On the Use of Fuzzy Regression in Parametric Software Estimation Models: Integrating Imprecision in COCOMO Cost Drivers

F. JAVIER. CRESPO, MIGUEL-ÁNGEL SICILIA, JUAN J. CUADRADO
Computer Science Department
Carlos III University
Avda. Universidad, 30 – 28911 Leganés (Madrid)
SPAIN

Abstract: Parametric software estimation techniques make use of mathematical models elaborated from regression techniques to obtain effort-of-development estimates. As a matter of fact, the values of the cost drivers that act as variables in these models are inherently vague and uncertain, since they’re in many cases elaborated from human judgments about highly abstract concepts (like ‘required reliability’) or even from collective attitudinal characterizations (like ‘analyst capability’). In this paper, fuzzy regression techniques based on fuzzification of input values are explored as an alternative to conventional regression for obtaining estimating equations. The available COCOMO-81 project database and the Furea fuzzy regression tool are used as a case study, emphasizing more realistic approaches to the expression of widely used cost driver values.

Key-Words: Software effort estimation, fuzzy regression, parametric estimation models, fuzzy variables, COCOMO.

1 Introduction

Parametric software estimation models use mathematical expressions to derive effort of development estimates from a number of driving variables [9]. These variables include software size – measured in terms of lines of code or function points – and also a number of other factors influencing development time. Estimating formulas are elaborated from databases of past projects that serve as experimental data for classical regression techniques. Nonetheless, a fundamental limitation of existing models like COCOMO [1] is that they force the use of crisp real numbers to express the values of the variables, despite the fact that they are inherently fuzzy. The sources of that fuzziness include both the imprecision of the limited set of linguistic labels that are used to assess them, and also the unavoidable uncertainty of human approximate judgments about highly abstract concepts (like required reliability, complexity or technological volatility) or collective human capabilities (like experience or overall ability in performing specific work roles). In consequence, the exploration of fuzzy regression techniques [1] may eventually lead to more realistic and appropriate estimating models. In this paper, we describe a first step in that direction. Concretely, the original COCOMO-81 project database is used to explore fuzzy regression in an attempt to delineate future related directions of research. Related work on fuzziness regarding software estimation includes the generalization of input model parameters through fuzzy numbers [6], the use of similarity relations in estimating by analogy [7], and the fuzzy estimation of function points [8]. Nonetheless, none of the mentioned research works is related with the elaboration of parametric models through generalized-for-fuzziness regression techniques.

Fuzzy regression analysis approaches [1] can be roughly categorized in two groups. Classical fuzzy regression approaches initiated by Tanaka et al. [3] are based on the assumption that deviations are due to the fuzziness of the parameters of the model. A more recent approach reported in [4] departs from a different position by considering fuzziness in the experimental points while using a crisp model. In this work, the second approach is adopted, since a thorough examination of the variables that are commonly used in parametric software estimation reveal that both imprecision and uncertainty play a central role in input value formulation, and, consequently, it’s reasonable to assume that a large part of estimation deviations may come from imperfection in input assessment.

The rest of this paper is structured as follows. In Section 2, we describe the fuzzification of input variables and the results of using f-regression on the data of the original COCOMO model. Discussion and analysis of the obtained model are provided in
Section 3. Finally, conclusions and some future research directions are provided in Section 4.

2 Fuzzy Regression in Parametric Software Estimation

Our initial exploration of fuzzy regression takes the COCOMO-81 project as a set of 63 experimental points. Since we are focused on the suitability of fuzzy regression algorithms, the assumptions, model and input selection for estimating are simply adopted from the original COCOMO model [2].

The intermediate COCOMO model, as described in [2], uses fifteen predictor variables that are incorporated in the estimating formula (1) that obtains the effort \( e \) required for a project (measured in man-months) given the estimated size of the product (measured in lines of source code). Constant \( a \) is organization dependant, although the values used in COCOMO for stereotypes classes of projects can be used: 2.4 for simple, well-understood projects, 3.0 for moderate complexity projects where team members have limited experience in related systems, and 3.6 for complex projects. Constant \( b \) is also dependant on product complexity, and takes the values 1.05, 1.12 and 1.20 for the just described classes of projects.

\[
e = a \cdot \text{size}^b \cdot M
\]

Parameter \( M \) in (1) is the product of predictor variables or cost drivers (2) ranked on linguistic ordinal scales. Possible values for those variables and the rationale for selecting values for a given project are summarized in Table 1; linguistic scales are: very low (VL), low (L), nominal (N), high (H), very high (VH) and extremely high (EH).

\[
M = \prod_{i=1}^{15} c_i
\]

In consequence, the first step towards fuzzy regression on experimental points requires an assessment of the kind and impact of uncertainty on each of the cost drivers.

In the rest of this section, we first describe the rationale for the fuzzy modeling of COCOMO variables, and then we report the results obtained from our experiments with the Furea fuzzy regression tool [5] and a historical database of projects.

2.1 Modeling Fuzziness in Estimation Model Variables

The numeric variable values in Table 1 were selected to enhance the properties of the COCOMO model. As such they do not reflect a concrete accurate measure, but rather an approximate value derived from the use of the corresponding linguistic values. In consequence, at least two sources of imperfection are present in those values. First, the assessments possess linguistic vagueness, and second, their conversion to numeric values force them to an specific distance relationship with each other that is related to the final model properties, but not to the actual value of the variables for each given experimental point. In consequence, we are faced with estimating a reasonable error tolerance for those variables that will be used to select the shape of the fuzzy numbers to be used for the inputs to the fuzzy regression method. In our present work, we have estimated a single upper bound error value for all the variables, the specific consideration of each variable is left to future work.

As can be observed in Table 1, selection imperfection comes from a number of different causes:

(a) RELY is assessed by cognitive comparison with prototype cases (and in consequence, it’s conditioned by human categorization processes – see Rosch [10]). For example “nuclear reactor control system” is an exemplar of the label “very high”,

(b) DATA is estimated heuristically from two imprecise volume quantities,

(c) CPLX, STOR and TIME values are obtained by linguistic estimation (of simplicity of code, expressions and data structures for CPLX and of percentage of resource consumption for the others),

(d) VIRT entails the uncertain estimation of an imprecise time frame,

(e) TURN is a vague categorization of measurable (crisp) computer response times,

(f) ACAP and PCAP are aggregated measures of high-level human competences. AEXP, LEXP and VEXP measures team knowledge in terms of experience expressed in terms of time.

(g) MODP entails value judgments about the ‘modernity’, i.e. adequacy to current practice of development techniques. TOOL also entails a adequacy judgment, in this case, regarding the impact on productivity of computer-aided tools.
Summing up, different estimating domains are used, (from time frames to human and team characteristics, and also including computer and software properties, and suitability estimates of techniques and tools), and the method of estimation is also diverse. In consequence, only upper bound approaches can be considered reasonable.

<table>
<thead>
<tr>
<th>Cost Driver</th>
<th>VL</th>
<th>L</th>
<th>N</th>
<th>H</th>
<th>VH</th>
<th>EH</th>
<th>Value Selection Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELY: Required Software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Comparison with prototypical reliability situations.</td>
</tr>
<tr>
<td>Reliability</td>
<td>.75</td>
<td>.88</td>
<td>1.00</td>
<td>1.15</td>
<td>1.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA: Database size</td>
<td>.94</td>
<td>1.00</td>
<td>1.08</td>
<td>1.16</td>
<td></td>
<td></td>
<td>The ratio database size by program size (D/P)</td>
</tr>
<tr>
<td>CPLX: Product complexity</td>
<td>.70</td>
<td>.85</td>
<td>1.00</td>
<td>1.15</td>
<td>1.30</td>
<td>1.65</td>
<td>Level of complexity of the software expressed in vague terms of complexity of control, computational, device and data operations.</td>
</tr>
<tr>
<td>TIME: Execution time constraint</td>
<td></td>
<td></td>
<td>1.00</td>
<td>1.11</td>
<td>1.30</td>
<td>1.66</td>
<td>Linguistic assessment of the percentage of available execution time expected to be used by the software.</td>
</tr>
<tr>
<td>STOR: Main storage constraint</td>
<td></td>
<td>1.00</td>
<td>1.06</td>
<td>1.21</td>
<td>1.56</td>
<td></td>
<td>Linguistic assessment of the percentage of main storage expected to be used by the software.</td>
</tr>
<tr>
<td>VIRT: Virtual machine volatility</td>
<td>.87</td>
<td>1.00</td>
<td>1.15</td>
<td>1.30</td>
<td></td>
<td></td>
<td>Estimated minor and major change frequencies of the platform.</td>
</tr>
<tr>
<td>TURN: Computer turnaround time</td>
<td>.87</td>
<td>1.00</td>
<td>1.07</td>
<td>1.15</td>
<td></td>
<td></td>
<td>Degree of interactivity of the response time of the development environment.</td>
</tr>
<tr>
<td>ACAP: Analyst capability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Measure of the communication, cooperation and abilities of analysts.</td>
</tr>
<tr>
<td>AEXP: Applications experience</td>
<td>1.29</td>
<td>1.13</td>
<td>1.00</td>
<td>.91</td>
<td>.82</td>
<td></td>
<td>Vague categorization of the level of expertise of the team regarding the type of application.</td>
</tr>
<tr>
<td>PCAP: Programmer capability</td>
<td>1.42</td>
<td>1.17</td>
<td>1.00</td>
<td>.86</td>
<td>.70</td>
<td></td>
<td>Measure of the communication, cooperation and abilities of programmers.</td>
</tr>
<tr>
<td>VEXP: Virtual machine experience</td>
<td>1.21</td>
<td>1.10</td>
<td>1.00</td>
<td>.90</td>
<td></td>
<td></td>
<td>Vague categorization of the level of expertise of the team regarding the platform.</td>
</tr>
<tr>
<td>LEXP: Programming language experience</td>
<td>1.14</td>
<td>1.07</td>
<td>1.00</td>
<td>.95</td>
<td></td>
<td></td>
<td>Vague categorization of the level of expertise of the team regarding the programming language.</td>
</tr>
<tr>
<td>MODP: Use of modern programming practices</td>
<td>1.24</td>
<td>1.10</td>
<td>1.00</td>
<td>.91</td>
<td>.82</td>
<td></td>
<td>Degree of use of “modern programming practices”</td>
</tr>
<tr>
<td>TOOL: Use of software tools</td>
<td>1.24</td>
<td>1.10</td>
<td>1.00</td>
<td>.91</td>
<td>.83</td>
<td></td>
<td>Degree of support given by tools to the development process</td>
</tr>
<tr>
<td>SCED: Required software development experience</td>
<td>1.23</td>
<td>1.08</td>
<td>1.00</td>
<td>1.04</td>
<td>1.10</td>
<td></td>
<td>Approximate percentage of schedule acceleration or stretchout with respect to nominal schedule.</td>
</tr>
</tbody>
</table>

Table 1. Software Development Effort Multipliers

The diversity of imperfection sources leaded us to select a compromise approach: the selection of the degree of fuzziness in fuzzy numbers was determined by experimenting with the parameters \( m \) and \( s \) of \( L \)-type fuzzy numbers (fuzzy generalizations of intervals), searching for the best residual variances. The selected value for the spread of the fuzzy numbers in primary variables was \( s = 0.1 \) (the parameters for transformed variables are computed by the Furea tool for each value). In addition, \( m = 2 \) appeared and worked as a good initial candidate for the shape of the numbers, since it differentiates better close-to-center values than, for example, a value of \( m = 4 \) (see Figure 1).

It should be noted that the model in Figure 1 acts as an upper bound of uncertainty for parameters \( M \) and size in formula (1), and the importance of the cost drivers that are aggregated in \( M \) is that implicitly considered by the selection of parameter values showed in Table 1 (note that the amplitude of the intervals of cost drivers range from 0.95 to 0.13, and the gaps between values from 0.36 to 0.04).

2.2 Results of the Fuzzy Regression

The \( f \)-regression method [4] is based on obtaining a coefficient vector \( a \) for a given crisp model, assuming that experimental points (both inputs and actual outputs) are modeled as fuzzy numbers \( \mu_f(\bar{x}, y) \) modeled as the product \( T \)-norm of the fuzzy numbers
representing inputs and outputs. Then, a similarity measure $M_i(a)$ with function $f$ (for coefficient vector $a$) for a given fuzzy point $Q_i$ is defined (3).\footnote{This $M_i$ must not be confounded with $M$ in formula (1).}

$$M_i(a) = \sup \{ \mu_\phi(x, f(x,a)) \} \quad (3)$$

In addition, an aggregation operator $M$ is used to compute the similarity of a set of fuzzy points to a given coefficient vector for function $f$. We have used the arithmetic mean $MA(a)$ as aggregation operator (4) due to its compensatory effects (experiments with the geometric mean resulted in worse adjustment to project data).

$$MA(a) = \frac{1}{N} \sum_i M_i(a) \quad (4)$$

The optimal coefficient vector $a^*$ is obtained by random search techniques as described in [4].

Following the original COCOMO formula (1), a model with two coefficients $a_0$ and $a_1$ – equivalent to $a$ and $b$ in (1) – were used.

\[ e = 2.482 \cdot \text{size}^{1.193} \cdot M^{1.301} \quad (5) \]

Residual variance for (5) has a value of 0.06748, of the same approximate magnitude to that obtained through non-fuzzy regression methods, concretely, the solution provided by Furea using a conventional Least Square Method (LSM) algorithm has a value of 0.06278.

Figure 2 shows a depiction of the aggregation operator $MA(a)$ with the maximum selected by the regression algorithm, in terms of the possible values for parameters $a_0$ and $a_2$. The cutting plane at 0.1 helps in visualizing local maximums and the point

3 Discussion

Predictive characteristics of equation (5) are comparable to those obtained from classical regression approaches, but it should be taken into account that the database used for the described case study did not preserve the original uncertainty properties of the sources described in Section 2.1, since it was expressed in terms of stereotypical values discussed in [2].

![Figure 2. Aggregation $MA(a)$ in the parameter space of $a_0$ and $a_2$.](image)

Thus, the same ordinal linguistic scale for cost drivers is mapped to different non-regular intervals that suggest a loss of information when casting input variables to the numbers in Table 1. Nonetheless, our experiments regarding the explicit introduction of random small changes – proportional to the interval gaps of each cost driver – in the project database did not come up with a better model.

Nonetheless, if we change the original values of COCOMO in Table 1, respecting intervals around the original numbers to evaluate the goodness of the model, small improvements in the mean magnitude of relative error (MMRE) can be obtained by varying some of the experimental points in a “reasonable” manner. In our model, reasonability can be understood as respecting the fuzzy numbers used by the regression procedure to obtain the model. The use of intervals in COCOMO was first proposed by
Zonglian and Xihui [12], but they did not provide a rationale for the intervals selected in their model.

Figure 3 shows example modifications for several fuzzy numbers in the COCOMO data set. In that Figure, four experimental fuzzy values of $M$ are depicted as $L$-type fuzzy sets with $m=2$ and the spreads computed by the tool. Given these functions, the original crisp experimental values (0.346, 0.832, 1.730 and 2.288, in the center of the functions) can be substituted by the values 0.385, 0.789, 1.754 and 2.279 respectively, obtaining membership values in the corresponding functions above 0.87 (i.e. the changed values are inside the strong $A_{0.8}$ $\alpha$-cut of each fuzzy set). With this “reasonable” changes, an absolute difference reduction from predicted to real effort values of 222.16 is obtained. This finding points out that richer uncertainty modeling of the inputs may eventually lead to better models.

Figure 3. Example fuzzy numbers for variable $M$ with different spreads.

Outlier examination based on the similarity measure of the function with each given fuzzy point leads to considering a set of three points that have extremely low membership values, but its exclusion from the input data do not result in a better model.

4 Conclusions and Future Work

Parametric software estimating models are based on inherently imprecise and uncertain input variables. In this paper, preliminary results on using fuzzy inputs to f-regression have been reported.

The main conclusion is that fuzzy regression is able to obtain estimation models with similar predictive properties than existing basic estimation models.

Nonetheless, further work is required with more recent data, and also more research is needed in the fuzzy formulation of input values, which may eventually result in more accurate models, e.g. cost drivers selected in COCOMO are heterogeneous in uncertainty terms, so that different forms of membership functions would be required for a more realistic modeling.

In addition, linguistic expressions require membership elicitation techniques [11] to come up with more appropriate mathematical devices in the formulation of concrete values, so that experimental studies would be required to measure the actual psychological characteristics of many of the cost drivers used in software estimation.

References:


