A Fuzzy Logic Based Software Cost Estimation Model

Ziauddin¹, Shahid Kamal², Shafiullah khan³ and Jamal Abdul Nasir⁴

¹COMSATS Institute of Information Technology, Vehari, Pakistan
²FSKSM, Universiti Teknologi Malaysia
³Kohat University of Science & Technology, Kohat, Pakistan
⁴ICIT, Gomal University, D.I.Khan, Pakistan

¹ziasahib@gmail.com, ²skamaltipu@gmail.com, ³skkust@hotmail.com,
⁴jamal_icit@hotmail.com

Abstract

Software cost estimation is a challenging and onerous task. Estimation by analogy is one of the expedient techniques in software effort estimation field. However, the methodology utilized for the estimation of software effort by analogy is not able to handle the categorical data in an explicit and precise manner. Early software estimation models are based on regression analysis or mathematical derivations. Today’s models are based on simulation, neural network, genetic algorithm, soft computing, fuzzy logic modelling etc. This paper aims to utilize a fuzzy logic model to improve the accuracy of software effort estimation. In this approach fuzzy logic is used to fuzzify input parameters of COCOMO II model and the result is defuzzified to get the resultant Effort. Triangular fuzzy numbers are used to represent the linguistic terms in COCOMO II model. The results of this model are compared with COCOMO II and Alaa Sheta Model. The proposed model yields better results in terms of MMRE, PRED(n) and VAF.

Keywords: Fuzzy logic, Triangular Fuzzy Number, Membership function and Fuzziness, Software Effort Estimation

1. Introduction

Estimating the work-effort and the schedule required to develop and/or maintain a software system is one of the most critical activities in managing software projects. The task is known as Software Cost Estimation [27]. During the development process cost and time estimates are useful for the initial rough validation and monitoring of the project’s progress; after completion, these estimates may be useful for project productivity assessment for example. There are different techniques used in software cost estimation:

Algorithm model, also called parametric model, is designed to provide some mathematical equations to provide software estimation. LOC-based models are algorithm models such as [2, 6, 7, 8]. Ali Idri and Laila Kjjri [9] proposed the use of fuzzy sets in the COCOMO-81 models [8]. Musilek P. and others [18] proposed f-COCOMO model, using fuzzy sets. The methodology of fuzzy sets giving rise to f-COCOMO [18] is sufficiently general to be applied to other models of software cost estimation such as function point method [14]. W. Pedrycz and others [19] found that the concept of information granularity and fuzzy sets, in particular, plays an important role in making software cost estimation models more user friendly. Harish Mittal and Pradeep Bhatia [17] used triangular fuzzy numbers for fuzzy logic sizing. Lima, O.S.J. and Others [13] proposed the use of concepts and properties from fuzzy set theory to
extend function point analysis to Fuzzy function point analysis, using trapezoid shaped fuzzy numbers for the linguistic variables of function point analysis complexity matrixes.

In this paper we propose triangular fuzzy numbers to represent the linguistic variables. The results can be optimised for the given application by varying fuzziness of the triangular fuzzy numbers. To apply fuzzy logic first fuzzification is done using triangular fuzzy number, Fuzzy output is then evaluated and estimation is done by defuzzification technique given in this paper.

1.1. Fuzzy Logic

In 1965, Lofti Zadeh formally developed multi-value set theory, and introduced the term fuzzy logic [5]. Fuzzy Logic (FL) starts with the concept of fuzzy set theory. It is a theory of classes with un-sharp boundaries, and considered as an extension of the classical set theory. The membership μ of an element x of a classical set A, as subset of the universe X, is defined as follows:

\[ \mu_A(x) = 1 \text{ if } x \in A \]
\[ \mu_A(x) = 0 \text{ if } x \notin A \]

A system based on FL has a direct relationship with fuzzy concepts (such as fuzzy sets, linguistic variables, etc.) and fuzzy logic. The popular fuzzy logic systems can be categorised into three types: Pure fuzzy logic systems, Takagi and Sugeno’s fuzzy system, and fuzzy logic system with fuzzifier and defuzzifier. Since most of the engineering applications produce crisp data as input and expects crisp data as output, the last type is the most widely used type of fuzzy logic systems. Fuzzy logic system with fuzzifier and defuzzifier, first, proposed by Mamdani and it has been successfully applied to a variety of industrial processes and consumer products [16]. The main three steps of applying fuzzy logic to a model are:

Step 1: Fuzzification: It converts a crisp input to a fuzzy set


Fuzzy Inference Engine: Once all crisp input values are fuzzified into their respective linguistic values, the inference engine accesses the fuzzy rule base to derive linguistic values for the intermediate and the output linguistic variables. Step 3: Defuzzification: It converts fuzzy output into crisp output.

Use of fuzzy sets in logical expression is known as fuzzy logic. A fuzzy set 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 [3, 4, 20, 21]. The membership function may be triangular, trapezoidal, parabolic etc. Fuzzy numbers are special convex and normal fuzzy sets, usually with single modal value, representing uncertain quantitative information.

A fuzzy number is a quantity whose value is imprecise, rather than exact as in the case of ordinary single valued numbers [4, 9, 20, 21, 23]. Any fuzzy number can be thought of as a function, called membership function, whose domain is specified, usually the set of real numbers, and whose range is the span of positive numbers in the closed interval (0, 1). Each numerical value of the domain is assigned a specific value and 0 represents the smallest possible value of the membership function, while the largest possible value is 1. In many respects fuzzy numbers depict the physical world more realistically than single valued numbers. The curve in Figure 1a is a triangular fuzzy number, the curve in Figure 1b is a trapezoidal fuzzy number, and the curve in Figure 1c is bell shaped fuzzy number.
A triangular fuzzy number (TFN) is described by a triplet \((\alpha, m, \beta)\), where \(m\) is the modal value, \(\alpha\) and \(\beta\) are the right and left boundary respectively. Fuzziness of a TFN \((\alpha, m, \beta)\) is defined as:

\[
\text{Fuzziness of TFN (F)} = \frac{\beta - \alpha}{2m}, \quad 0 < F < 1
\]

The higher the value of fuzziness, the more fuzzy is TFN.

1.2. COCOMO II

The COCOMO I \([7]\) model is a regression-based software cost estimation model, which was developed by Boehm in 1981 and thought to be the most cited and the most plausible model among all traditional cost estimation models. The COCOMO I was a stable model on that time. One of the problems with the use of COCOMO I today is that it does not match the development environment of the late 1990’s. Therefore, in 1997, Boehm developed the COCOMO II to solve most of the COCOMO I problems.

Figure 2 shows the process of software schedule, cost, and manpower estimation in the COCOMO II. The COCOMO II includes several software attributes such as: 17 Effort Multipliers (EMs), 5 Scale Factors (SFs), Software Size (SS), and Effort estimation that are used in the Post Architecture Model of the COCOMO II.

The formula for the process is given by:

\[
\text{Effort (PM)} = A \times \text{Size}^B + 0.01 \times \sum_{j=1}^{5} \text{SF}_j \times \prod_{i=1}^{17} \text{EM}_i
\]

\[
\text{Schedule (Months)} = C \times \text{Effort}^D + 0.01 \times \sum_{j=1}^{5} \text{SF}_j
\]

Personnel = Effort/Schedule

Where \(A=2.94\); \(B=0.91\); \(C=3.67\); \(D=0.28\)

More details about the Model can be found in [7].
2. Proposed Model

Our model is established based on the COCOMO II and Fuzzy Logic. The COCOMO II includes a set of input software attributes: 17 Effort Multipliers (EMs), 5 Scale Factors (SFs), one Size in KLOC (SZ) and one output, Effort. The architecture of the Model is shown in Figure 3.

- EMs (17)
- SF (5)
- SZ (1)

Fuzzification

Rule Base

Inference Engine

Defuzzification

EFFORT

2.1. Inputs

There are three set of inputs
- Size in KLOC
- 17 Effort Multipliers
- 5 Scale Factors

All these inputs are provided as crisp data. These inputs are directed to fuzzification module.

2.2. Fuzzification

This module is sub divided into three sub modules. SZ-Fuzzifications module converts the Size input to the fuzzy variable, SF-Fuzzification module converts Scale Factors input to fuzzy variables whereas EM- Fuzzification module converts the Effort Multiplier input to Fuzzy variable.

The Terms Very Low, Low, Nominal, High, Very High and extra High have been defined for each variable. We defined a fuzzy set for each linguistic value for Triangular Membership Function.

A Triangular Membership Function (TMF) is a three-point function [8], defined by minimum (α), Maximum (β) and modal (m) values, that is, TMF (α, m, β), where (α ≤ m ≤ β). We take each linguistic variables as a triangular Fuzzy numbers, TFN (α, m, β), α≤m, β≥m. The membership function (μ(x)) for which is defined as:

\[
\mu(x) = \begin{cases} 
0, & x \leq \alpha \\
(x-\alpha) / (m-\alpha), & \alpha \leq x \leq m \\
(\beta-x) / (\beta-m), & m \leq x \leq \beta \\
0, & x \geq \beta 
\end{cases}
\]

Where m is the mean of input sizes.
The Fuzzy Set defined for Size is shown in Figure 4a. The Size range has been divided into small intervals to get a better fuzzy number. Figure 4b represents Fuzzy Set for Scale Factor RESL whereas Figure 4c represents Fuzzy Set for Effort Multiplier PERS. All the scale factors and effort multipliers have scaling range from Very Low to Extra High.

The Fuzzication Process converts the crisp data to linguistic variables which are passed to Inference Engine.
2.3. Inference Engine

Inference engine operates on a set of fuzzy rules. The Fuzzy Model rules contain the linguistic variables related to the project. We have implemented this model using FIS tool in MATLAB. A sample of the rules is presented below:

if (PREC is VL) then (EFFORT is XH)
if (PREC is LO) then (EFFORT is VH)
if (FLEX is VL) then (EFFORT is XH)

The number of rules that have been used in the proposed model are 281.

2.4. Defuzzification

This module calculates and converts the fuzzy output into crisp data form. The defuzzification has been performed using Centeroide Method. It has been used to calculate the Centre of Gravity (COG) area under the curve as

$$ COG = \frac{\sum_{x=a}^{b} \mu_A(x)x}{\sum_{x=a}^{b} \mu_A(x)} $$

3. Experimental Study

In order to carry out our experiments to evaluate the efficiency of our proposed model, we have chosen 30 projects data, developed in local software house, ranging from Very Small to Large. We have chosen two estimation methods, COCOMO II and Alaa Sheta [1a] estimation method.

We have used following MMRE, PRED (L) and VAF as criteria for assessment of these models.

The Mean Magnitude of Relative Error, MMRE, is probably the most widely used evaluation criterion for assessing the performance of competing models. One purpose of MMRE is to assist us to select the best model as it gives us accuracy prediction of the competing models. MMRE is calculated as average of the Magnitude of Relative Errors MREs of each estimate. The MRE and MMRE are calculated as:

$$ MRE = \frac{|E - E'|}{\bar{E}} $$

$$ MMRE = \frac{\sum_{i=1}^{n} MRE_i}{n} $$

Variance Account For (VAF) is used in the context of statistical models whose main purpose is the prediction of future outcomes on the basis of other related information. It is the proportion of variability in a data set that is accounted for by the statistical model. It provides a measure of how well future outcomes are likely to be predicted by the model. It is calculated as:

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

Where E is the actual Effort and E’ is the Calculated Effort.
Prediction $\text{PRED}(L)$ is the probability of the model having relative Error less than or equal to $L$. It is calculated:

$$\text{PRED}(L) = \frac{K}{N}$$

Where $K$ is the number of Observations where MRE is less than or equal to $L$ and $N$ are total number of observations.

Table 1 shows estimated efforts of all three models. It also shows MRE of all three models for every estimate. Table 2 shows comparison of these models. The estimation results are graphically shown in Figure 5, 6 and 7. Comparison of the model results in Table 2 shows that the proposed model has better estimation accuracy as compared to other models having 7.512 MMRE with 98.77% likeliness to have the same ratio of MMRE if the model is tested under different environment with different information. In order to further verify the prediction we have analysed the results for 25%, 15%, 10% and 8% prediction. The results show that the proposed model is 63.33% confident to have its average MRE, which shows the stability of its estimates.

The stability of estimates of the proposed model can also be verified from Figure 5, 6, 7. The Figure 7 shows that there are minute variations in actual and estimated calculations.

### Table 1. Estimated Efforts and MREs of all Three Models

<table>
<thead>
<tr>
<th>P.No</th>
<th>Size</th>
<th>Actual</th>
<th>COCOMO II</th>
<th>Alaa Sheta</th>
<th>Proposed</th>
<th>MRE COCOMO</th>
<th>MRE Alaa Sheta</th>
<th>MRE Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18</td>
<td>301</td>
<td>214</td>
<td>243</td>
<td>293</td>
<td>28.90365</td>
<td>19.2691</td>
<td>2.657807</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>1063</td>
<td>841</td>
<td>931</td>
<td>984</td>
<td>20.88429</td>
<td>12.41769</td>
<td>7.431797</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>605</td>
<td>601</td>
<td>453</td>
<td>537</td>
<td>0.661157</td>
<td>25.12397</td>
<td>11.23967</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>243</td>
<td>282</td>
<td>181</td>
<td>201</td>
<td>16.04938</td>
<td>25.5144</td>
<td>17.28395</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>141</td>
<td>128</td>
<td>95</td>
<td>127</td>
<td>9.219858</td>
<td>32.62411</td>
<td>9.929078</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>112</td>
<td>131</td>
<td>91</td>
<td>121</td>
<td>16.96429</td>
<td>18.75</td>
<td>8.035714</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>16</td>
<td>15</td>
<td>13</td>
<td>16</td>
<td>6.25</td>
<td>18.75</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>91</td>
<td>78</td>
<td>65</td>
<td>82</td>
<td>14.28571</td>
<td>28.57143</td>
<td>9.89011</td>
</tr>
<tr>
<td>9</td>
<td>42</td>
<td>781</td>
<td>751</td>
<td>634</td>
<td>722</td>
<td>3.841229</td>
<td>18.82202</td>
<td>7.554417</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>233</td>
<td>241</td>
<td>201</td>
<td>249</td>
<td>3.433476</td>
<td>13.73391</td>
<td>6.866953</td>
</tr>
<tr>
<td>11</td>
<td>21</td>
<td>334</td>
<td>304</td>
<td>289</td>
<td>318</td>
<td>8.982036</td>
<td>13.47305</td>
<td>4.790419</td>
</tr>
<tr>
<td>12</td>
<td>52</td>
<td>982</td>
<td>1043</td>
<td>756</td>
<td>890</td>
<td>6.211813</td>
<td>23.01426</td>
<td>9.368635</td>
</tr>
<tr>
<td>13</td>
<td>23</td>
<td>305</td>
<td>361</td>
<td>300</td>
<td>281</td>
<td>18.36066</td>
<td>1.639344</td>
<td>7.868852</td>
</tr>
<tr>
<td>14</td>
<td>27</td>
<td>421</td>
<td>459</td>
<td>398</td>
<td>401</td>
<td>9.026128</td>
<td>5.463183</td>
<td>4.750594</td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>45</td>
<td>48</td>
<td>36</td>
<td>41</td>
<td>6.666667</td>
<td>20</td>
<td>8.888889</td>
</tr>
<tr>
<td>16</td>
<td>15</td>
<td>189</td>
<td>197</td>
<td>178</td>
<td>192</td>
<td>4.232804</td>
<td>5.820106</td>
<td>1.587302</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>17</td>
<td>19</td>
<td>14</td>
<td>17</td>
<td>11.76471</td>
<td>17.64706</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>6</td>
<td>22</td>
<td>23</td>
<td>19</td>
<td>22</td>
<td>4.545455</td>
<td>13.63636</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>0</td>
<td>12.5</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 2. Comparison of the Models

<table>
<thead>
<tr>
<th></th>
<th>COCOMO II</th>
<th>Alaa Sheta</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMRE</td>
<td>11.003%</td>
<td>16.838%</td>
<td>7.512%</td>
</tr>
<tr>
<td>VAF</td>
<td>95.86%</td>
<td>93.90%</td>
<td>98.77%</td>
</tr>
<tr>
<td>PRED(25)</td>
<td>93.33%</td>
<td>80%</td>
<td>96.33%</td>
</tr>
<tr>
<td>PRED(15)</td>
<td>63.33%</td>
<td>53.33%</td>
<td>93.33%</td>
</tr>
<tr>
<td>PRED(10)</td>
<td>50%</td>
<td>20%</td>
<td>80%</td>
</tr>
<tr>
<td>PRED(8)</td>
<td>40%</td>
<td>10%</td>
<td>63.33%</td>
</tr>
</tbody>
</table>

Figure 5. COCOMO II Vs Actual Calculation of Effort
4. Conclusion

In this paper, we have presented a Software Effort Estimation Model using Fuzzy Logic Model. Triangular membership function has been used to generate linguistic fuzzy values. One can easily develop the same model using trapezoidal or Gauss-Bell membership functions. Similarly, there is still margin of improvement in fuzzification process. If fuzzification process is further optimized, it will certainly result in substantial reduction in number of fuzzy rules implemented in inference engine.
References


Authors

Ziauddin

Dr. Ziauddin is working as Assistant Professor in COMSATS Institute of Information Technology, Vehari. He has worked in Gomal University for 22 years. He has published more than 15 research papers in reputed International journals. His area of expertise is Software Engineering, Software Process Improvement, Software Reliability Engineering, Software Development Improvement, Software Quality Assurance and Requirement Engineering. He is Gold Medalist from Peshawar University in M.Sc Computer Science. He has Diploma in Computer Forensics from School of Education, Indiana University USA.

Shahid Kamal

Shahid Kamal Tipu is working as Assistant Professor in Institute of Computing & Information Technology, Gomal University, D.I.Khan since 2000. Currently he is doing his Ph.D. in Malaysia. His area of expertise is Information System Security and Software Quality. He has published his research in reputed International Journals. He is a renowned software engineer in local industry.

Shafiullah Khan

Dr. Shafiullah Khan is Assistant professor in Institute of Information Technology, Kohat University of Science and Technology, Pakistan. He has done his PhD from the School of Engineering and Design, Brunel University, UK. His research mainly focuses on wireless broadband network architecture, security and privacy, security threats and mitigating techniques.

Jamal Abdul Nasir

Jamal Abdul Nasir is working as Assistant Professor in Institute of Computing & Information Technology, Gomal University, D.I.Khan since 1988. Currently he is doing his Ph.D. in CIIT, Islamabad. His area of expertise is Data Mining and Artificial Intelligence. He has published his research in reputed International Journals. He is a one of the pioneers of Information technology in the country.