Application of Fuzzy Logic Approach to Software Effort Estimation

Prasad Reddy P.V.G.D
Dept. of CSSE
Andhra University
Visakhapatnam, INDIA

Sudha K. R
Dept. of EE
Andhra University
Visakhapatnam, INDIA

Rama Sree P
Dept. of CSSE
Aditya Engineering College, JNTUK
Kakinada, INDIA

Abstract— The most significant activity in software project management is Software development effort prediction. The literature shows several algorithmic cost estimation models such as Boehm's COCOMO, Albrecht's Function Point Analysis, Putnam's SLIM, ESTIMACS etc., but each model do have their own pros and cons in estimating development cost and effort. This is because project data, available in the initial stages of project is often incomplete, inconsistent, uncertain and unclear. The need for accurate effort prediction in software project management is an ongoing challenge. A fuzzy model is more apt when the systems are not suitable for analysis by conventional approach or when the available data is uncertain, inaccurate or vague. Fuzzy logic is a convenient way to map an input space to an output space. Fuzzy Logic is based on fuzzy set theory. A fuzzy set is a set without a crisp, clearly defined boundary. It is characterized by a membership function, which associates with each point in the fuzzy set a real number in the interval [0, 1], called degree or grade of membership. The membership functions may be Triangular, GBell, Gauss and Trapezoidal etc. In the present paper, software development effort prediction using Fuzzy Triangular Membership Function and GBell Membership Function is implemented and compared with COCOMO. A case study based on the NASA93 dataset compares the proposed fuzzy model with the Intermediate COCOMO. The results were analyzed using different criterions like VAF, MARE, VARE, MMRE, Prediction and Mean BRE. It is observed that the Fuzzy Logic Model using Triangular Membership Function provided better results than the other models.

Keywords: Development Effort; EAF; Cost Drivers; Fuzzy Identification; Membership Functions; Fuzzy Rules; NASA93 dataset.

I. INTRODUCTION

In algorithmic cost estimation [1], costs and efforts are predicted using mathematical formulae. The formulae are derived based on some historical data [2,19]. The best known algorithmic cost model called COCOMO (CONstructive COST MOdel) was published by Barry Boehm in 1981[3]. It was developed from the analysis of sixty three (63) software projects. Boehm projected three levels of the model called Basic COCOMO, Intermediate COCOMO and Detailed COCOMO [3, 5]. In the present paper we mainly focus on the Intermediate COCOMO.

A. Intermediate COCOMO

The Basic COCOMO model [3] is based on the relationship: Development Effort, \( DE = a \times (SIZE)^b \); where, SIZE is measured in thousand delivered source instructions. The constants a, b are dependent upon the ‘mode’ of development of projects. DE is measured in man-months. Boehm proposed 3 modes of projects [3]:

a) Organic mode – simple projects that engage small teams working in known and stable environments.

b) Semi-detached mode – projects that engage teams with a mixture of experience. It is in between organic and embedded modes.

c) Embedded mode – complex projects that are developed under tight constraints with changing requirements.

The accuracy of Basic COCOMO is limited because it does not consider the factors like hardware, personnel, use of modern tools and other attributes that affect the project cost. Further, Boehm proposed the Intermediate COCOMO[3,4] that adds accuracy to the Basic COCOMO by multiplying ‘Cost Drivers’ into the equation with a new variable: EAF (Effort Adjustment Factor) shown in Table I.

<table>
<thead>
<tr>
<th>Development mode</th>
<th>Intermediate Effort Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organic</td>
<td>( DE = EAF \times 3.2 \times (SIZE)^{1.05} )</td>
</tr>
<tr>
<td>Semi-detached</td>
<td>( DE = EAF \times 3.0 \times (SIZE)^{1.12} )</td>
</tr>
<tr>
<td>Embedded</td>
<td>( DE = EAF \times 2.8 \times (SIZE)^{1.2} )</td>
</tr>
</tbody>
</table>

The EAF term is the product of 15 Cost Drivers [5, 11] that are listed in Table II. The multipliers of the cost drivers are Very Low, Low, Nominal, High, Very High and Extra High. For example, for a project, if RELY is Low, DATA is High, CPLX is extra high, TIME is Very High, STOR is High and rest parameters are nominal then \( EAF = 0.75 \times 1.08 \times 1.65 \times 1.30 \times 1.06 \times 1.0 \). If the category values of all the 15 cost drivers are “Nominal”, then \( EAF = 1 \).

<table>
<thead>
<tr>
<th>S. No</th>
<th>Cost Driver Symbol</th>
<th>Very low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very high</th>
<th>Extra high</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RELY</td>
<td>0.75</td>
<td>0.88</td>
<td>1.00</td>
<td>1.15</td>
<td>1.40</td>
<td>—</td>
</tr>
</tbody>
</table>
The 15 cost drivers are broadly classified into 4 categories [3, 5].

- **Product**: RELY - Required software reliability
  DATA - Data base size
  CPLX - Product complexity
- **Platform**: TIME - Execution time
  STOR - Main storage constraint
  VIRT - Virtual machine volatility
  TURN - Computer turnaround time
- **Personnel**: ACAP - Analyst capability
  AEXP - Applications experience
  PCAP - Programmer capability
  VEXP - Virtual machine experience
  LEXP - Language experience
- **Project**: MODP - Modern programming
  TOOL - Use of software tools
  SCED - Required development schedule

Depending on the projects, multipliers of the cost drivers will vary and thereby the EAF may be greater than or less than 1, thus affecting the Effort [5].

II. FUZZY IDENTIFICATION

A fuzzy model [13,15] is used when the systems are not suitable for analysis by conventional approach or when the available data is uncertain, inaccurate or vague [7]. 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 a fuzzy model from the data available, the steps to be followed are,

- Select a Sugeno type Fuzzy Inference System.
- Define the input variables mode, size and output variable effort.
- Set the type of the membership functions (TMF or GBellMF) for input variables.

- Set the type of the membership function as linear for output variable.
- 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.

A. 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 [8] is shown in “Fig. 1”.

![Fuzzy Framework](image)

Fuzzy approach [9] 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 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 needs appropriate membership functions and rules.

B. Fuzzy Membership Functions

A membership function (MF) [9, 12] 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 also called as the universe of discourse. For our problem, we have used 2 types of membership functions:

1) Triangular membership function
2) Generalized Bell membership function

Triangular membership function (TMF):

It is a three-point function [17], defined by minimum \((\alpha)\), Maximum \((\beta)\) and modal \((m)\) values, that is, TMF \((\alpha, m, \beta)\), where \((\alpha \leq m \leq \beta)\). The parameters \(\alpha\) and \(\beta\) locate the "feet" of the triangle and the parameter \(m\) locates the peak. The triangular curve is a function of a vector, \(x\), and depends on three scalar parameters \(\alpha\), \(m\), and \(\beta\) as given by

\[
f(x; \alpha, m, \beta) = \max \left( \min \left( \frac{x - \alpha}{m - \alpha}, \frac{\beta - x}{\beta - m} \right), 0 \right)
\]

Please refer to “Fig. 2” for a sample triangular membership function.
The fuzzy set definitions for the MODE of development appear in “Fig. 3” and the fuzzy set [8] for SIZE appear in “Fig. 4”.

**Figure 3: Fuzzy Set for Mode**

**Figure 4: Fuzzy set for SIZE**

**Generalized Bell Membership Function (GBellMF):**

It is a three-point function, defined by minimum ($\alpha$), maximum ($\beta$) and modal ($m$) values, that is, $\text{GBellMF}(\alpha, m, \beta)$, where ($\alpha \leq m \leq \beta$). Please refer to “Fig. 5” for a sample Generalized Bell membership function. The generalized bell function depends on three parameters $\alpha$, $m$, and $\beta$ as given by

$$f(x; \alpha, m, \beta) = \frac{1}{1 + \left|\frac{x - \beta}{\alpha}\right|^{2m}}$$

(2)

where the parameter $m$ is usually positive. The parameter $\beta$ locates the center of the curve.

**Figure 5: A Sample Generalized Bell Membership Function**

We can get the Fuzzy sets for MODE, SIZE and Effort for GBellMF in the same way as in triangular method, but the difference is only in the shape of the curves.

**C. Fuzzy Rules**

Our rules based on the fuzzy sets [9, 20] 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 s1 then EFFORT is EF5
If MODE is embedded and SIZE is s2 then EFFORT is EF5
If MODE is embedded and SIZE is s3 then EFFORT is EF6
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

.....

**III. VARIOUS CRITERION FOR ASSESSMENT OF SOFTWARE EFFORT**

**a) Variance Accounted For (VAF)**

$$\text{VAF} \% = \left(1 - \frac{\text{var}(E - \hat{E})}{\text{var}E}\right) \times 100$$

(3)

**b) Mean Absolute Relative Error (MARE)**

$$\text{MARE} \% = \frac{\sum f(R_E)}{\sum f} \times 100$$

(4)

**c) Variance Absolute Relative Error (VARE)**

$$\text{VARE} \% = \frac{\sum f(R_E - \text{meanR}_E)^2}{\sum f} \times 100$$

(5)

**d) Prediction (n)**

Prediction at level $n$ is defined as the % of projects that have absolute relative error less than $n$.

**e) Balance Relative Error (BRE)**

$$\text{BRE} = \frac{|E - \hat{E}|}{\min(E, \hat{E})}$$
Where, $E = \text{Estimated effort} \quad \hat{E} = \text{Actual effort}$

Absolute Relative Error (RE) $= \frac{|E - \hat{E}|}{|E|}$

(7)

f) Mean Magnitude of Relative Error (MMRE)

$$\text{MMRE} = \frac{1}{N} \sum_{i=1}^{N} \frac{MRE_i \times 100}{(8)}$$

Where $MRE = \frac{|\hat{E} - E|}{E}$

(9)

$N = \text{No. of Projects}, \quad E = \text{Estimated effort}, \quad \hat{E} = \text{Actual effort}$

A model which gives higher VAF is better than that which gives lower VAF. A model which gives higher Pred(n) is better than that which gives lower Pred(n). A model which gives lower MARE is better than that which gives higher MARE [11, 12]. A model which gives lower VARE is better than that which gives higher VARE [6]. A model which has lesser MMRE is better than the model which has higher MMRE. A model which gives lower BRE is better than that which gives higher BRE. A model which gives lower MMRE is better than that which gives higher MMRE.

IV. EXPERIMENTAL STUDY

In carrying out our experiments, we have chosen 93 projects of the NASA 93 dataset. The estimated efforts using Intermediate COCOMO, Fuzzy using TMF and GBellMF are shown in Table III. Table IV and “Fig. 6” to “Fig. 12” shows the comparisons of various models basing on different criterions.

### TABLE III. ESTIMATED EFFORT IN MAN MONTHS OF VARIOUS MODELS

<table>
<thead>
<tr>
<th>Project ID</th>
<th>Actual Effort</th>
<th>Estimated Effort using</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>COCOMO</td>
</tr>
<tr>
<td>1</td>
<td>8.40</td>
<td>2.30</td>
</tr>
<tr>
<td>9</td>
<td>36.00</td>
<td>27.80</td>
</tr>
<tr>
<td>12</td>
<td>42.00</td>
<td>32.30</td>
</tr>
<tr>
<td>13</td>
<td>42.00</td>
<td>35.40</td>
</tr>
<tr>
<td>17</td>
<td>50.00</td>
<td>36.30</td>
</tr>
<tr>
<td>19</td>
<td>60.00</td>
<td>50.60</td>
</tr>
<tr>
<td>26</td>
<td>72.00</td>
<td>33.00</td>
</tr>
<tr>
<td>36</td>
<td>120.00</td>
<td>116.70</td>
</tr>
<tr>
<td>41</td>
<td>192.00</td>
<td>574.20</td>
</tr>
<tr>
<td>45</td>
<td>239.00</td>
<td>224.70</td>
</tr>
<tr>
<td>49</td>
<td>300.00</td>
<td>290.50</td>
</tr>
<tr>
<td>53</td>
<td>352.80</td>
<td>290.50</td>
</tr>
<tr>
<td>59</td>
<td>420.00</td>
<td>436.90</td>
</tr>
<tr>
<td>61</td>
<td>432.00</td>
<td>615.50</td>
</tr>
<tr>
<td>68</td>
<td>576.00</td>
<td>821.10</td>
</tr>
<tr>
<td>77</td>
<td>882.00</td>
<td>1332.30</td>
</tr>
<tr>
<td>81</td>
<td>1248.00</td>
<td>1113.60</td>
</tr>
<tr>
<td>83</td>
<td>1368.00</td>
<td>1139.60</td>
</tr>
<tr>
<td>92</td>
<td>4560.00</td>
<td>24726.50</td>
</tr>
</tbody>
</table>

### REFERENCES

Referring to Table 4, we see that Fuzzy using TMF yields better results for maximum criterions when compared with the other methods. Thus, basing on VAF, MMRE, VARE, MARE, Pred(30) & Mean BRE, we come to a conclusion that the Fuzzy method using TMF (triangular membership function) is better than Fuzzy method using GBellMF or Intermediate COCOMO. It is not possible to evolve a method, which can give 100 % VAF. By suitably adjusting the values of the parameters in FIS we can optimize the estimated effort.

V. CONCLUSION

Referring to Table 4, we see that Fuzzy using TMF yields better results for maximum criterions when compared with the other methods. Thus, basing on VAF, MMRE, VARE, MARE, Pred(30) & Mean BRE, we come to a conclusion that the Fuzzy method using TMF (triangular membership function) is better than Fuzzy method using GBellMF or Intermediate COCOMO. It is not possible to evolve a method, which can give 100 % VAF. By suitably adjusting the values of the parameters in FIS we can optimize the estimated effort.

REFERENCES


