Improved Software Development Effort Estimation Based on Four Fuzzy Logic Functions

Rshma Chawla, Deepak Ahlawat

1Assistant Professor at MMICTBM, MMU Mullana
2MPhil Research Scholar at MMICTBM, MMU Mullana

Email: 1 rshma.chawla@mmumullana.org, 2deepakahlawat1983@gmail.com

Abstract: The systems and software development industry is characterized by a paradigm of project failure. One of the known contributing causes of these project failures is poor requirements engineering and management, which has been repeatedly and widely discussed and documented. But there are other factors also like poor software project management practices, poor design strategy and inefficient testing principles also contributing to project failures. A fuzzy model is more pertinent when the systems are inadequate for analysis by conventional means or when the available data is uncertain, inaccurate or vague.

In this paper, software development effort estimation using Fuzzy Triangular Membership Function, Gbell Membership Function, Gauss2 Membership Function and Trapezoidal Membership Function is implemented using Mamdani Type Fuzzy inference system of Fuzzy Logic Toolbox Software of Matlab R2013a and the results of these membership functions are compared with each other and with COCOMO model. It is found that the Fuzzy Logic Model using Gaussian2 Membership Function provided best results.

Keywords: Software Engineering, Software Effort Estimation, Fuzzy Logic, COCOMO Model

I. INTRODUCTION

1.1. Software Development Effort Estimation

Software metric and especially software estimation is based on measuring of software attributes which are typically related to the product, the process and the resources of software development [1]. This kind of measuring can be used as parameters in project management models [2] which provide assessments to software project managers in managing software projects to avoid problems such as cost overrun and behind the schedule. One of the most widely researched areas of software measurement is software effort estimation. Software effort estimation models divided into two main categories: algorithmic and non-algorithmic. The most popular algorithmic estimation models include Boehm’s COCOMO [3], Putnam’s SLIM [4] and Albrecht’s Function Point [5]. These models require as inputs, accurate estimate of certain attributes such as line of code (LOC), complexity and so on which are difficult to obtain during the early stage of a software development project. The models also have difficulty in modelling the inherent complex relationships between the contributing factors, are unable to handle categorical data as well as lack of reasoning capabilities [6]. The limitations of algorithmic models led to the exploration of the non-algorithmic techniques which are soft computing based. These include artificial neural network, evolutionary computation, fuzzy logic models, case-based reasoning and combinational models and so on. Artificial neural network is used in effort estimation due to its ability to learn from previous data [7][8]. It is also able to model complex relationships between the dependent (effort) and independent variables (cost drivers) [7][8]. In addition, it has the ability to generalise from the training data set thus enabling it to produce an acceptable result for previously unseen data. Most of the work in the application of neural network to effort estimation made use of feed-forward multi-layer Perceptron, Backpropagation algorithm and sigmoid function [7]. Selecting good models for software estimation is very critical for software engineering. In the recent years many software estimation models have been developed [4, 5, 6, 7, 8, 9]. Gray and MacDonell compared function point analysis, regression techniques, feed-forward neural network and fuzzy logic in software effort estimation. Their results showed that fuzzy logic model achieved good performance, being outperformed in terms of accuracy only by neural network model with considerably more input variables. Also they developed FULSOME (Fuzzy Logic for Software Metrics) which is a set of tools that helps in creating fuzzy model. Fei and Lui [10] introduced the f-COCOMO model which applied fuzzy logic to the COCOMO model for software effort estimation. Since there was no comparison of the results between the f-COCOMO and other estimation models in their study, the estimation capability of the former is unknown. Roger [11] also proposed a fuzzy COCOMO model which adopted the fuzzy logic method to model the uncertainty of software effort drivers, but the effectiveness of the proposed model is not mentioned. Idri[7, 8] further defined a fuzzy set for the linguistic values of each effort driver with a trapezoid-shaped membership function for the fuzzy COCOMO model. The effort multipliers in the original COCOMO model
model were obtained from the fuzzy sets. This fuzzy COCOMO model was less sensitive to the software effort drivers as compared to the intermediate COCOMO81. In 2004, Xue and Khoshgoftaar[13] presented a fuzzy identification effort estimation modeling technique to deal with linguistic effort drivers, and automatically generated the fuzzy membership functions and rules by using the COCOMO81 database. The proposed fuzzy identification model provided significantly better effort estimates than the original three COCOMO models, i.e., basic, intermediate, and detailed.

1.2. Fuzzy Logic Approach

Since fuzzy logic foundation by LotfiZadeh in 1965, it has been the subject of important investigations [12]. It is a mathematical tool for dealing with uncertainty and also it provides a technique to deal with imprecision and information granularity [11]. The fuzzy logic model uses the fuzzy logic concepts introduced by LotfiZadeh [12]. Fuzzy reasoning consists of three main components [11, 12, 13, 14]: fuzzification process, inference from fuzzy rules, and defuzzification process. Fuzzification process is where the objective term is transformed into a fuzzy concept. The membership functions are applied to the actual values of variables to determine the confidence factor or membership function (MF). Fuzzification allows the input and output to be expressed in linguistic terms. Inference involves defuzzification of the conditions of the rules and propagation of the confidence factors of the conditions to the conclusion of the rules. A number of rules will be fired and the inference engine assigns the particular outcome with the maximum membership value from all the fired rules.

1.3. Parameters Analysis

The main parameter for the evaluation of cost estimation models is the Magnitude of Relative Error (MRE) [13] which is defined as follows:

\[ \text{Magnitude Relative Error (RE)} = \frac{|E - \hat{E}|}{E} \]

Where \( E \) = Estimated Effort, \( \hat{E} \) = Actual Effort.

The MRE value is calculated for each observation whose effort is predicted. The aggregation of MRE over multiple observations (N), can be achieved through the Mean MRE (MMRE) as follows:

\[ \text{MRE} = \frac{1}{N}(\text{Magnitude Relative Error}) \]

II. FUZZY IDENTIFICATION

A fuzzy model [20, 8] is used when the systems are not suitable for analysis by conventional approach or when the available data is uncertain, inaccurate or vague [15]. The point of fuzzy logic is to map an input space to an output space using a list of if-then statements called rules. All rules are evaluated in parallel, and the order of the rules is unimportant. For writing the rules, the inputs and outputs of the system are to be identified. To obtain [21] a fuzzy model from the data available, the steps to be followed are,

- Select a Mamdani type Fuzzy Inference system.
- Define the input variables and output variable effort.
- Set the type of the membership functions (TMF or Gaussian or Trapezoidal) for input variables.
- The data is now translated into a set of if-then rules written in Rule editor.
- A certain model structure is created, and parameters of input and output variables can be tuned to get the desired output.

2.1 Fuzzy Approach for Prediction of Effort

The Intermediate COCOMO model data is used for developing the Fuzzy Inference System (FIS) [10]. The inputs to this system are MODE and SIZE. The output is Fuzzy Nominal Effort. The framework [16] is shown in “Fig. 1”.

![Figure 1: Fuzzy Framework.](Image)

Fuzzy approach [17] specifies the SIZE of a project as a range of possible values rather than a specific number. The MODE of development is specified as a fuzzy range. The advantage of using the fuzzy ranges [18] is that we will be able to predict the effort for projects that do not come under a precise mode i.e. comes in between 2 modes. This situation cannot be handled using the COCOMO. The output of this FIS is the Fuzzy Nominal Effort. The Fuzzy Nominal Effort multiplied by the EAF gives the Estimated Effort. The FIS [19] needs appropriate membership functions and rules.

2.2 Fuzzy Rules

Our rules are based on the fuzzy sets of MODE, SIZE and EFFORT appears in the following form:

- If MODE is Organic and SIZE is S1 then EFFORT is EF1
- If MODE is Semidetached and SIZE is S1 then EFFORT is EF2
- If MODE is Embedded and SIZE is S1 then EFFORT is EF3
- If MODE is Organic and SIZE is S2 then EFFORT is EF4
- If MODE is Semidetached and SIZE is S2 then EFFORT is EF5
- If MODE is Embedded and SIZE is S3 then EFFORT is EF5
- If MODE is Embedded and SIZE is S4 then EFFORT is EF3
- If MODE is Organic and SIZE is S3 then EFFORT is EF4
If MODE is Embedded and SIZE is S5 then EFFORT is EF6
If MODE is Organic and SIZE is S4 then EFFORT is EF4

This is represented in MATLAB as shown in figure below:

Figure 2: Fuzzy Rules

2.3 Membership Functions

A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse, a fancy name for a simple concept. One of the most commonly used examples of a fuzzy set is the set of tall people. In this case the universe of discourse is all potential heights, say from 3 feet to 9 feet, and the word tall would correspond to a curve that defines the degree to which any person is tall. If the set of tall people is given the well-defined (crisp) boundary of a classical set, we might say all people taller than 6 feet are officially considered tall. But such a distinction is clearly absurd. It may make sense to consider the set of all real numbers greater than 6 because numbers belong on an abstract plane, but when we want to talk about real people, it is unreasonable to call one person short and another one tall when they differ in height by the width of a hair.

Fuzzy Logic Membership Functions used in Fuzzy Logic toolbox.

1. **Trimf** - Triangular-shaped built-in membership function

   Syntax: \( y = \text{trimf}(x, \text{params}) \)
   \( y = \text{trimf}(x, [a \ b \ c]) \)

   Description: The triangular curve is a function of a vector, \( x \), and depends on three scalar parameters \( a \), \( b \), and \( c \), as given by
   \[ f(x; a, b, c) = \max\left( \min\left( \frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \]

2. **Gbellmf** - Generalized bell-shaped built-in membership function

   Syntax: \( y = \text{gbellmf}(x, \text{params}) \)

   Description: The generalized bell function depends on three parameters \( a \), \( b \), and \( c \) as given by
   \[ f(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \]

   Where the parameter \( b \) is usually positive. The parameter \( c \) locates the center of the curve. Enter the parameter vector \( \text{params} \), the second argument for \( \text{gbellmf} \), as the vector whose entries are \( a \), \( b \), and \( c \), respectively.

3. **Trapmf** - Trapezoidal-shaped built-in membership function

   Syntax: \( y = \text{trapmf}(x, [a \ b \ c \ d]) \)

   Description: The trapezoidal curve is a function of a vector, \( x \), and depends on four scalar parameters \( a \), \( b \), \( c \), and \( d \), as given by
   \[ f(x; a, b, c, d) = \max\left( \min\left( \frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right) \]

   The parameters \( a \) and \( d \) locate the "feet" of the trapezoid and the parameters \( b \) and \( c \) locate the "shoulders."

4. **Gauss2mf** - Two-sided Gaussian membership function.

   Syntax: \( y = \text{gauss2mf}(x, [\text{sig1} \ \text{c1} \ \text{sig2} \ \text{c2}]) \)

   Description: The Gaussian function depends on two parameters \( \text{sig1} \) and \( \text{c1} \) for the left-most curve. The second function specified by \( \text{sig2} \) and \( \text{c2} \) determines the shape of the right-most curve. Whenever \( \text{c1} < \text{c2} \), the gauss 2mf function reaches a maximum value of 1. Otherwise, the maximum value is less than one. The parameters are listed in the order:
   \([\text{sig1}, \text{c1}, \text{sig2}, \text{c2}]\).

III. EXPERIMENTAL RESULTS

Experiments were done by taking original data from COCOMO dataset [14]. The software development efforts obtained when using COCOMO and other membership functions were observed. After analyzing the results attained by means of applying COCOMO, trapezoidal MF for cost drivers, and Gaussian MF for both size and cost drivers together, it is observed that the effort estimation of the proposed model is giving more precise results than the other models.

COCOMO used Mamdami FIS method due to its intuitive, widespread acceptance and well suited for human input nature. Figure 3 show the fuzzification of
cost attributes using MATLAB.

![Figure 3: Fuzzification of various cost using FIS tool in the MATLAB software.](image)

**Table 1** - Comparison between obtained results from COCOMO 81 and FL-COCOMO in terms of MMRE

<table>
<thead>
<tr>
<th>MODEL</th>
<th>MMRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic COCOMO 81</td>
<td>0.60197</td>
</tr>
<tr>
<td>Intermediate COCOMO 81</td>
<td>0.18889</td>
</tr>
<tr>
<td>Detailed COCOMO 81</td>
<td>0.18829</td>
</tr>
<tr>
<td>FL-COCOMO (Using Triangular function)</td>
<td>0.2454</td>
</tr>
<tr>
<td>FL-COCOMO (Using Trapezoidal function)</td>
<td>0.1953</td>
</tr>
<tr>
<td>FL-COCOMO (Using Gaussian2 function)</td>
<td>0.1799</td>
</tr>
<tr>
<td>FL-COCOMO (Using GBell function)</td>
<td>0.1832</td>
</tr>
</tbody>
</table>

The effort estimated by means of fuzzifying size and cost drivers together and using Gaussian MF is yielding better estimate which is very nearer to the actual effort. Therefore, using fuzzy sets, size and cost drivers of a software project can be specified by distribution of its possible values, by means of which we can evaluate the associated imprecision residing within the final results of cost estimation.

**IV. CONCLUSION AND FUTURE RESEARCH**

This research work is to provide a technique for software cost estimation that performs better than other techniques on the accuracy of effort estimation.

In this research an improved approach to software project effort is projected by the use of fuzzy sets rather than classical intervals in the COCOMO model. This study explores four fuzzy logic membership functions Fuzzy Triangular Membership Function, GBell Membership Function, Gaussian2 Membership Function, and Trapezoidal Membership Function is implemented and compared with COCOMO. The Gaussian2 Membership Function used in this research has shown good results by handling the imprecision in inputs quite well and also their ability to adapt further make them a valid choice to represent fuzzy sets.

Mean Relative Error shows the comparison between Fuzzy Membership Functions. Trapezoidal Membership Function has highest Mean Relative Error this implies it has lowest accuracy. Mean Magnitude Relative Error of Gaussian2 Membership Function shows better software effort estimates as compared to the traditional COCOMO. Lower the MMRE better is the prediction accuracy of the model. The above research work can be analyzed in terms of feasibility and acceptance in the industry. It can be deployed on COCOMO II environment with experts providing required information for developing fuzzy sets and an appropriate rule base.

**REFERENCES**


ISSN (Print): 2319-2526, Volume -4, Issue -2, 2015