of validation set;
   %test.P=book11(161:170,2:5)';       % Explanatory data of test  set;
   %test.T=book11(161:170, 1)';        % Response data of test set;
   % Step for pre-processing of training data set;
            [pn,minp,maxp,tn,mint,maxt] = premnmx(p,t);
   % Creating of MLP neural network with two hidden layer using gradient
   descent back propagation algorithms;

net=newff(minmax(pn),[11,16,1],{'logsig','logsig','purelin'},'traingd');
   % Creating of MLP neural network with single hidden layer using
   conjugate gradient descent algorithms;
          %net=newff(minmax(pn),[15,1],{'logsig','purelin'},'traincgp');
            net.trainParam.show = 5;
   % Not included while training using Conjugate descent algorithms;
            net.trainParam.lr=0.05;
    % Number of epoch is 500 for Conjugate gradient descent algorithms;
            net.trainParam.epochs = 1500;
            net.trainParam.goal=1e-5;
   % Randomly initialize the neural network for training data set;
            net = init(net);
   % Train the neural network for training data set;
            [net,tr]=train(net,pn,tn);
   % Simulate the network for training data set;
           an=sim(net,pn);
   % Convert the result in orginal scale of measurement;
           a=postmnmx(an,mint,maxt);
   plot(a,pn,'rd')
           [m,b,r]=postreg(a,t)
   % Repeat the same above described process for validation of data set;
       [Pn,minP,maxP,Tn,minT,maxT] = premnmx(P,T);
    net1=newff(minmax(Pn),[11,16,1],{' logsig ',' logsig ','purelin'},'traingd');
       net1.trainParam.show = 5; net.trainParam.lr=0.05;
       net1.trainParam.epochs = 1500;
       net1.trainParam.goal = 1e-5;
       net1 = init(net1); [net1,tr]=train(net1,Pn,Tn); cn=sim(net1,Pn);
       c=postmnmx(cn,minT,maxT);
       %plot(c,Pn,'bd')
```

```
%[m,b,r]=postreg(c,T);
```

Architecture of Neural Network

```
     numInputs: 1
     numLayers: 3
   biasConnect: [1; 1; 1]
  inputConnect: [1; 0; 0]
  layerConnect: [0 0 0; 1 0 0; 0 1 0]
 outputConnect: [0 0 1]
 targetConnect: [0 0 1]

    numOutputs: 1  (read-only)
    numTargets: 1  (read-only)
numInputDelays: 0  (read-only)
numLayerDelays: 0  (read-only)

subobject structures:

        inputs: {1x1 cell} of inputs
        layers: {3x1 cell} of layers
       outputs: {1x3 cell} containing 1 output
       targets: {1x3 cell} containing 1 target
        biases: {3x1 cell} containing 3 biases
  inputWeights: {3x1 cell} containing 1 input weight
  layerWeights: {3x3 cell} containing 2 layer weights

functions:

      adaptFcn: 'trains'
       initFcn: 'initlay'
    performFcn: 'mse'
      trainFcn: 'traincgp'

parameters:

    adaptParam: .passes
     initParam: (none)
  performParam: (none)
    trainParam: .epochs, .goal, .max_fail,
                .min_grad, .show, .time

weight and bias values:

            IW: {3x1 cell} containing 1 input weight matrix
            LW: {3x3 cell} containing 2 layer weight matrices
             b: {3x1 cell} containing 3 bias vectors
```

CHAPTER VII

NEURO-FUZZY APPROACH FOR MODELLING AND FORECASTING

---

1. Background

In a realistic situation, assumptions of Multiple linear regression (MLR) methodology is rarely satisfied. Another limitation of MLR is that underlying phenomenon, response variable, and predictors are all assumed to be "crisp" or "precise". In reality, one or more of these is vague, or imprecise, or fuzzy. Accordingly, area of "Fuzzy logic" has been developed (Klir and Yuan, 2000). In chapter V of this thesis "Artificial neural networks" (ANNs) was developed to forecast maize crop yield on the basis of four predictors.

Incorporating the concept of Fuzzy in ANN methodology, a rapidly developing area of "Neuro-fuzzy" has been emerging (Rutkowaska, 2002). In an excellent paper, Abraham *et al.* (2004) applied Neuro-fuzzy techniques for forecasting time-series meteorological subdivisions level data of Kerala. However, in that paper, development of the process only over time was considered. Extension and application of this type of work when several predictors are present, is a challenging task.

Purpose of this work is to make an indepth study of "Adaptive neuro-fuzzy inference system (ANFIS)", which is most popular in the family of Neuro-fuzzy modelling. As an illustration, ANFIS model, using "MATLAB Fuzzy Logic Toolbox" is applied for development of a forecasting model for secondary data of yield of 100 banana plants on the basis of data at six different stages of growth using biometrical characters, like Plant height, plant girth, and leaf length as predictors.

2. Neuro-fuzzy computing

In a real world situation, traditional equation based techniques are not suitable for modeling nonlinearity. Neuro-fuzzy computing (Jang *et al.,* 2004) is a judicious integration of merits of neural and fuzzy approaches. This incorporates generic advantages of Artificial neural networks, like massive parallelism, robustness, and learning in data-rich environments into the system. Modelling of imprecise and qualitative knowledge as well as transmission of uncertainty is possible through use of Fuzzy logic (Klir and Yuan, 2000). Present study is based on ANFIS, which is a multilayered feedforward ANN consisting of nodes and directional links through which nodes are connected. Moreover, part or all the nodes are adaptive, which means that their outputs depend on incoming signals and on the parameter(s) pertaining to these nodes. ANFIS either uses input/output data sets to construct a fuzzy inference system whose membership functions are tuned using a learning algorithm or an expert may specify a fuzzy inference system and then the system is trained with data pairs by an adaptive network. Conceptual diagram of ANFIS is shown in Figure 1.

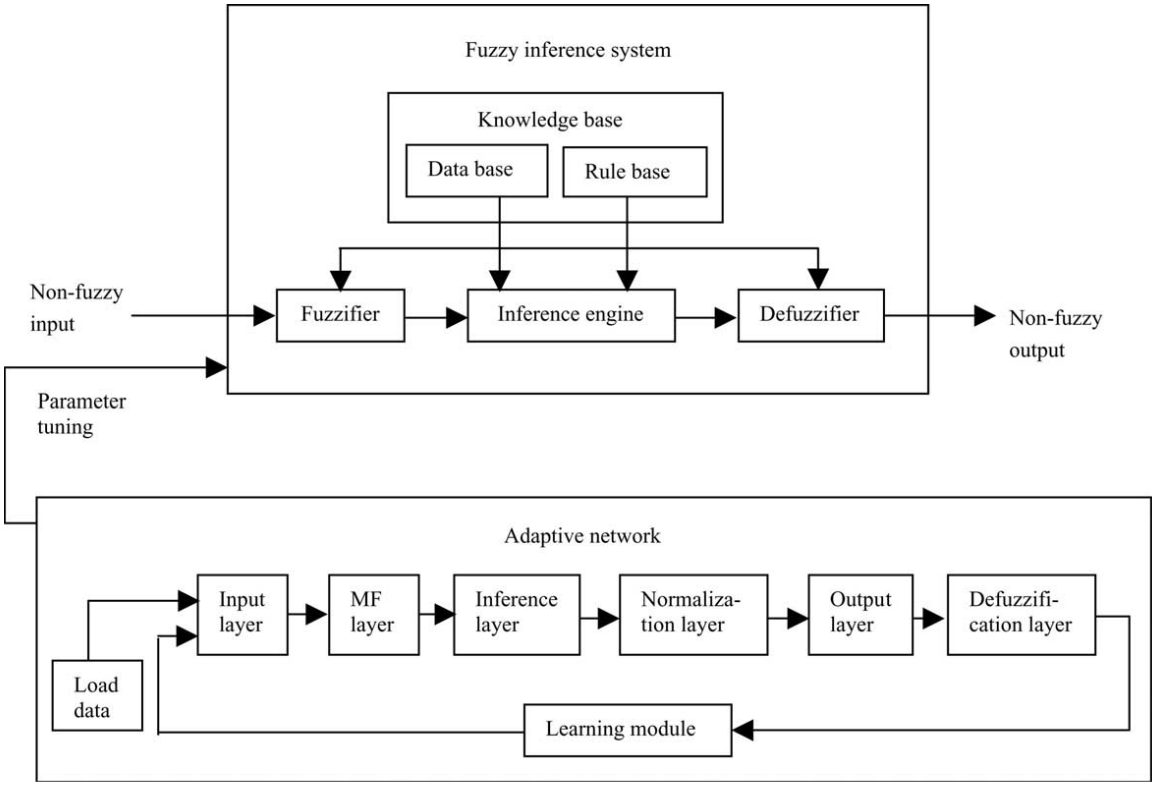


Figure 1: Conceptual diagram of ANFIS

A Fuzzy inference system (FIS) has five functional blocks. A fuzzifier converts real numbers of input into fuzzy sets. The database (or dictionary) contains the membership functions of fuzzy sets. The membership functions provide flexibility to fuzzy sets in modeling. A rule base consists of a set of linguistic statements of the form, if x is A then y is B, where A and B are labels of fuzzy sets on universes of discourse X and Y, respectively. An inference engine performs the inference operations on the rules to infer the output by a fuzzy reasoning method. Defuzzifier converts the fuzzy outputs obtained by inference engine into a non-fuzzy output real number domain. In order to incorporate the capability of learning from input/output data sets in fuzzy inference systems, a corresponding Adaptive network is generated. An Adaptive network is a multilayered feedforward network consisting of nodes and directional links through which nodes are connected. As shown in Fig.1, layer 1 is the input layer; layer 2 describes membership functions of each fuzzy input. Layer 3 is inference layer and normalization is performed in layer 4. Layer 5 gives output and layer 6 is defuzzification layer. Learning rule specifies how parameters of adaptive nodes should be changed to minimize a prescribed error measure. Change in values of parameters results in change in shape of membership functions associated with FIS. The modelling process based on ANFIS can broadly be classified in three steps:

Step 1: System identification

First step in system modelling is identification of input and output variables called system's variables.

Step 2: Determining network structure

Once input and output variables are identified, Neuro-fuzzy system is realized using a six-layered network.

Layer 1 (Input layer)

Each node in layer 1 represents input variables of the model. This layer simply transmits these input variables to fuzzification layer.

Layer 2 (Fuzzification layer)

This layer describes membership function of each input fuzzy set. Membership functions are used to characterize fuzziness in fuzzy sets. Output of each node i in this layer is given by $\mu_{A_i}(x_i)$, where $\mu_A(x)$ denotes membership function. Its value on unit interval [0,1] measures degree to which element x belongs to fuzzy set A, x_i is input to node i and A_i is linguistic label for each input variable associated with this node. Gausian membership functions are employed, since they are nonlinear and smooth and their derivatives are continuous and is given by

$$\mu(x) = exp[-(x-b/a)]^2 \tag{2.1}$$

Layer 3 (Inference layer)

Third layer is the inference layer. Each node in this layer is a fixed node and represents the IF part of a fuzzy rule. This layer aggregates membership grades using any fuzzy intersection operator which can perform fuzzy AND operation. The fuzzy intersection operators are commonly referred to as T-norm (triangular norm) operators. Most frequently used T-norm operators are *min* or *product* operators. For example, IF x_1 is A_1 AND x_2 is A_2 AND x_3 is A_3, THEN y is $f(x_1,x_2,x_3)$, where $f(x_1,x_2,x_3)$ is a linear function of input variables or may be a constant. The output of *i*th node is given as

$$w_i = \mu_{A_1}(x_1) \times \mu_{A_2}(x_2) \times \mu_{A_3}(x_3) \tag{2.2}$$

Layer 4 (Normalization layer)

The ith node of this layer is also a fixed node and calculates ratio of ith rule's firing strength in inference layer to sum of all the rules' firing strengths as

$$\overline{w_i} = w_i / (w_1 + w_2 + ... + w_R) \tag{2.3}$$

where $i = 1,2,...,R$ and R is total number of rules.

Layer 5 (Output layer)

This layer represents the THEN part (i.e. the consequent) of the fuzzy rule. The operation performed by the nodes in this layer is to generate qualified consequent (either fuzzy or crisp) of each rule depending on firing strength. Every node i in this layer is an adaptive node. The output of the node is computed as

$$O_i = \overline{w}_i f_i \tag{2.4}$$

where w_i is a normalized firing strength from layer 3 and f_i is a linear function of input.

Layer 6 (Defuzzification layer)

This layer aggregates consequents to produce a crisp output. The single node in this layer is a fixed node. It computes weighted average of output signals of output layer as

$$O = \sum_i O_i = \sum_i \overline{w}_i f_i = \sum_i w_i f_i \Big/ \sum_i w_i \tag{2.5}$$

Step 3: Learning algorithm and parameter tuning

ANFIS model fine-tunes parameters of membership functions using either backpropagation learning algorithm or hybrid learning rule. Backpropagation algorithm is an error-based supervised learning algorithm. It uses gradient descent method to update parameters. Network output is compared with desired output values. The error measure EP, for pattern P at the output node in layer 6, may be given as

$$E^P = 1/2\left(T^P - O_6^P\right)^2 \tag{2.6}$$

where T^P is target or desired output and O_6^P, single node output of defuzzification layer in the network. Further, sum of squared errors for entire training data set is

$$E^P = \sum_P E^P = \frac{1}{2}\sum_P \left(T^P - O_6^P\right)^2 \tag{2.7}$$

The error measure with respect to node output in layer 6 is given by

$$\delta = \partial E/\partial O_6 = -\left(T - O_6\right) \tag{2.8}$$

This delta value gives the rate at which output must be changed in order to minimize error function. This delta value must be propagated backward to inner layers in order to distribute error of output unit to all layers connected to it and adjust corresponding parameters. The delta value for layer 5 is given as

$$\partial E/\partial O_5 = \left(\partial E/\partial O_6\right) * \left(\partial O_6/\partial O_5\right) \tag{2.9}$$

Now, if α is a set of design parameters of the given adaptive network, then

$$\partial E/\partial \alpha = \sum_{O' \in P}\left(\partial E/\partial O'\right) * \left(\partial O'/\partial \alpha\right) \tag{2.10}$$

where P is set of adaptive nodes whose output depends on α. Thus, update for parameter α is given by

$$\Delta\alpha = -\eta * \left(\partial E/\partial \alpha\right) \tag{2.11}$$

Here η is learning rate:

$$\eta = k \Big/ \sqrt{\sum_\alpha \left(\partial E/\partial \alpha\right)^2} \tag{2.12}$$

where k is step size. Value of k must be properly chosen as change in value of k influences rate of convergence. Thus, design parameters are tuned according to real input/output data pairs of the system. Change in values of the parameters results in change in shape of membership functions initially defined by an expert. The new membership functions thus obtained after training gives a more realistic model of the system.

2. An Illustration

An hectare of banana yields 40 million calories of energy as compared to 2.5 million calories by wheat (Rao, 2005). India ranks second among banana producing countries of the world. Venugopalan and Shamasundaran (2005) developed a statistical model for evolving crop-logging parameters across different growth stages of banana plants collected from farmers' field located at Kestur, Bangalore. In present study, data culled from Venugopalan and Shamasundaran (2005), has been used to apply ANFIS to forecast banana yield at different stages of its growth using a number of predictors, like number of leaves (N.O.L.), plant height (P.H.) (cm.), plant girth (P.G.) (cm.), leaf length (L.L.) (cm.), leaf breadth (L.B.) (cm.), number of hands/bunch (N.H.B.), and number of fingers/hand (N.F.H.). Out of total data for 100 banana plants, data for 80 banana plants is used for "Training" while data for remaining 20 plants is used for "Validation". In order to have a visual idea, a MATLAB plot of variables for first stage growth is exhibited in Figure 2. Fuzzy Logic Toolbox available in MATLAB is used for training ANFIS. A computer program was written in MATLAB and the same is appended as Annexure-I. In view of availability of data at different stages of growth corresponding to different numbers of predictors, is a five input-one output system for first four growth stages of banana and seven input-one output system for last two stages is considered for development of ANFIS model.

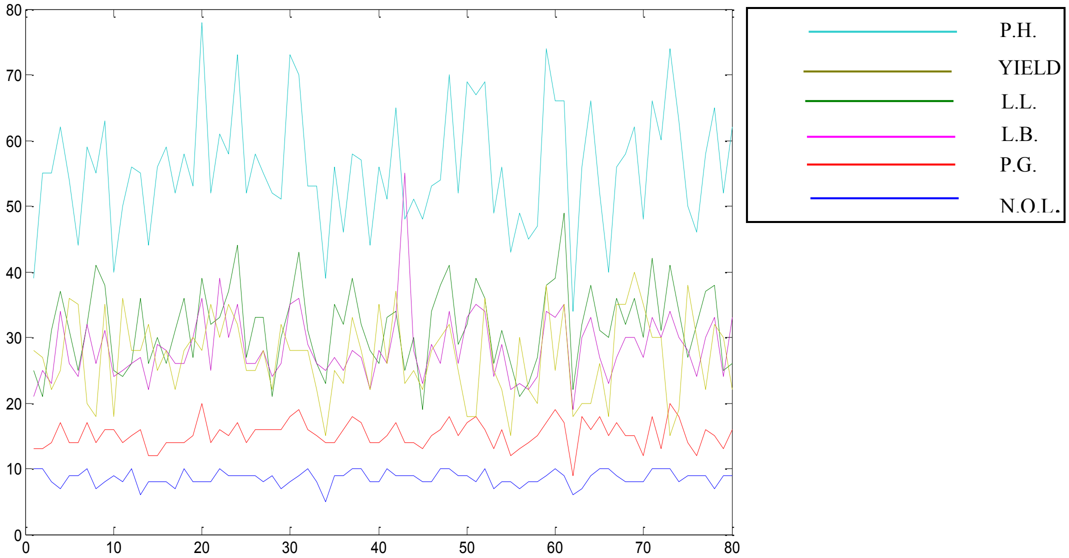


Figure 2: MATLAB Plot of input-output variables in training set for first stage of banana plant.

The input variables are represented with Gaussian membership function. For building ANFIS, membership function of each input was tuned using hybrid method consisting of backpropagation for parameters associated with input membership function (MF) and least-square estimation for parameters associated with output membership functions. Computations of membership function parameters are facilitated by a gradient vector, which provides a measure of how well the FIS system is modelling input/output data. For a given set of parameters, numbers of nodes in training data were found. The numbers of linear parameters and nonlinear parameters were identified. The hypothesized initial number of membership functions and type used for each input were taken as six. Now, hypothesized FIS model is trained to emulate training data by modifying MF parameters according to chosen error criterion. A suitable configuration has to be chosen for best performance of the network. Goal for the error was set to be 0.1 and number of training epochs was given as 200. After training (with 200 epochs) was complete, final configuration for FIS is reported in Table 1. Several two and three-dimensional plots were drawn for proposed ANFIS model, two of which are depicted in Figures 3 and 4. Evidently, the surface is complex and highly nonlinear.

Table1. Fuzzy Information Structure for different stages of banana plant

Stages Fuzzy Information System	I	II	III	IV	V	VI
Number of Inputs	5	5	5	5	7	7
Number of MF for each input	6	6	6	6	6	6
Number of fuzzy rules	30	30	30	30	42	42
Number of linear parameters	140	146	164	184	108	80
Number of nonlinear parameters	300	310	340	360	154	110
Number of training epochs	200	200	200	200	200	200
Number of training data set	80	80	80	80	80	80
Number of test data set	20	20	20	20	20	20
Error goal	0.01	0.01	0.01	0.01	0.01	0.01

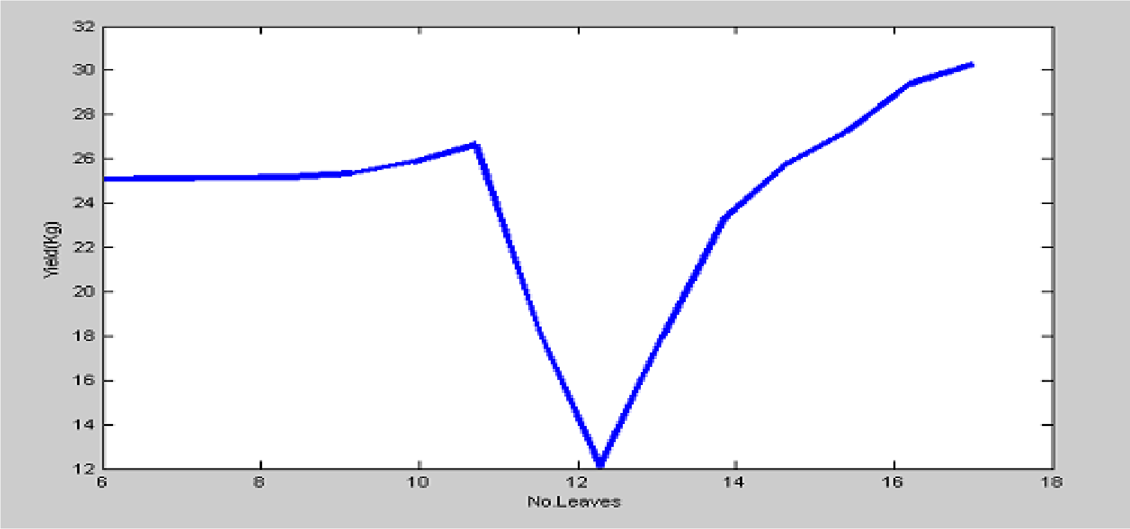


Figure 3: Two-dimensional plot for "second stage" between Yield (Kg.), and Number of leaves

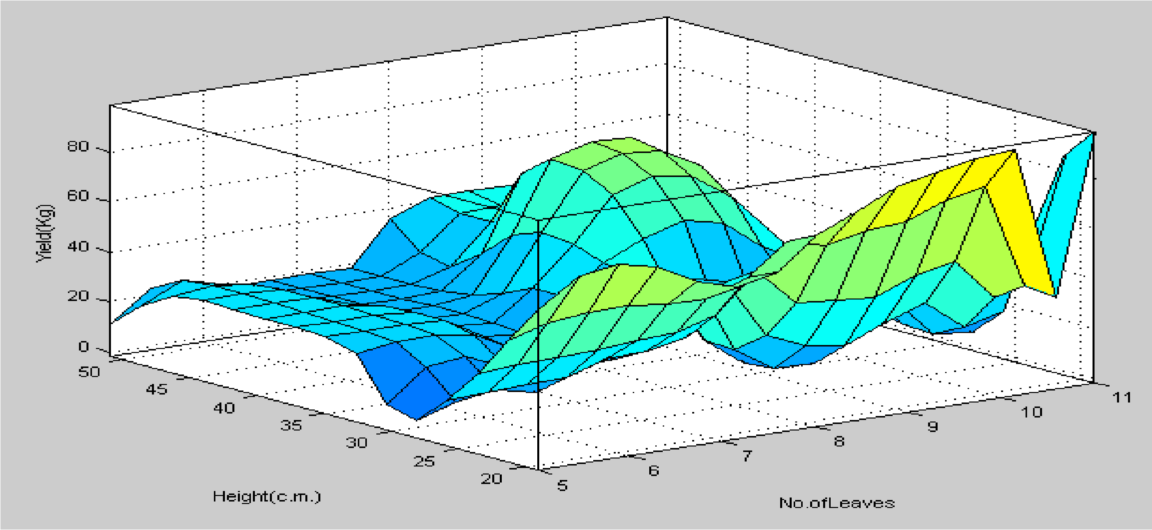


Figure 4: Three-dimensional plot for "second stage" between Yield, Height and Number of leaves

Model was trained for 200 epochs and it was observed that most of the learning was complete in 150 epochs as error goal settles down to almost zero percent at around 150th epochs for all the six stages. Rule viewer for first stage is depicted in Figure 5. Mean square error (MSE) for validation set is computed to compare performance of ANFIS model for different stages and the same is reported in Table 2.

Table 2: Comparison of MSE (Validation data set) for all six stages of growth in banana

Stages / MSE	I	II	III	IV	V	VI
Validation data set	366.04	236.55	102.47	95.35	79.29	39.08

Evidently, MSE decreases as number of stage of plant growth increases, which is quite logical. Further, MSE at first and second stages are very high. As there is a considerable decrease in third stage, banana yield may be forecast at third stage with reasonable accuracy. However, if much more accuracy is required, one would have to wait until sixth stage.

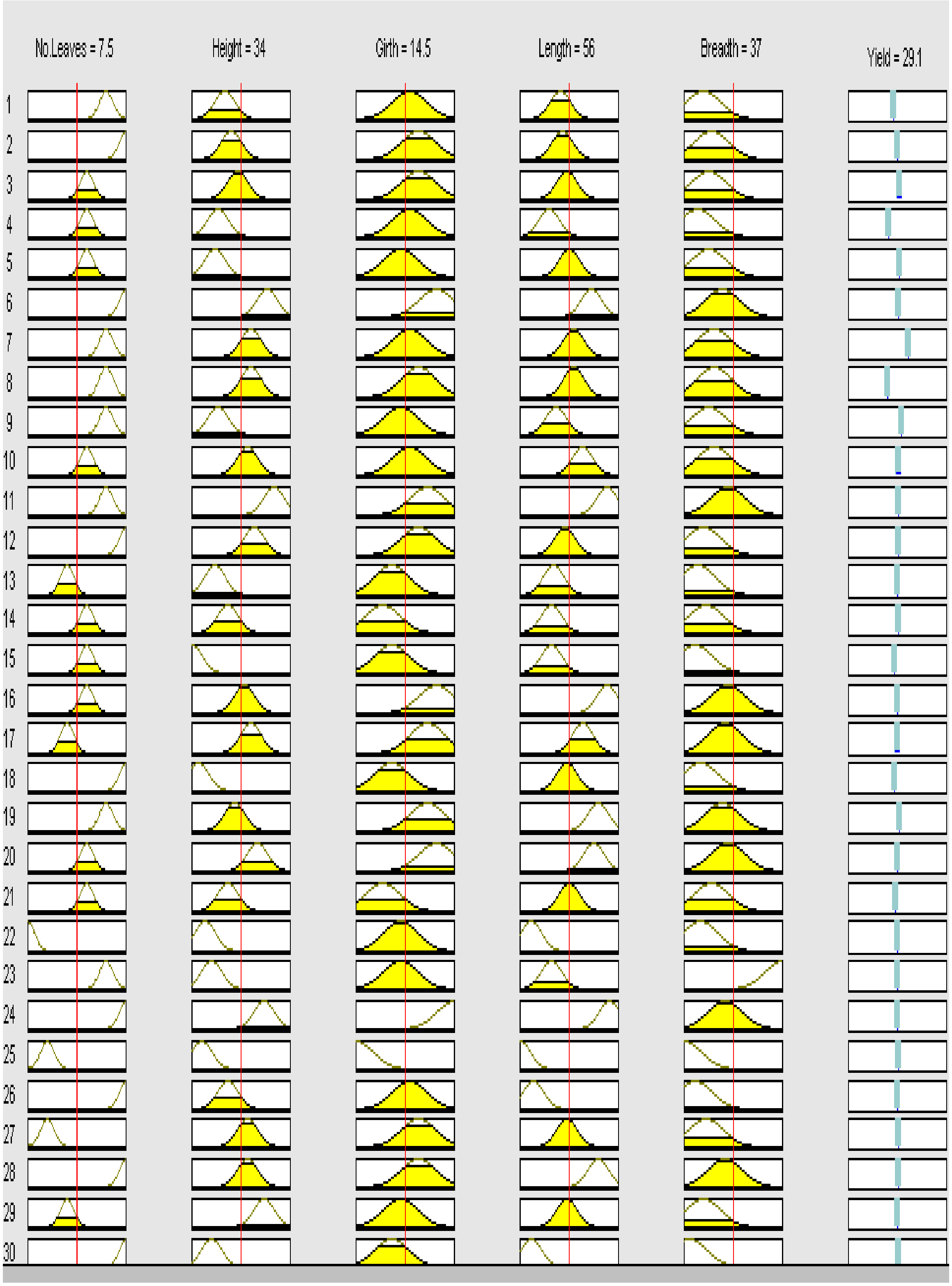


Figure 5: Rule-viewer for membership functions of variables under study in "first stage" of growth in banana plant

Concluding Remarks: Purpose of this work is to highlight the importance of a very powerful and versatile methodology of Neuro-fuzzy modelling and not development of a forecasting model for banana plant yield *per se*. In the illustration considered, we had data of only 100 banana plants. Had data of more number of banana plants say, 500 were available, resultant model would have been more efficient! As a rule of thumb, at least 500 data points are required for training and validation of an ANFIS model efficiently. It is hoped that, in future, research workers would start applying ANFIS Neuro-fuzzy modelling approach to their data sets.

ANNEXURE -I

MATLAB CODES FOR ANFIS

```
load 'stage1.txt';
input=stage1(1:80,1:5);
output=stage1(1:80,6);

input_chk=stage1(81:100,1:5);
out_chk=stage1(81:100,6);
trndata=[input output];
chkdata=[input_chk out_chk];
stepsize = 0.1;

fismat=genfis2(input,output, [.2 .3 .5 .3 .5 .5],[ ],[1.25 .5 .15 0]);
[fismat1, error1, stepsize, fismat2 error2]=anfis(trndata,fismat,[200 0.1 .9 1.1],[1
1 1 1], chkdata);

result=evalfis(input,fismat1);
result1=evalfis(input_chk,fismat1);
plot(output,result,'bd',out_chk, result1,'ks');


plot(result, 'DisplayName', 'result', 'YDataSource', 'result'); figure(gcf)

plot(result, 'DisplayName', 'result', 'YDataSource', 'result'); hold all; plot(output,
'DisplayName', 'output', 'YDataSource', 'output'); hold off; figure(gcf)

plot(result, output, 'DisplayName', 'output vs result', 'XDataSource', 'result',
'YDataSource', 'output'); figure(gcf)

scatter(result, output, 'DisplayName', 'output vs result', 'XDataSource', 'result',
'YDataSource', 'output'); figure(gcf)

plot(output, 'DisplayName', 'output', 'YDataSource', 'output'); hold all; plot(result,
'DisplayName', 'result', 'YDataSource', 'result'); hold off; figure(gcf)

surf(input); figure(gcf)

surfview(fismat)

plot(trndata, 'DisplayName', 'trndata', 'YDataSource', 'trndata'); figure(gcf)
```

SUMMARY

In this thesis concept of Fuzzy regression and Artificial neural network modeling has been thoroughly discussed. Artificial neural network and Fuzzy regression modelling provide attractive ways to capture nonlinearities present in a complex system. Neuro-fuzzy modelling, which is a newly emerging versatile area, is a judicious integration of merits of above mentioned two approaches is also studied. Aplication of the above metioned topics in real data set is dealt. To begin with, in second chapter, theory of fuzzy sets and Possibilistic regression analysis is discussed. Three methods, viz. Minimization, Maximization, and Conjunction are considered. The methodology is applied to employ farmers' estimates at block level for modeling cotton crop yield at block levels of Sirsa district, Haryana. It is found that Conjunction method performed the best.

In third chapter , Possibility and Necessity measures for obtaining reliable fuzzy estimates of crop yield have been thoroughly studied. Estimation of parameters is carried out using "Fuzzy least-squares" procedure. As an illustration, the methodology is applied to Pearl Millet crop yield data in order to build block level estimates for Bhiwani district, Haryana based on farmers' estimates.

In fourth chapter , a modified fuzzy least-squares approach for estimation of parameters is thoroughly studied. Relevant computer program has been developed in SAS/IML software package. Performance evaluation criterion based on “Difference in membership functions” is adopted for computation of error in estimation. As an illustration, the methodology is applied for Pearl Millet crop yield estimation at Block level. Superiority of present approach over Possibility fuzzy least-squares approach is demonstrated for data under consideration.

In fifth chapter , fitting of fuzzy von Bertalanffy growth model is done when response variable is reported in intervals corresponding to various values of explanatory variable. A efficient two-stage procedure proposed by Kao and Chyu, based on fuzzy least squares, is employed. The methodology is thoroughly discussed and, for its application, relevant computer programs are developed in “Nonlinear programming solver LINGO, Version 8” software package. Finally, an illustration to pearl oyster age-length data is discussed.

In sixth chapter, a particular type of Artificial neural network, viz. Multilayered feedforward artificial neural network (MLFANN) is thoroughly studied. In order to train such a network, two types of learning algorithms, namely Gradient descent algorithm (GDA) and Conjugate gradient descent algorithm (CGDA), are described. The methodology is illustrated by considering Maize crop yield data as response variable and Total human labour, Farm power, Fertilizer consumption, and Pesticide consumption as predictors. The data is taken from a recently concluded National Agricultural Technology Project of Division of Agricultural Economics, I.A.R.I., New Delhi. To train the neural network, relevant computer programs are written in MATLAB software package using Neural network toolbox.

In seventh chapter, an important model from this class, viz. Adaptive Neuro-fuzzy inference system (ANFIS) is thoroughly studied. The model is implemented on Fuzzy Logic Toolbox of MATLAB using ANFIS. As an illustration, the methodology is applied for development of a forecasting model for secondary data of yield of 100 banana plants on the basis of data at six different stages of growth using several biometrical characters, like plant height, plant girth, and leaf length as predictors.

lkjka'k

igys v/;k; esa Qt+h fjxzslu ,ao d`f=e ra=dh; usVodZ ekWMfyax dh foLrkj ls ppkZ dh xbZ gSA Qt+h fjxzslu ekWMfyax tfVy iz.kkyh esa mifLFkr ukWu yhfu;fjVh dks vklkuh ls idM+us esa lgk;d gS] bl v/;k; esa U;wjks&Qt+h ekWMfyax dk Hkh v/;;u fd;k x;k gS A mijksDr fof/k;ksa esa okLrfod vk¡dM+ksa ds leqPp;ksa dk Hkh iz;ksx fd;k x;k gS A nwljs v/;k; esa Qt+h leqPp;ksa dh F;ksjh ,ao laHkkO; fjxzslu fo'ys"k.k dh ppkZ dh xbZ gS A rhu fof/k;ksa & feuhekbt+s'ku] eSElhekbt+s'ku] rFkk datD'ku dks viuk;k x;k gSA bl dk;Zfof/k dh gfj;k.kk ds fljlk ftys esa dkWVu Qly dh mit dh ekWMfyax dk fdlkuksa }kjk CykWd&Lrj ij dkeyu djus ds fy, fd;k x;k gS A v/;;u ls ik;k x;k fd datD'ku fof/k loksZre ifj.kke n'kkZrk gS A

rh

ljs v/;k; esa Qly mit ds fy, fo'oluh; Qt+h vkdyu izkIr djus ds fy, fd;s x, iz;klksa dk foLrkj ls v/;;u fd;k x;k gS A Qt+h yhLV&LDos;lZ fof/k ls izkpyksa dk vkdyu fd;k x;k gS A mnkgj.k ds rkSj ij gfj;k.kk ds fHkokuh ftys esa cktjk dh mit vk¡dM+ksa ds vkdyu ds fy, bl dk;Zfof/k dk mi;ksx fd;k x;k gS A pkSFks v/;k; esa izkpyksa ds fy, la'kksf/kr yhLV&LDos;lZ fof/k dk foLrkj ls v/;;u fd;k x;k gS A SAS/IML ds vraxZr dEI;wVj izksxzke fodflr fd;s x;s gSa A =qfV;ksa ds vkdyu dh x.kuk ds fy, ^^fMQjsal bu eSEcjf'ki QD'kal** dks viuk;k x;k gS A mnkgj.k ds rkSj ij dk;Zfof/k dh CykWd&Lrj ij cktjk mit vkdyu ds fy, viuk;k x;k gS A orZeku fof/k dh laHkkfor Qt+h yhLV&LDos;lZ fof/k dh Js"Brk n'kkZ;h xbZ gS A iakpos v/;k; esa Qt+h ukWu cVZySuQh xzksFk ekWMy dks fQV fd;k x;k gS tc jSlik¡l oSfj,scy varjkyksa eas fjiksVZ fd;k x;k gS A Qt+h yhLV&LDos;lZ ij vk/kkfjr] dkvks ,ao P;w }kjk izLrkfor ,d f}Lrjh; fof/k dks lq>k;k x;k gS A bl dk;Zfof/k ij foLrkj ls ppkZ dh xbZ gS rFkk blds mi;ksx ds fy, ^^ukWu yhfu;j izksxzkfeax lkWYoj LINGO Hkkx&8** esa dEI;wVj izksxzke fodflr fd;s x;s gSa rFkk iyZvks;LVj ,atySaFk vk¡dM+ksa dh mnkgj.k ds rkSj ij ppkZ dh xbZ gS A

N

Vs v/;k; esa ,d fo'ks"k izdkj dk d`f=e ra=dh; usVodZ vFkkZr ^^eYVhys;MZ QhMQkWjoMZ vkVhZfQf'k;y U;wjy usVodZ (MLFANN) dk foLrkj ls v/;;u fd;k x;k gS A bl izdkj ds usVodZ ds izf'k{k.k ds fy, nks izdkj ds yfuZax ,YxksfjFke&xzsfM,aV fMlSUV ,YxksfjFke (GDA) rFkk datq,V xzsfM,aV fMlSUV ,YxksfjFke (CGDA) dk o.kZu fd;k x;k gS A bl dk;Zfof/k dks eDdk Qly ds mit vk¡dM+ksa dks jSlik¡l osfj,scy rFkk ekuo {k.k] QkeZ ikoj] mojZd [kir rFkk isLVhlkbM [kir dks izhfMDVj dh rjg fy;k x;k gS A bl v/;;u ds fy, vk¡dM+s us'kuy ,xzhDYpjy VSDuksykth] ifj;kstuk Hkk-d`-v-la] ubZ fnYyh ds vFkZfefr izHkkx ls fy;s x;s gSa A ra=dh; usVodZ ds izf'k{k.k ds fy, MATLAB lkW¶Vos;j iSdst ds varxZr dEI;wVj izksxzke fodflr fd;s x;s gSa A

lkrosa v/;k; esa] ,sMsfIVo U;wjks Qt+h bUQjSUl flLVe (ANFIS) uked ,d egRoiw.kZ ekWMy dk foLrkj ls v/;;u fd;k x;k gS A bl ekWMy dks MATLAB ds Qt+h yksftd VwyckWDl esa ykxw fd;k x;k gS A bl dk;Zfof/k dks dsys ds yxHkx 100 iks/kkSa dh lsdsUMjh mit vk¡dM+ksa ds iwokZuqeku ds fy, fodflr fd;k x;k gS A bl fof/k esa fofHkUu ck;kseSVªhdy y{k.kksa (dSjsDVjksa)] tSls iks/kkSa dh yEckbZ] iks/kkSa dh etcwrh] rFkk ifRr;ksa dh yEckbZ dks izsfMDVlZ ds :i esa iz;ksx fd;k x;k gS A

REFERENCES

Abraham, A., Philip, S., and Mahanti, P.K. (2004). Soft computing models for weather forecasting. *Int. J. Appl. Sci. Compu.*, **11**, 106-17.

Chang, Y. O. and Ayyub, B. M. (2001). Fuzzy regression methods- a comparative assessment. *Fuzz. Set. Syst.*, **119**, 187-203.

Chellam, A. (1988). Growth and biometric relationship of Pearl Oyster *Pinctada fucata* (Gould). *Ind. J. Fish.*, **35,** 1-6.

Cheng, B. and Titterington, D. M. (1994). Neural networks: A review from a statistical perspective. *Statistical Science,* **9,** 2-54.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, **2,** 303-14.

Diamond, P. (1988). Fuzzy least squares. *Inform. Sci.,* **46,** 141–57.

Devaraj, M. (1983). Fish population dynamics. Course manual. *Bull. Cent. Inst. Fish. Edn.*, Mumbai, **3**(10), pp 98.

Dubois, D. and Prade, H. (1980). *Fuzz. Set. and Syst.: Theory and Applications*. Academic Press, New York.

D' Urso, P. (2003). Linear regression analysis for fuzzy/crisp input and fuzzy/crisp output data. *Comp. Stat. Dat. Anal.,* **42**, 47-72.

Ghosh, H. Singh, R . K .and Prajneshu (2007). Possibilistic linear regression analysis with fuzzy response variable for crop yield estimation. (Communicated)

Haykin, S. (2004). *Neural Networks. A Comprehensive Foundation.* New Delhi: Pearson Education.

Heravi, S., Osborn, D. R. and Birchenhall, C. R. (2004). Linear versus neural network forecasts for european industrial production series. *International Journal of Forecasting*, **20**: 435-46.

Hertz, J., Krogh, A. and Palmer, R.G. (1991). *Introduction to the Theory of Neural Computation.* Reading, MA: Addison-Wesley.

Kacprzyk, J. and Fedrizzi, M. (1992). *Fuzzy Regression Analysis*, Omnitech Press, Warsaw.

Kandala, V. M. and Prajneshu (2002). Fuzzy regression methodology for crop yield forecasting using remotely sensed data. *J. Ind. Soc. Rem. Sensing*, **30**, 191-95.

Kandala, V. M. and Prajneshu (2003). Application of fuzzy regression methodology in agriculture. *Ind. J. Agric. Sci.*, **73**, 456-58.

Kandala, V. M. and Prajneshu (2004a). Fuzzy von Bertalanffy growth model for determining age-length relationship. *Ind. J. Fish.,* **51**, 55-59.

Kandala, V. M. and Prajneshu (2004b). Fitting allometric model using fuzzy least squares. *J. Mar. biol. Assoc. Ind.*, **46**, 120-23.

Kao, C. and Chyu, C.L. (2002). A fuzzy linear regression model with better explanatory power. *Fuzz. Set. Syst.,* **126,** 401-09.

Kim, B. and Bishu, R. R. (1998). Evaluation of fuzzy linear regression models by comparing membership functions. *Fuzz. Set. Syst.,* **100,** 343-52.

Klir, G. J. and Yuan, B. (2000). *Fuzzy Sets and Fuzzy Logic: Theory and Application*. 2nd Edn. Prentice Hall of India, New Delhi.

Kukolj, d. (2002). Design of adaptive Takagi-Sugeno-Kang fuzzy models. *Appl. Soft comput.*, **2**, 89-103.

LINDO Systems Inc. (2002). *LINGO user's guide.* LINDO Systems Inc., Chicago, U.S.A.

McCulloch, W. S., and Pitts, W. (1943). “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics*, **5**, 115-33.

Matlab® (2006). *Neural Network Toolbox User’s Guide.* U.S.A.: The Math Works, Inc.

Modarres, M., Nasrabadi, E. and Nasrabadi, M. M. (2004). Fuzzy linear regression analysis from the point of view risk. *Int. J. Uncert., Fuzzi. Know. Bas. Sys.*, **12**, 635-49.

Modarres, M., Nasrabadi, E. and Nasrabadi, M. M. (2005). Fuzzy linear regression models with least square errors. *Appl. Maths. Compu.*, **163**, 977-89.

Piegat, A. (2001). *Fuzzy Modeling and Control,* Physica-Verlag, Heidelberg, New York.

Quinn, T. J. II and R. B. Deriso (1999). *Quantitative Fish Dynamics*, Oxford Univ. Press, U.S. A.

Rao, J.N.K. (2003). *Small area estimation*. Wiley Interscience, U.S.A.

Rao, V. N. M. (2005). *Banana.* Directorate of information and publications of agriculture, Indian council of Agricultural Research, New Delhi. Pp. 69.

Rutkowaska, D. (2002). *Neuro-fuzzy Architectures and Hybrid Learning*. Physica-Verlag, U.S.A.

SAS (2006). *SAS/IML User’s Guide*. SAS Institute, North Carolina.

Sengupta, K. Nandi, S. and Chakroborty, N. (2001) Effect of sulphur-containing fertilizers on productivity of rainfed greengram (phaseolus radiatus). *Ind. J. of Agril. Sci.*, **71**, 408-10.

Singh, R. K., Ghosh, H., and Prajneshu (2007). Possibility and necessity measures for fuzzy linear regression analysis (Communicated).

Singh, R. K. and Prajneshu (2007). Artificial neural network methodology for modelling and forecasting maize crop yield. (Communicated).

Singh, R. P., Kumar, R. Singh, B. B. Awasthi, P. K. Atibudhi, H. N. Chahal, S. S. Varghese, K. A. Singh, R. K. and Maurya, S. P. (2004). *Technological Change and Production Performance in Irrigated Maize-based Agro-ecosystem : The Interplay of Economic, Technological and*

Institutional Factors. New Delhi: Division of Agricultural Economics, IARI, Research Report 2004-01, pp. 1-107.

Sud, U.C., Mathur, D.C., Sethi, S.C., Jha, G.K. and Bhasin, R.M. (2006). *Crop yield estimation at smaller area level using farmers' estimates.* I.A.S.R.I. Project report, New Delhi, pp. 1-72.

Tanaka, H. (1987). Fuzzy data analysis by Possibilistic linear model. *Fuzz. Set. Syst.,* **24,** 363-75.

Tanaka, H. Hayashi, I. and Watada, J. (1989). Possibilistic linear regression analysis for fuzzy data. *Eur. J. Operat. Res.,* **40,** 389-96.

Tanaka, H. Uejima, S. and Asia, K. (1982). Linear regression analysis with fuzzy model. *IEEE Trans. Syst. Man. Cybernet.,* **12,** 903-07.

Venugopalan, R. and Samasundaran, K.S. (2005) Statistical model for evolving crop-logging technique in banana. *Trop. Agril.,* **82**, 23-27.

Warner, B. and Misra, M. (1996). Understanding neural networks as statistical tools. *American Statistician,* **50,** 284-93.

Walde, J. F., Tappeiner, G. Tappeiner, U. Tasser, E. and Holub, H. W. (2004). Statistical aspects of multilayer perceptrons under data limitations, *Computional Statistics & Data Analysis,* **46**,173-88.

Zadeh, L. A. (1965). Fuzzy sets. *Inform. Con.,* **8**, 338-53.

Zhang, G. P. (2007). Avoiding pitfalls in neural network research. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, **37,** 3-16.

Zhang, G., Patuwo, B. E. and Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting,* **14**, 35-62.