A Fuzzy Model of Software Project Effort Estimation

Oumout Chouseinoglou *
Hacettepe University, Faculty of Engineering,
Department of Industrial Engineering, 06800 Ankara, Turkey
E-mail: uhus@hacettepe.edu.tr
*Corresponding author

Özlem Müge Aydın
Hacettepe University, Faculty of Engineering,
Department of Industrial Engineering, 06800 Ankara, Turkey
E-mail: ozlemaydin@hacettepe.edu.tr

Abstract

Software project effort estimation is defined as one of the most difficult tasks in software engineering, embodying extensive uncertainty and vagueness in estimation parameters. In order to deal with this uncertainty, it has been suggested that the fuzzy approach can be employed in the effort estimation process. This work in progress paper is a proof of the concept that the estimation approach can be fuzzified. The proposed fuzzy estimation method is applied on a clustered ISBSG 11 dataset and it is concluded that the obtained results are acceptable.

Keywords: Software effort estimation, Fuzzy estimation, Software projects, COCOMO models

1. Introduction

Due to their special nature and complex processes, estimating the work-effort, cost and schedule of software development projects is one of the most difficult tasks in software project management. Software engineering, which can be defined as the application of a systematic, disciplined, and quantifiable approaches to the areas related to software, is trying to address these estimation difficulties. With that respect, several cost estimation models have been developed with the aim of constructively modeling the development processes and accurately predicting the cost of developing software. An exhaustive list of the different software effort and cost estimation studies is given by Jorgensen and Shepperd (2007).

Among the proposed effort estimation models, the most common ones are the algorithmic models, such as COCOMO and function points. As stated by Shepperd and Schofield (1997), the general structure of the model is in the form of
\[ e = a_0 S^{\alpha_1} \]  

(1)

where \( e \) is effort, \( S \) is size, typically measured as lines of code (LOC) or estimated as function points (fp), \( a_0 \) is a productivity parameter and \( \alpha_1 \) is an economies or dis-economies of scale. \( fp \) are a unit of measurement used in software engineering to express the amount of business functionality a software program is intended to provide to a user and they can be estimated prior to the development of the software program with the use of different counting methods, the most known being the COSMIC, NESMA and IFPUG.

Nevertheless, as stated by Xu and Khoshgoftaar (2004), no model has proven to be consistently successful in providing accurate effort and cost estimations, and despite all the efforts undertaken by the software engineering discipline, the estimation problems still exist today, resulting in delayed and over budget software. The main reason for this is that the inputs of any effort estimation formula, like the one given in Eq (1), are vague and are result of informed guessing rather than exact measurements (Musilek et al., 2000). Moreover, the information about the effort that is required to complete the software is often uncertain, imprecise or incomplete (Xu and Khoshgoftaar, 2004). Therefore, the vagueness and uncertainty in the effort estimation inputs necessitates the utilization of alternative approaches, such as fuzzy models. The sources of uncertainty in software cost and effort estimation models, and how a software project can be described as a fuzzy set are given by Musilek et al. (2000).

This paper is the initial part of a larger study aiming to develop a complete fuzzy model to estimate software project effort by utilizing fuzzy approach in all processes and parameters of the estimation. With respect to the overall aim, this study is a proof-of-concept, presenting that fuzzy approach even in a simple effort estimation model does not only generate acceptable results, but also provides the project management with a fuzzy number that can be used as the range of the expected effort, instead of a single value. The proposed approach is empirically validated in the International Software Benchmarking Standards Group (ISBSG) 11 dataset and the results are presented in detail.

The present paper is organized as follows: Section 2 presents a literature review regarding the use of fuzzy logic approaches in software effort and cost estimation models. Section 3 describes the fuzzy arithmetic and the model evaluation and quality measures that have been used in this paper. Section 4 details the fuzzy effort estimation proposed and lists the results obtained from the empirical evaluation. The last section presents conclusion and directions for future work.

2. Related work

There are several examples with respect to the use of fuzzy approaches and logic in software effort and cost estimation literature. Xu and Khoshgoftaar (2004) propose a fuzzy identification cost estimation model to deal with linguistic data, and automatically generate fuzzy membership functions and rules. By using the COCOMO81 project database, the authors cluster the project data with the use of fuzzy c-means and use as
inputs to the proposed model the cost driver attributes of the COCOMO model. By using a set of rules they have generated with the use of the Takagi-Sugeno models they calculate the values of the categories and subsequent membership functions they have devised. Finally they extract a crisp value from the fuzzy sets, similar to the defuzzification process, and they use the “Centroid of Areas” method to calculate the defuzzification values. The authors investigate 63 project data from the COCOMO81 database and they empirically cluster them in 5 clusters. They conclude that the cost estimation accuracy of the fuzzy models generated by the proposed approach is significantly better than that of the COCOMO models. Musilek et al. (2000) propose the f-COCOMO model, a fuzzy set-based generalization of the COCOMO model. In f-COCOMO, instead of using a single number as the software size, the authors propose the use of a fuzzy number, which in return yields to another fuzzy number as the cost estimate. The authors conduct an analysis of their model with different membership functions such as the triangular and the parabolic fuzzy sets and they further implement the use of fuzzy numbers as parameters to the f-COCOMO. They experiment with 63 projects and propose that the fuzzy logic approach needs to be applied in other software cost and effort studies, such as the fp. Aroba et al. (2008) have proposed the use of fuzzy clustering for the development of segmented software cost estimation models, where a software project may belong to more than one segments. Their approach is tested on the ISBSG dataset, where they report their findings for clusters in the size of 11, 15 and 20.

Idri et al. (2001) propose the fuzzy analogy to be used to estimate the cost of software by providing a solution to the vagueness and impreciseness of the software attributes. Estimation by analogy is a four step process where first the similar cases are retrieved, the information gathered from them are reused, the proposed solution is revised and finally some of the parts of this experience are retained to be used in future projects. The authors use fuzzy logic and linguistic quantifiers in reasoning by analogy to estimate the effort of software projects and validate their approach by conducting an experiment on the COCOMO dataset. Similarly Azzeh et al. (2011) propose an analogy-based software effort estimation using fuzzy numbers, namely Generalized Fuzzy Number Software Estimation. They compute the similarity between two generalized fuzzy numbers based on their geometric distances, center of gravities and height of the generalized fuzzy numbers, and use fuzzy c-means to cluster the existing software project data. The estimations are conducted with the use of generalized fuzzy number operations and the effort of a project is estimated as a fuzzy number which is defuzzified with the method of center of gravity. The authors conduct empirical evaluations with the use of jack-knifing in benchmark software data from the ISBSG, Desharmais, Kemerer, Albrecht and COCOMO datasets. Azzeh et al. (2010) have also proposed one other estimation by analogy model that incorporates fuzzy set theory and grey relational analysis. In this model fuzzy set is employed to reduce uncertainty in distance measure between two tuples whereas the grey relational analysis is utilized as a problem solving method to assess the similarity between the tuples with a number of features. The authors compare their results with case-based reasoning, multiple linear regression and artificial neural networks, using the datasets given in their work (Azzeh et al. 2011).
Lopez-Martin et al. (2008) compare three personal fuzzy logic models to estimate the effort of small software programs, namely triangular, trapezoidal and Gaussian membership functions, with linear regression model. They develop the fuzzy logic and linear regression models using the data gathered from 105 small programs, and then the estimations generated by these models are compared with each other using 20 small programs.

3. Fuzzy arithmetic and model validation measures

For a triangular fuzzy number $M=(m, \alpha, \beta)$ let $m$ be the mean value, and $\alpha, \beta$ be the left and right spreads, respectively. Membership function of $M$ can be written as,

$$\mu_M(x) = \begin{cases} L\left(\frac{m-x}{\alpha}\right), & x \leq m \\ R\left(\frac{x-m}{\beta}\right), & x \geq m \\ 0, & \text{otherwise} \end{cases}$$

where $\alpha, \beta > 0$. Bansal (2010) defines the multiplication of two fuzzy numbers $M=(m, \alpha, \beta)$ and $N=(n, \gamma, \delta)$ as,

$$M \otimes N \equiv (mn, m\gamma + n\alpha - \alpha\gamma, m\delta + n\beta + \beta\delta)$$

and exponentiation function of two fuzzy numbers as,

$$M^N = (m^n, m^n - (m - \alpha)^{n-\gamma}, (m + \beta)^{n+\delta} - m^n)$$

The indicators to be used by software practitioners when comparing the results of different models are given by Kitchenham et al. (2001), and are used in a variety of studies investigated in this paper (Aroba et al. 2008, Azzeh et al. 2011, Azzeh et al. 2010, Idri et al. 2002).

Magnitude Relative Error (MRE) computes the absolute percentage of error between actual effort ($ea$) and estimated effort ($ee$), for each investigated project, as shown in Eq (5).

$$MRE_i = \frac{|ea_i - ee_i|}{ea_i}$$

On the other hand, Magnitude Error Relative (MER) is given in Eq (6).

$$MER_i = \frac{|ea_i - ee_i|}{ee_i}$$
Mean Magnitude Relative Error (MMRE) calculates the average of MRE over all investigated items \((n)\), as shown in Eq (7). Similarly the Mean Magnitude of Error Relative (MMER) is given in Eq (8).

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

\[
MMER = \frac{1}{n} \sum_{i=1}^{n} MER_i
\]  

Finally, \(PRED(q)\) is used to count the percentage of estimates that fall within less than or equal to \(q\) of the actual values.

\[
PRED(q) = \frac{\lambda}{N}
\]  

where \(\lambda\) is the number of projects where \(MRE_i \leq q\) and \(N\) is the number of all estimates.

Aroba et al. (2008), referencing Conte et al. (1986), state that to evaluate the performance of a given model, a model whose \(MMRE \leq 0.25\) and \(PRED(0.25) \geq 0.75\) is considered to be a good one. In general, an estimation model with lower MMRE and higher \(PRED(q)\) can be interpreted that its derived estimates are more accurate than other models.

4. Fuzzy numbers for software project effort estimation

Crisp effort estimation function given in Eq (1) is calculated as a function of \(fp\), assuming that \(fp\) value is exactly known. However, \(fp\) mostly depend on imprecise software attributes (Xu and Khoshgoftaar, 2004). In order to overcome this uncertainty, fuzzy logic is inserted into the model. As the main vagueness come from \(fp\) values, \(fp\) are defined by triangular fuzzy numbers and denoted as \(\tilde{fp}\). Hence the estimated effort is achieved as a fuzzy number and denoted as \(\tilde{e}\). Writing Eq (10) for fuzzy effort estimation, Eq (1) is rewritten as,

\[
\tilde{e} = a_0 \tilde{fp}^{a_1}
\]  

In this study, in addition to \(fp\) values, \(a_o\) and \(a_1\) parameters are also fuzzified due to their confidence intervals, which eventually give symmetric triangular fuzzy numbers. The final fuzzy effort estimation model is defined in the form of,

\[
\tilde{e} = \tilde{a}_0 \tilde{fp}^{\tilde{a}_1}
\]
5. Model adequacy checking

The described fuzzy methodology was applied over the ISBSG 11 dataset, to empirically validate its applicability. ISBSG dataset contains 5052 software projects of various types, with 118 attributes per project. As the ISBSG projects widely vary with respect to these attributes, in order to obtain a uniform dataset only the projects whose \( fp \) count is IFPUG and unadjusted \( fp \) rating is classified as A were selected, resulting to 2257 software projects. Moreover, as obtaining a single estimation function would not be appropriate for that final set of projects, prior to estimating the effort function, data was clustered according to both \( fp \) and effort values. SPSS 17 was used to cluster data set by \( k \)-means methodology. Number of clusters was empirically defined as 20 and 50 iterations were conducted at the initiation. Within the determined 20 clusters, the ones including at least 20 observations were taken into consideration (totally 2207 projects) and thus effort estimates depending on six clusters were calculated by non-linear regression model estimation in SPSS. Crisp parameter estimations \( a_0 \) and \( a_1 \) are presented in the third and fifth columns of Table 1. As stated previously, due to the aforementioned uncertainty, \( fp \) values were considered as fuzzy numbers. In fuzzifying the \( fp \) in data set, the spreads were determined by extending the mean (crisp) \( fp \) by \( \pm 0.05 \times fp \). Thus, crisp \( fp \) were converted to symmetric triangular fuzzy numbers by using the extension principle of fuzzy logic.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Number of observations</th>
<th>( a_0 )</th>
<th>( a_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>mean</td>
<td>spread</td>
</tr>
<tr>
<td>1</td>
<td>79</td>
<td>16180.877</td>
<td>3867.681</td>
</tr>
<tr>
<td>2</td>
<td>139</td>
<td>10276.013</td>
<td>1934.095</td>
</tr>
<tr>
<td>3</td>
<td>599</td>
<td>2508.763</td>
<td>319.284</td>
</tr>
<tr>
<td>4</td>
<td>1051</td>
<td>312.112</td>
<td>52.848</td>
</tr>
<tr>
<td>5</td>
<td>318</td>
<td>6448.994</td>
<td>1040.687</td>
</tr>
<tr>
<td>6</td>
<td>21</td>
<td>25349.009</td>
<td>6328.847</td>
</tr>
</tbody>
</table>

Spreads of each fuzzy number according to clusters are also given in Table 1.

For each cluster, both crisp and fuzzy effort estimations were calculated using each cluster’s parameter estimates. Quality of the estimators were measured by MMRE, MMER and PRED(0,25) values and the results for crisp and fuzzy estimation models are given in Table 2.

At the last step of this study, six clusters were merged and accuracy errors, calculated by estimations according to their own estimators, were examined. Quality measures for overall data are presented in Table 3. It is seen in Table 3 that, fuzzy model is some worse than the crisp one, however, it is not such an inadequate model to be avoided.
Especially, in cases where lack of data or ill-defined data exists, fuzzy estimation model can be used to get satisfactory results.

Table 2. Estimation quality for crisp and fuzzy estimation models

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Measure</th>
<th>CrispModel</th>
<th>Fuzzy Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MMRE</td>
<td>0,1056</td>
<td>0,1244</td>
</tr>
<tr>
<td></td>
<td>MMER</td>
<td>0,1056</td>
<td>0,1116</td>
</tr>
<tr>
<td></td>
<td>PRED(0,25)</td>
<td>1,0000</td>
<td>0,9494</td>
</tr>
<tr>
<td>2</td>
<td>MMRE</td>
<td>0,1308</td>
<td>0,1420</td>
</tr>
<tr>
<td></td>
<td>MMER</td>
<td>0,1293</td>
<td>0,1311</td>
</tr>
<tr>
<td></td>
<td>PRED(0,25)</td>
<td>0,9424</td>
<td>0,8633</td>
</tr>
<tr>
<td>3</td>
<td>MMRE</td>
<td>0,2405</td>
<td>0,2464</td>
</tr>
<tr>
<td></td>
<td>MMER</td>
<td>0,2264</td>
<td>0,2245</td>
</tr>
<tr>
<td></td>
<td>PRED(0,25)</td>
<td>0,5492</td>
<td>0,5525</td>
</tr>
<tr>
<td>4</td>
<td>MMRE</td>
<td>1,4277</td>
<td>1,4637</td>
</tr>
<tr>
<td></td>
<td>MMER</td>
<td>0,4936</td>
<td>0,4833</td>
</tr>
<tr>
<td></td>
<td>PRED(0,25)</td>
<td>0,2683</td>
<td>0,2712</td>
</tr>
<tr>
<td>5</td>
<td>MMRE</td>
<td>0,1771</td>
<td>0,1858</td>
</tr>
<tr>
<td></td>
<td>MMER</td>
<td>0,1724</td>
<td>0,1718</td>
</tr>
<tr>
<td></td>
<td>PRED(0,25)</td>
<td>0,7610</td>
<td>0,7673</td>
</tr>
<tr>
<td>6</td>
<td>MMRE</td>
<td>0,0500</td>
<td>0,0739</td>
</tr>
<tr>
<td></td>
<td>MMER</td>
<td>0,0498</td>
<td>0,0667</td>
</tr>
<tr>
<td></td>
<td>PRED(0,25)</td>
<td>1,0000</td>
<td>1,0000</td>
</tr>
</tbody>
</table>

Table 3. Estimation quality of overall data

<table>
<thead>
<tr>
<th>Measure</th>
<th>Crisp Model</th>
<th>Fuzzy Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMRE</td>
<td>0,7832</td>
<td>0,8048</td>
</tr>
<tr>
<td>MMER</td>
<td>0,3337</td>
<td>0,3288</td>
</tr>
<tr>
<td>PRED(0,25)</td>
<td>0,4912</td>
<td>0,4875</td>
</tr>
</tbody>
</table>

Table 2 and 3 show that the calculated accuracy measures for both models are similar to each other. Although the proposed fuzzy model could not serve better results in all clusters, it should be kept in mind that fuzzy model achieved these similarities by using extended values. This extension eases decision makers in determining the \( fp \) and the other parameters. Moreover, within the 2207 examined software projects, 56.54% of the
actual efforts are taking place within the estimated fuzzy effort interval. With all these taken into account, the fuzzified model is concluded to be sufficient.

6. Conclusions and future work

Effort estimation is one of the most significant fields in software project management as it includes extensive uncertainty and vagueness, with various types of software projects, thus making generalizations impossible. $fp$, which are used as an input for the estimation model, consist of some imprecise attributes. In order to decrease any kind of vagueness related to the effort estimation, fuzzy logic can be inserted into the model and even though it does not guarantee to give the best result, acceptable results can be achieved.

This work in progress study, being an initial step in the development of a larger fuzzy effort estimation approach, aimed to seek an acceptable fuzzy estimation model based on fuzzy $fp$, by using fuzzy parameter estimates for clustered ISBSG data. Even though it is noticeable to conclude that the fuzzy effort estimation model can substitute the crisp estimation model, this undertaking was successful only with the clustered data in this study. Therefore, prior to estimations, one has to decide on the most appropriate cluster.

Based on these findings, the current research is on developing a fuzzy effort estimation model where the fuzzy effort function parameters and the fuzzy $fp$ number are calculated based on the project attributes. Moreover, a more general fuzzy estimation model can be investigated for the whole data, independent of clusters. Additionally, the results of this study are efficient for crisp clustered data, whereas a project could be placed in a number of clusters at the same time. In such cases, fuzzy clustering methods can be used before estimating the project effort.

References


