Fuzzy logic for Software Effort Estimation Using Polynomial Regression as Firing Interval

1J.N.V.R Swarup Kumar, MIEEE, CSE Dept, GEC, Gudlavalleru, India, swarupjnvr@yahoo.co.in.
2Aravind Mandala, CSE Dept, GEC, Gudlavalleru, India, aravind_mandala@yahoo.com.
3M. Vishnu Chaitanya, SITE, VIT University, India, get2vishnu@gmail.com.
4G.V.S.N.R.V Prasad, HOD of CSE Dept, GEC, Gudlavalleru, India, guttaprasad@yahoo.co.in.

Abstract

The process of estimating the cost and time required to develop a software system is known as Software effort estimation. In software project decisions like resource allocation and bidding which are major parts of planning, software effort estimation plays a prominent role. The substratal goals of planning are to scout for the future, to diagnose the attributes that are essentially done for the consummation of the project successfully. So, the effective Software cost estimation is one of the most challenging and important activities in Software development. This paper introduces a new model using fuzzy logic to estimate the effort required in software development. MATLAB is used for tuning the parameters of famous various cost estimation models. On published software projects data, the performance of the model is evaluated. Comparison of results from our model with existing ubiquitous models is done.

Index Terms— COCOMO, Effort Estimation, Fuzziness, Fuzzy Logic, KLOC, Membership Function.

1. Introduction

The process of predicting the amount of time (Effort) required to build a software system is known as software effort estimation. In order to perform cost-benefit analysis, cost estimation is to be performed by client or developer. Cost Estimation is achieved in terms of person-months (PM), which can be translated into actual dollar cost. Estimation carries inherent risk and this risk leads to obscurity. Due to practicality and demand of software cost estimation, this concept of software cost estimation has been growing rapidly. In today’s environment people are expecting high quality software with a low cost: So many popular cost estimation models like COCOMO81, COCOMOII, SLIM, FP, Delphi, Halsted Equation, Bailey-Basili, Doty, Walston Felix Model and Anish Mittal Model have come into existence. When regression analysis methods applied to historical data, these models are created. As per the recent review of surveys on software cost estimation found that software projects have cost overruns. COCOMOII is followed by most of the software companies for estimating the cost of products; but we found some variations in this model [2], [5-10]. There are several reasons like “unrealistic over-optimum”, “complexity”, “and overlooked tasks” [11], [12].

Researchers particularly from 1990’s have turned their attention to a set of approaches that are soft computing based. These include artificial neural networks, fuzzy logic models and genetic algorithms. Fuzzy logic with its features of a powerful linguistic representation can signify imprecision in inputs and outputs, while providing a more expert knowledge based approach to model building. The fuzzy logic model uses the fuzzy logic concepts introduced by Lofti A. Zadeh [3], [4], [13].

A. Membership Functions

One of the ways to describe data vagueness, obscurity and imprecision is fuzzy numbers. A fuzzy number is an extension of a regular number in the sense that it does not refer to one single value but rather to a connected set of possible values, where each possible value has its own weight between ‘0’ and ‘1’. This weight is called the membership function. The membership function is increasing towards the mean and decreasing away from it. The Fuzzy number can be of three types 1) Triangular fuzzy Number 2) Trapezoidal fuzzy number 3) Bell shaped fuzzy number.

Fuzziness in a fuzzy set is characterized by its 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. Different shapes may exist in graphical representations. An important criterion that has to be considered is there are certain restrictions regarding the shapes used. There are different methods to form membership functions.

B. T Function (Figure 1)

Defined by its lower limit a, its upper limit b, and the modal value m, so that a < m < b. We call the value b − m margin when it is equal to the value m − a.

\[
A(x) = \begin{cases} 
0 & \text{if } x \leq a \text{ or } x \geq b \\
\frac{(x-a)}{(m-a)} & \text{if } x \in (a, m) \\
1 & \text{if } x \in (m, b) \\
\frac{(b-x)}{(b-m)} & \text{if } x \in (m, b) 
\end{cases} \quad \ldots (1)
\]
2. COST ESTIMATION MODELS LITERATURE REVIEW

To improve the accuracy of cost estimation many software cost estimation models [2], [6-10] were introduced, within last few decades. Because of the inherent obscurity in software development projects and the impact of software development cost use, it seems to be impractical. Still, it is likely that the estimation can be improved because software development cost estimates are systematically overoptimistic and very inconsistent. Estimation models use KDLOC (Thousands of Delivered Lines of Code) as the primary input. This input is not sufficient for accurately estimating the cost of products. Several other parameters have to be considered.

A. COCOMO Basic Model

Boehm described COCOMO as a collection of three variants: basic model, intermediate model, detailed model [11]. The basic COCOMO model computes effort as function of program size, and it is same as single variable method.

\[ \text{Effort} = a (\text{KLOC})^b \]  
\[ \text{Effort} = 3.2 (\text{KLOC})^{1.05} \]  
\[ \text{(2)} \]
\[ \text{(3)} \]

B. Doty Model: [1]

\[ \text{Effort} = 5.288 (\text{KLOC})^{1.047} \]  
\[ \text{(4)} \]

C. Halsted Equation:

\[ \text{Effort} = 5.2 (\text{KLOC})^{1.50} \]  
\[ \text{(5)} \]

D. Baili-Basili Model:

\[ \text{Effort} = 5.5+0.73 (\text{KLOC})^{1.16} \]  
\[ \text{(6)} \]

E. Walston Felix Model:

\[ \text{Effort} = 5.2 (\text{KLOC})^{0.91} \]  
\[ \text{(7)} \]

3. PROPOSED MODEL EFFORT ESTIMATION USING FUZZY

A. Fuzzification

We use Triangular Fuzzy number T(N) which is defined as follows:

\[ T(N) = \begin{cases} 
0 & \text{if } N \leq a \\
\frac{(N-a)}{(m-a)} & \text{if } a \leq N \leq m \\
\frac{(m-N)}{(m-\beta)} & \text{if } m \leq N \leq \beta \\
0 & \text{if } N \geq \beta 
\end{cases} \]  
\[ \text{(8)} \]

Where N is size as input. E is effort as output, a, m and b are the parameters of membership function T(N), m is the modal value, a and b are the left and right boundaries respectively.

Let (m, 0) divider internally, the base q the triangle in ratio k:1 where k is real positive number.

So that,

\[ m = \frac{\alpha + K\beta}{K+1} \]  
\[ \alpha + K\beta = (K+1) m \]  
\[ \text{(9)} \]

By definition of fuzziness

\[ F = \frac{\beta - \alpha}{2m} \]  
\[ \beta - \alpha = 2mF \]  
\[ \text{(10)} \]

By solving (9) & (10), we get

\[ \alpha = \left( 1 - \frac{2KF}{K+1} \right) m \]  
\[ \beta = \left( 1 + \frac{2F}{K+1} \right) m \]  
\[ \text{(11)} \]
\[ \text{(12)} \]

Similarly, the TFN \( \mu(E) \) is defined as

\[ \mu(E) = \begin{cases} 
0 & \text{if } E \leq a + b \alpha + c \alpha^2 \\
\frac{m-a}{m-\alpha} & \text{if } a + b \alpha + c \alpha^2 \leq E \leq a + bm + cm^2 \\
\frac{m-a}{\beta-m} & \text{if } a + bm + cm^2 \leq E \leq a + b\beta + c\beta^2 \\
0 & \text{if } E \geq a + b\beta + c\beta^2 
\end{cases} \]  
\[ \text{(13)} \]

Table 1 gives the values \( \alpha \) and \( \beta \) for F = 0.1, 0.2 and 0.3 for various values of K using equations (11) and (12), where m, size estimate in KLOC.

Table 1: values of a and \( \beta \) for various values of F for various values of K.

<table>
<thead>
<tr>
<th>F</th>
<th>K=1</th>
<th>K=2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>( \alpha = 0.9m )</td>
<td>( \alpha = 0.86m )</td>
</tr>
<tr>
<td></td>
<td>( \beta = 1.1m )</td>
<td>( \beta = 1.06m )</td>
</tr>
<tr>
<td>0.2</td>
<td>( \alpha = 0.8m )</td>
<td>( \alpha = 0.73m )</td>
</tr>
<tr>
<td></td>
<td>( \beta = 1.2m )</td>
<td>( \beta = 1.13m )</td>
</tr>
<tr>
<td>0.3</td>
<td>( \alpha = 0.7m )</td>
<td>( \alpha = 0.6m )</td>
</tr>
<tr>
<td></td>
<td>( \beta = 1.3m )</td>
<td>( \beta = 1.12m )</td>
</tr>
</tbody>
</table>

B. Defuzzification

The output, fuzzy estimate of E, can be computed as a weighted average of the optimistic \((a+h\alpha+c\alpha^2)\), most likely
(a+b\beta+c\beta^2)

Fuzzy effort estimate (E) is given as

$$E = \frac{w_1(a+bx+cx^2) + w_2(a+bx+cx^2) + w_3(a+bx+cx^2)}{w_1+w_2+w_3}$$

Where w_1, w_2 and w_3 are weights of the optimistic, most likely and pessimistic estimate respectively. Maximum weight should be given to the most expected estimate.

$$\alpha = \left(1 - \frac{2KF}{K+1}\right) m$$

$$\beta = \left(1 + \frac{2F}{K+1}\right) m$$

Here K, F, w_1, w_2 and w_3 are arbitrary constants. The effort is estimated in man months (MM).

4. RESEARCH METHODOLOGY

Proposed software effort estimation model performance is evaluated by comparing against various software cost estimation models. The methodology used in empirical evaluation is described as follows:

- Using MRE we evaluate the impact of estimation accuracy using (MRE, MARE) evaluation criteria, for each model.

5. EXPERIMENTAL RESULT ANALYSIS

The Data is taken from [2] and given in Table 2

Table 2: Data for Experimental Study

<table>
<thead>
<tr>
<th>SL.NO</th>
<th>Project No</th>
<th>KLOC</th>
<th>Actual Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>39</td>
<td>72</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>40.5</td>
<td>82.5</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>50</td>
<td>84</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>128.6</td>
<td>230.7</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>161.4</td>
<td>157</td>
</tr>
<tr>
<td>6</td>
<td>13</td>
<td>164.8</td>
<td>246.9</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>200</td>
<td>130.3</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>214.4</td>
<td>86.9</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>253.6</td>
<td>287</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>254.2</td>
<td>258.7</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>289</td>
<td>116</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>449.9</td>
<td>336.3</td>
</tr>
</tbody>
</table>

Let F=0.3, k=1 then from table \(\alpha = 0.7m, \beta = 1.3m\) we have taken \(w_1 = 50, w_2 = 2, w_3 = 1\) for our model. Table 3 below gives the experimental results.

Table 3: Effort Estimated of Various Models

<table>
<thead>
<tr>
<th>Proj.No</th>
<th>KLOC</th>
<th>Actual Effort</th>
<th>COCOMO Basic Model</th>
<th>Doty Model</th>
<th>Halsted Equation</th>
<th>Bailey-Basili</th>
<th>Walston Felix</th>
<th>Swarup Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39</td>
<td>72</td>
<td>149.89</td>
<td>245</td>
<td>1266.5</td>
<td>56.7</td>
<td>145.8</td>
<td>116.761258</td>
</tr>
<tr>
<td>2</td>
<td>40.5</td>
<td>82.5</td>
<td>155.95</td>
<td>254.9</td>
<td>1340.3</td>
<td>59</td>
<td>150.9</td>
<td>116.181401</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>84</td>
<td>194.57</td>
<td>317.8</td>
<td>1838.5</td>
<td>73.8</td>
<td>182.8</td>
<td>112.731302</td>
</tr>
<tr>
<td>4</td>
<td>128.6</td>
<td>230.7</td>
<td>524.63</td>
<td>854.4</td>
<td>7583.4</td>
<td>209.7</td>
<td>431.9</td>
<td>98.9188141</td>
</tr>
<tr>
<td>5</td>
<td>161.4</td>
<td>157</td>
<td>665.96</td>
<td>1083.8</td>
<td>10662.5</td>
<td>271.3</td>
<td>531.1</td>
<td>100.928712</td>
</tr>
<tr>
<td>6</td>
<td>164.8</td>
<td>246.9</td>
<td>680.7</td>
<td>1107.8</td>
<td>11001.2</td>
<td>277.8</td>
<td>541.3</td>
<td>101.398913</td>
</tr>
<tr>
<td>7</td>
<td>200</td>
<td>130.3</td>
<td>834.13</td>
<td>1356.7</td>
<td>14707.8</td>
<td>346.3</td>
<td>645.6</td>
<td>109.157623</td>
</tr>
<tr>
<td>8</td>
<td>214.4</td>
<td>86.9</td>
<td>897.3</td>
<td>1459.1</td>
<td>16324.5</td>
<td>374.9</td>
<td>687.7</td>
<td>113.85122</td>
</tr>
<tr>
<td>9</td>
<td>253.6</td>
<td>287</td>
<td>1070.3</td>
<td>1739.5</td>
<td>21000.4</td>
<td>454.4</td>
<td>801.3</td>
<td>131.098465</td>
</tr>
<tr>
<td>10</td>
<td>254.2</td>
<td>258.7</td>
<td>1072.96</td>
<td>1743.9</td>
<td>21075</td>
<td>485.6</td>
<td>803</td>
<td>131.413259</td>
</tr>
<tr>
<td>11</td>
<td>289</td>
<td>116</td>
<td>1227.7</td>
<td>1994.6</td>
<td>25547.6</td>
<td>527.9</td>
<td>902.4</td>
<td>152.292295</td>
</tr>
<tr>
<td>12</td>
<td>449.9</td>
<td>336.3</td>
<td>1953.99</td>
<td>3170.3</td>
<td>49622.4</td>
<td>878.3</td>
<td>1350</td>
<td>315.820321</td>
</tr>
<tr>
<td>13</td>
<td>450</td>
<td>1107.31</td>
<td>1954.44</td>
<td>3171.1</td>
<td>49638.9</td>
<td>878.6</td>
<td>1350.3</td>
<td>315.956208</td>
</tr>
</tbody>
</table>

Figure 3 below show the comparison of experimental estimated effort from various models with swarup model and Actual effort.
Figure 3: Effort Model Vs Mittal Model and Actual Effort

Figure 4 below shows the comparison of effort from various models versus estimated effort.

![Comparison of Effort Estimated of various models](image)

Figure 4: Comparison of Effort Estimated of various models

Comparison of various models on the basis of various performance criterions for software cost estimation is given in below Table 4. Figure 5 below shows the Mean Absolute Relative Error (%) comparison of various models.

Table 4: Various Models MARE % Values

<table>
<thead>
<tr>
<th>Model Type</th>
<th>MARE %</th>
</tr>
</thead>
<tbody>
<tr>
<td>COCOMO81 Basic Model</td>
<td>4532.4</td>
</tr>
<tr>
<td>Doty Model</td>
<td>8186.1</td>
</tr>
<tr>
<td>Bailey-Basili</td>
<td>1324.9</td>
</tr>
<tr>
<td>Walston-felix</td>
<td>3225.4</td>
</tr>
<tr>
<td>Halsted Eq</td>
<td>105605.2</td>
</tr>
<tr>
<td>Swarup Model</td>
<td>548.6</td>
</tr>
</tbody>
</table>

![MARE (%) Comparison of various models](image)

Figure 5: MARE (%) Comparison of various models

6. CONCLUSION

To estimate the software development effort required accurately and reliably in the early stages of software development life cycle is the important issue for project managers, so that the resources allocation can be done perfectly. In this paper we postulated fuzzy software cost estimation model that handles ambiguity, obscurity and then compared with other popular software cost estimation models. Following are the conclusions from the empirical evaluation: Better software effort estimates in view of the MARE evaluation criteria by the proposed model as compared to the traditional estimation models. To address the problem of obscurity and vagueness existed in software effort drivers, the above results demonstrate that applying fuzzy logic method to the software effort estimation is an expedient approach. Hence, in future, fuzzy logic utilization for other applications in the software engineering field can also be explored.

7. REFERENCES