Fuzzy Based Software Cost Estimation Methods: A Comparative Study

Mizbah Fatima
Lecturer
Faculty of Computer Science and Information Technology
Integral University, Lucknow, India

Sheikh Fahad Ahmad
Assistant Professor
Faculty of Computer Science and Information Technology
Integral University, Lucknow, India

Mustafa Hasan
Software Engineer
Aricent Pvt Ltd, Gurgaon, India

Abstract

Software cost estimation is the process of predicting the effort required to develop a software system. Many estimation models have been proposed over the last 30 years. With the development of software engineering, estimation of project cost and duration has been a very important work. It plays an important role in project bid and project planning. Many papers have been published regarding this topic, which aims at predicting costs of projects to a tolerable degree of accuracy at the early stage. In this paper, several existing fuzzy logic methods for software cost estimation are illustrated and they are compared with the intermediate COCOMO model. Comparing the features of the methods, it could be applied for clustering based on abilities and is also useful for selecting the special method for each project.

Keywords: Cost Estimation, COCOMO, Fuzzy Logic, Membership Functions, Software Effort Estimation Models.

I. INTRODUCTION

It has been surveyed that nearly 25% projects overrun their budget and delivered late and around 60% of many major projects substantially overrun their original estimates. Software development starts with the requirement phase and these requirements are very uncertain [10] due to which the software cost estimation becomes a hectic task. Without reasonably accurate cost estimation capability, project managers are not able to determine how much time and manpower cost the project should take which means the software portion of the project is out of control from its beginning [1]. There are aspects of the process that are peculiar to software estimation. Some of the unique aspects of software estimation are driven by the nature of software as a product. Many software managers struggle with estimating their projects. The inherent problem with estimation is that small projects can be estimated easily, but the required accuracy is not important. On the other hand, large projects are very difficult to estimate, but the required accuracy is extremely important [16].

A. Intermediate COCOMO

The Basic COCOMO model [4] [15] is based on the relationship: Development Effort, DE =a*(SIZE)^b; where SIZE is measured in thousand delivered source instructions. The constants a, b depends upon the ‘mode’ of development of projects. DE is measured in man-months. The accuracy of Basic COCOMO [6] is limited as it does not consider the factors like personnel, use of modern tools, hardware and other attributes that affect the project cost. Later, Boehm proposed the Intermediate COCOMO [4] [11] which adds accuracy to the Basic COCOMO by adding ‘Cost Drivers’ into the equation with a new variable: EAF (Effort Adjustment Factor). The development effort is now calculated as:

If development mode is Organic, then DE =EAF * 3.2 * (SIZE)^1.05

If development mode is Semi-detached, then DE =EAF * 3.0 * (SIZE)^1.12

If development mode is Embedded, then DE = EAF * 2.8 * (SIZE)^1.2

The Effort Adjustment Factor (EAF) is the product of 15 Cost Drivers [12] [13]. There are also six multipliers for the cost drivers. These are Very Low, Low, Nominal, High, Very High and Extra High. For example, in a project, if DATA is High, RELY is Low, TIME is Very High, STOR is High, CPLX is extra high and the remaining parameters are nominal then EAF = 0.75 *1.08 *1.65 *1.30 *1.06 *1.0. If the values of all the 15 cost drivers are “Nominal”, then EAF is equal to 1.

II. FUZZY BASED ESTIMATION TECHNIQUE

All the systems whose work is based on fuzzy logic try to embed reasoning in human behavior. In many of the problems where the decision making tends to be difficult where the conditions being uncertain and vague [10], fuzzy systems act as an efficient tool. This technique always keeps in mind the facts which may be ignored [5].
A. Fuzzy Membership Functions
A fuzzy set is characterized by a membership function [9] [10], which associates a real number in the interval (0, 1) with each point in the fuzzy set, called degree or grade of the membership. The membership function may be triangular, trapezoidal, Gaussian etc.

B. Triangular Membership Function
Triangular membership can be defined using a triplet (α, β, γ), where ‘β’ is the modal value, ‘α’ is the right boundary and ‘γ’ is the left boundary. The triangular membership function with straight lines can formally be described as follows:

\[ \Lambda(u; \alpha, \beta, \gamma) = \begin{cases} 
0 & \text{if } u < \alpha \\
\frac{u - \alpha}{\beta - \alpha} & \text{if } \alpha \leq u \leq \beta \\
\frac{\alpha - u}{\beta - \alpha} & \text{if } \beta < u \leq \gamma \\
0 & \text{if } u > \gamma 
\end{cases} \]

Fig. 1: Triangular Membership Function

C. Trapezoidal Membership Function
Trapezoidal Membership Function furnished in Fig. 2 is defined as follows:

\[ f(x; a, b, c, d) = \begin{cases} 
0 & \text{if } x < a \text{ and } x > d \\
\frac{x - a}{b - a} & \text{if } a \leq x \leq b \\
1 & \text{if } b \leq x \leq c \\
\frac{d - x}{d - c} & \text{if } c \leq x \leq d 
\end{cases} \]

Fig. 2: A Typical Form of The Trapezoidal Membership Function

D. Gaussian Membership Function
Gaussian membership function is often used to represent vague, linguistic terms and is defined as:

\[ G(u; m, \sigma) = \exp\left[-\frac{(u - m)^2}{2 \sigma^2}\right] \]

Where the parameters m and σ are used to control the center and width of the membership function.

Fig. 3: A Typical Form of The Gaussian Function

III. Fuzzy Rules
The rules based on the fuzzy sets [14] of MODE, SIZE and EFFORT appears in the following form:
For organic MODE and SIZE = S1, EFFORT is EF1
For semidetached MODE and SIZE = S, EFFORT is EF2
For embedded MODE and SIZE = S1, EFFORT is EF3
For organic MODE and SIZE = S2, EFFORT is EF4
For semidetached MODE and SIZE = S2, EFFORT is EF5
For embedded MODE and SIZE = S3, EFFORT is EF6
For organic MODE and SIZE = S4, EFFORT is EF4
For embedded MODE and SIZE = S5, EFFORT is EF5
For organic MODE and SIZE = S4, EFFORT is EF4.

IV. Criteria for Software Effort Assessment
(1) Balance Relative Error (BRE)
BRE = \left| \frac{E - E'}{\min(E, E')} \right|

(2) Prediction (n)
Prediction at level n is defined as the % of projects that have absolute relative error less than n [2].

(3) Variance Absolute Relative Error (VARE)
\[ \text{VARE}(\%) = \sum f(R_E - \text{mean}R_E)^2 * 100 \quad \ldots[8] \]

(4) Mean Absolute Relative Error (MARE)
\[ \text{MARE}(\%) = \frac{\sum |RE|}{\sum f} * 100 \quad \ldots[7] \]

(5) Variance Accounted For (VAF)
\[ \text{VAR}(\%) = \left(1 - \frac{\text{var}(E - \hat{E})}{\text{var}E}\right) * 100 \quad \ldots[3] \]

Where E = estimated effort, E' = actual effort
And Absolute Relative Error (RE) = \[\frac{|E - E'|}{E}\]

\[\text{Var } x = \sum f(x-x')^2 \sum f\]
\[x' = \text{mean of } x\]
\[f = \text{frequency}\]

A model which gives higher Variance Accounted For (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 [12]. A model which gives lower VARE is better than that which gives higher VARE [14]. A model which gives lower BRE is better than that which gives higher BRE.

V. EXPERIMENTAL STUDY

The COCOMO81 database contains 63 projects data [4] [12], in which 28 are Embedded Mode, 12 are Semi-Detached Mode and 23 are Organic Mode projects. Hence, there is no uniformity for selecting the projects of different modes. For our purpose, we have chosen 15 projects data, with their lines of code to be less than 100KDSI. Further, we have estimated the efforts using Intermediate COCOMO, Triangular Membership Function, Gaussian Membership Function and Trapezoidal Membership Function which are shown in Table 1 whereas the Table 2 shows the comparison of various models based on different criteria.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Size</th>
<th>EAF</th>
<th>Actual Effort</th>
<th>COCOMO Effort</th>
<th>TMF Effort</th>
<th>Gaussian Effort</th>
<th>Trapezoidal Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.05</td>
<td>46</td>
<td>1.17</td>
<td>240</td>
<td>212</td>
<td>246</td>
<td>252</td>
<td>249</td>
</tr>
<tr>
<td>1.05</td>
<td>16</td>
<td>0.66</td>
<td>33</td>
<td>39</td>
<td>41</td>
<td>41</td>
<td>40</td>
</tr>
<tr>
<td>1.05</td>
<td>4</td>
<td>2.22</td>
<td>43</td>
<td>30</td>
<td>34</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>1.05</td>
<td>6.9</td>
<td>0.4</td>
<td>8</td>
<td>9.8</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>1.05</td>
<td>13</td>
<td>2.81</td>
<td>98</td>
<td>133</td>
<td>143</td>
<td>143</td>
<td>142</td>
</tr>
<tr>
<td>1.2</td>
<td>22</td>
<td>7.62</td>
<td>1075</td>
<td>869</td>
<td>1078</td>
<td>1116</td>
<td>1098</td>
</tr>
<tr>
<td>1.2</td>
<td>30</td>
<td>2.39</td>
<td>423</td>
<td>397</td>
<td>484</td>
<td>485</td>
<td>484</td>
</tr>
<tr>
<td>1.2</td>
<td>18</td>
<td>2.38</td>
<td>321</td>
<td>214</td>
<td>239</td>
<td>231</td>
<td>228</td>
</tr>
<tr>
<td>1.2</td>
<td>20</td>
<td>2.38</td>
<td>218</td>
<td>243</td>
<td>287</td>
<td>303</td>
<td>270</td>
</tr>
<tr>
<td>1.2</td>
<td>24</td>
<td>0.85</td>
<td>79</td>
<td>108</td>
<td>138</td>
<td>138</td>
<td>138</td>
</tr>
<tr>
<td>1.12</td>
<td>3</td>
<td>5.86</td>
<td>73</td>
<td>60</td>
<td>63</td>
<td>62</td>
<td>60</td>
</tr>
<tr>
<td>1.12</td>
<td>90</td>
<td>0.70</td>
<td>453</td>
<td>326</td>
<td>453</td>
<td>486</td>
<td>475</td>
</tr>
<tr>
<td>1.12</td>
<td>2.14</td>
<td>1</td>
<td>7.3</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>1.12</td>
<td>1.98</td>
<td>0.91</td>
<td>5.9</td>
<td>5.8</td>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>1.12</td>
<td>12</td>
<td>0.68</td>
<td>55</td>
<td>33</td>
<td>31</td>
<td>31</td>
<td>31</td>
</tr>
</tbody>
</table>

VI. CONCLUSION AND FUTURE WORK

This paper has presented an overview of a variety of non-algorithmic software cost estimation techniques. We have observed that no method or any estimation model should be preferred over the others. Referring to Table 2, we have observed that fuzzy
method using Triangular Membership Function yields better results for maximum criteria when compared to other fuzzy logic methods. Based on VAF, MARE & Mean BRE, we have concluded that the Fuzzy method using TMF (triangular membership function) is better when compared to the Gaussian Bell, Trapezoidal or Intermediate COCOMO. There is no estimation method which can present the best estimates in all the various criteria and each technique can be considered as suitable in the especial project. Trying to make the performance of the existing methods better and comparing the results of various fuzzy logic methods with the Detailed COCOMO and also exploring the new methods for software cost estimation based on today’s software project requirements can be the future works in this area.

REFERENCES