Available online at www.scientiaresearchlibrary.com

Scientia Research Library **ISSN 2348-0424**
USA CODEN: JETRB4

Journal of Engineering And Techonology Research, 2014, 2 (1):10-16

(*http://www.scientiaresearchlibrary.com/arhcive.php*)

New Fuzzy Logic Model for Effort Estimation in Software Module Development

Prof. (Dr.) Vishwa Nath Maurya[1] , Dr. Rajender Kumar Bathla[2] , Er. Avadhesh Kumar Maurya[3] , Diwinder Kaur Arora[4]

[1]Department of Mathematics, School of Science & Technology, University of Fiji, Fiji
[2]Department of Computer Science & Engineering
Haryana Institute of Engineering & Technology, Kaithal, Haryana, India
[3]Department of Electronics & Communication Engineering
Lucknow Institute of Technology, G.B. Technical University, Lucknow-226002, India
[4]Central Reserve Police Force, Ministry of Home Affairs, Govt. of India

ABSTARCT

In the present paper, a new fuzzy logic approach for software development time estimation is proposed. In section 1, an introduction along with work done by previous researchers in the relevant field of software estimation is described. In section 2, fuzzy logic approach is presented. Section 3 addresses the parameters analysis under our present consideration. The methodology used here is based on empirical model studied by some previous noteworthy researchers. Using the fuzzy logic approach, results are investigated and are presented in section 5. As a numerical illustration, the membership function plots corresponding to table 5.1 are shown in figures 5.1-5.4. Here, the advantages of fuzzy logic and good generalization are obtained. The major difference between our proposed work and previous works is that two-sided Gaussian membership function in fuzzy technique has been used for software development time estimation and then it is validated with gathered data. Finally, some significant conclusions are drawn in section 6.

Keywords: Software engineering, software metric, software effort estimation, Fuzzy logic approach, COCOMO model

INTRODUCTION

Challenges for software developers are predicting the development effort for a software system based on developer abilities, size, complexity and other metrics for the last decades. The ability to give a good estimation on software development efforts is required by the project managers. Most of the traditional techniques such as function points, regression models, COCOMO, etc, require a long-term estimation process. New paradigms as Fuzzy Logic may offer an alternative for this challenge. Software metric and especially software estimation is based on measuring of software attributes which are typically related to the product, the process and the resources of software development. This kind of measuring can be used as parameters in project management models which provide assessments to software project managers in managing software projects to avoid problems such as cost overrun and behind the schedule. One of the most widely research areas of software measurement is software effort estimation. Software effort estimation models are divided into two main categories: algorithmic and non-algorithmic. For more details, we refer [4], [5] & [15].

The most popular algorithmic estimation models include Boehm's COCOMO (1981), Putnam's SLIM (1978) and Albrecht's Function Point. These models require as inputs, accurate estimate of certain attributes such as line of code (LOC), complexity and so on which are difficult to obtain during the early stage of a software development project. The algorithmic estimation models also have difficulty in modeling the inherent complex relationships between the contributing factors. The algorithmic estimation models are unable to handle categorical data as well as lack of reasoning capabilities Saliu et al. (2004). The limitations of algorithmic models led to the exploration of the non-algorithmic techniques which are soft computing based.

These include artificial neural network, evolutionary computation, fuzzy logic models, case-based reasoning, and combinational models and so on. Artificial neural network are used in effort estimation due to its ability to learn from previous data, for more details, we refer ([6] & [7]). It is also able to model complex relationships between the dependent (effort) and independent variables (cost drivers). In addition, it has the ability to generalize from the training data set thus enabling it to produce acceptable result for previously unseen data. Most of the work in the application of neural network to effort estimation made use of feed-forward multi-layer perception, back propagation algorithm and sigmoid function [6]. Selecting good models for software estimation is very critical for software engineering. In the recent years many software estimation models have been developed, we refer ([2], [6], [8], [9], [10], [12] & [13]).

MacDonnell et al. (1999) compared results using function point analysis, regression techniques, feed-forward neural network and fuzzy logic in software effort estimation. Their results showed that fuzzy logic model achieved good performance, being outperformed in terms of accuracy only by neural network model with considerably more input variables. Also they developed FULSOME (Fuzzy Logic for Software Metrics) which is a set of tools that helps in creating fuzzy model. Fei and Lui (1992) introduced the f-COCOMO model which applied fuzzy logic to the COCOMO model for software effort estimation. Since there was no comparison of the results between the f-COCOMO and other effort estimation models in their study, the estimation capability of the former is unknown. Roger (1993) also proposed a fuzzy COCOMO model which adopted the fuzzy logic method to model the uncertainty of software effort drivers, but the effectiveness of the proposed model is not mentioned. McDonnel (1999) and Idri (2002) defined further a fuzzy set for the linguistic values of each effort driver with a trapezoid-shaped membership function for the fuzzy COCOMO model. The effort multipliers in the original COCOMO model were obtained from the fuzzy sets. This fuzzy COCOMO model was less sensitive to the software effort drivers as compared to the intermediate COCOMO81. In 2004, Xue and Khoshgoftaar [14] presented a fuzzy identification effort estimation modeling technique to deal with linguistic effort drivers, and automatically generated the fuzzy membership functions and rules by using the COCOMO81 database. The proposed fuzzy identification model provided significantly better effort estimates than the original three COCOMO models, i.e., basic, intermediate, and detailed.

MATERIALS AND METHODS

Fuzzy Logic Approach

Many of the problems of the existing effort estimation models can be solved by incorporating fuzzy logic. Since fuzzy logic foundation by Lotfi Zadeh in 1965, it has been the subject of important investigations [15]. It is a mathematical tool for dealing with uncertainty and also it provides a technique to deal with imprecision and information granularity [12]. The fuzzy logic model uses the fuzzy logic concepts introduced by Lotfi Zadeh [15]. Fuzzy reasoning consists of three main components [8], [12], [14], [15]; fuzzification process, inference from fuzzy rules and defuzzification process. In Fuzzification process, the objective term is transformed into a fuzzy concept. The membership functions are applied to the actual values of variables to determine the confidence factor or membership function (MF). Fuzzification allows the input and output to be expressed in linguistic terms. Inference process involves defuzzification of the conditions of the rules and propagation of the confidence factors of the conditions to the conclusion of the rules. A number of rules will be fired and the inference engine assigned the particular outcome with the maximum membership value from all the fired rules.

Parameters Analysis

The main parameter for the evaluation of cost estimation models is the Magnitude of Relative Error (MRE) [13] which is defined as following;

$$MRE_i = \frac{|ActualEffort_i - PredictedEffort_i|}{ActualEffort_i} \tag{3.1}$$

The MRE value is calculated for each observation whose effort is predicted. The aggregation of MRE over multiple observations (N), can be achieved through the Mean MRE (MMRE) as following;

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i \tag{3.2}$$

Methodology Used

Here, the empirical study carried out is based on the methodology used by Lopez-Martin et al. They used the sets of system development projects. The development time of forty-one modules and for each module, coupling (Dhama), complexity (McCabe), and lines of code metrics were registered, all programs were written in Pascal, hence, module categories belong to procedures or functions. The development time of each of the forty-one modules were registered including five phases: requirements understanding, algorithm design, coding, compiling and testing [2], [8].

RESULTS AND DISCUSSION

A subset of projects, 10 projects, is selected. Statistics and a brief description related to each module are depicted in Table 1 which is prepared by Lopez-Martin et al. [8, 14]. In Table 5.1, modules description and metrics, MC: McCabe Complexity, DC: Dhama Coupling, LOC: Lines of Code, DT: Development Time (minutes) is given.

Implementing a fuzzy system requires that the different categories of the different inputs be resented by fuzzy sets, which in turn is presented by membership functions. A natural membership function type that readily comes to mind is the two-sided Gaussian membership function.

A two-sided Gaussian membership function, defined by minimum (a), maximum (c) and modal (b) values, that is MF(a, b, c) where scalar parameters (a, b, c) are defined as follows:

$$MF(x) = 0 \text{ if } x < a$$
$$MF(x) = 1 \text{ if } x = b$$
$$1MF(x) = 0 \text{ if } x > c$$

Following six rules are suggested in [13]:

(i) If complexity is low and lines of code (LOC) is small then DT is low
(ii) If complexity is average and size (LOC) is medium then DT is average
(iii) If complexity is high and size (LOC) is big then DT is high
(iv) If coupling is low then DT is low
(v) If coupling is average then DT is average
(vi) If coupling is high then DT is high

The membership function plots corresponding to Table 5.1 are shown in figures 5.1-5.4. The MRE and MMRE are calculated in view of equations (3.1) and (3.2) respectively. The result in our present study shows that the value of MMRE (Mean of Magnitude of Relative Error) applying Fuzzy Logic was substantially lower than MMRE applying by other fuzzy logic models.

Table 5.1: Modules description and metrics

S. No	Module Description	MC	DC	LOC	DT
1	Calculates Value	1	0.25	4	13
2	Insert a new element in a linked list	1	0.25	10	13
3	Calculates a value according to normal distribution equation	1	0.333	4	9
4	Calculates the variance	2	0.083	10	15
5	Generates range square	root	2	0.111	23
6	Determines both	Min and	Max. value	stored linked list	2
7	Turns each linked list	value into its z value	2	0.125	9
8	Copies a list of values	from a file to an array	2	0.125	14
9	Determines parity of a	number	2	0.167	7

10	Defines segment limits	2	0.16 7	8	18

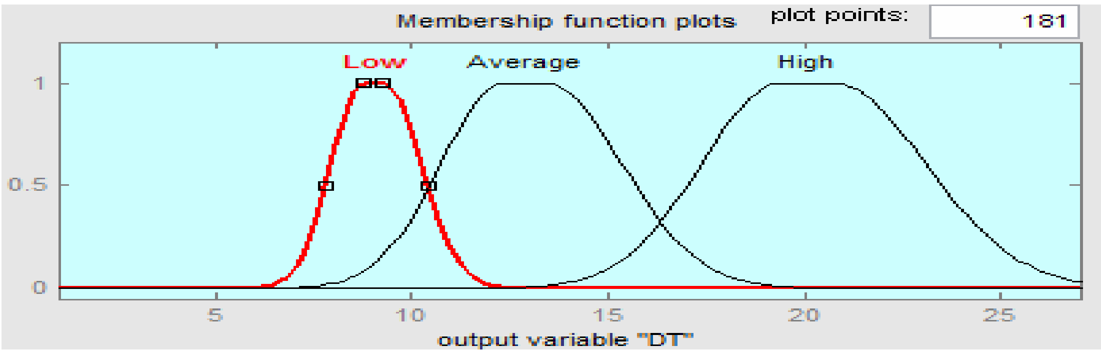


Figure 5.1: Development Time Plot (input)

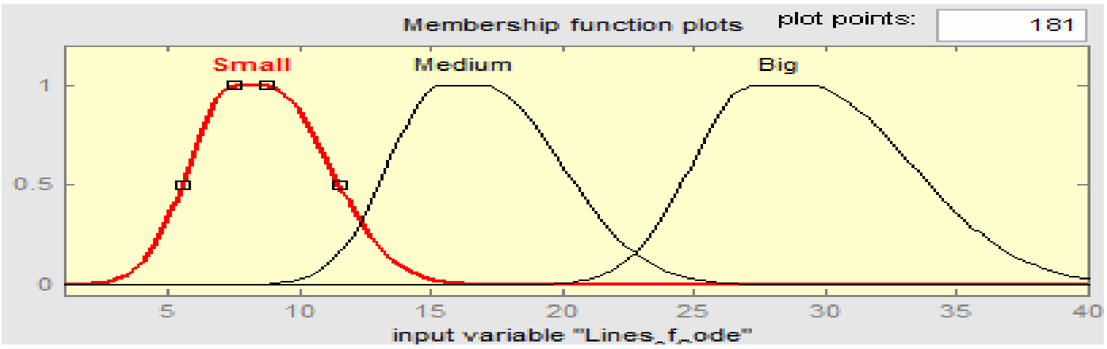
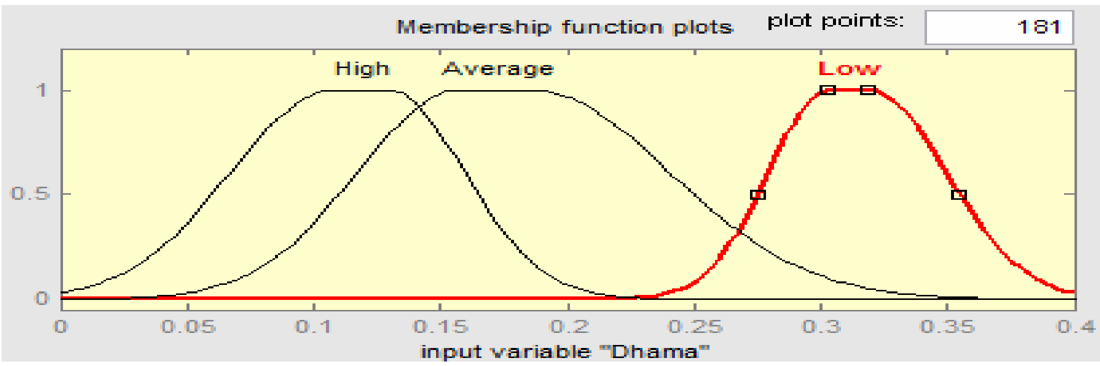


Figure 5.2: Development Time Plot (input)



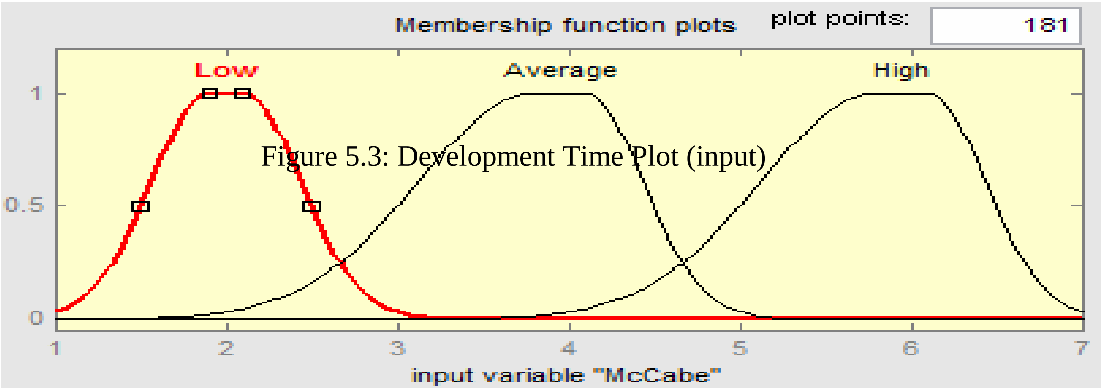


Figure 5.3: Development Time Plot (input)

Figure 5.4: Development Time Plot (input)

CONCLUSION

This paper describes an enhanced fuzzy logic model for the estimation of software development effort. In this paper, we have proposed a new fuzzy logic approach for estimating of software projects development time. The advantages of fuzzy logic and good generalization have been demonstrated well. The results explored here shows following conclusive observations;

- The value of MMRE (Mean of Magnitude of Relative Error) applying fuzzy logic was substantially lower than MMRE applying by other fuzzy logic models.
- The major difference between our work and previous works is that two-sided Gaussian membership function in fuzzy technique is used for software development time estimation and then it's validated with gathered data.
- The main benefit of this model is its good interpretability by using the fuzzy rules.
- Another great advantage of the present research is that it can be put together expert knowledge (fuzzy rules) project data into one general framework that may have a wide range of applicability in software estimation.

REFERENCES

[1] Boehm B.W., Software Engineering Economics, Englewood Cliffs, NJ, Prentice Hall,**1981**
[2] Burgess C.J., Lefley M.. Can Genetic Programming improve software effort estimation? A comparative evaluation, Information and Software Technology, Vol. 43, No.14, **2001**, pp. 863-873
[3] Fei Z, Liu X., Fuzzy constructive cost model in software engineering (.f-COCOMO), IEEE International Conference on Fuzzy Systems, **1992**, pp 331-337.
[4] Fenton N.E., Pfleeger S.L., Software metrics, A rigorous and practical approach, 2nd Edition, PWS Publishing Company, Thomson Publishing, Boston, **1997**
[5] Gray A.R., MacDonell S.G., Applications of fuzzy logic to software metric models for development effort estimation, Fuzzy Information Processing Society, 97th Annual Meeting of the North American, 21- 24th September **1997**, pp. 394-399.
[6] Idri A., Abran A., Khoshgoftaar T.M., Estimating software project effort by analogy based on linguistic values, Proceedings of the Eighth IEEE Symposium on Software Metrics, 4-7th June **2002**, pp. 21 -30.
[7] Idri A., Khoshgoftaar T.M., Abran A, Can neural networks be easily interpreted in software cost estimation?, IEEE Trans. Software Engineering, Vol. 2, **2002** b, pp. 1162-1167

[8] MacDonell S.G., Gray A.R., Calvert J.M., Fuzzy Logic for Software Metric Practitioners and Researchers (FULSOME), Proceedings of the 6th International Conference on Neural Information Processing ICONIP!99, ANZIIS!99, ANNES!99 and ACNN!99, Perth, Western Australia, IEEE Computer Society Press, **1999**, pp.308-313.
[9] Maurya V.N. & Bathla Rajender Kumar, Analytical study on manual vs. automated module testing using critique on overly simplistic cost model, International Journal of Electronics Communication & Electrical Engineering, Algeria, Vol.2, Issue 1, **2012**, pp. 203-216
[10] Maurya V.N., Bathla Rajender Kumar, Maurya Avadhesh Kumar & Arora Diwinder Kaur, A dynamic innovative scenario of automated regression testing using software testing tool, International Journal of Information Technology and Operations Management, Academic and Scientific Publisher, New York, USA, Vol. 1, No.1, **2013**, pp. 1-10
[11] Putnam L.H., A general empirical solution to the macro software sizing and estimating problem. IEEE Transactions on Software Engineering, SE-4(4), **1978**, pp 345-361
[12] Roger J.S., Adaptive network based fuzzy inference systems (ANFIS), IEEE Trans. Syst. Man Cybern 3(3), **1993**, pp. 665-685
[13] Saliu M.O., Ahmed M. and Ghamdi J. Al, Cowards adaptive soft computing based software effort prediction, Fuzzy Information, Proceedings of IEEE Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), 27-30th June **2004**, Vol.1, pp. 16-21
[14] Xu Z. and Khoshgoftaar, T. M. Identification of fuzzy models of software cost estimation, **2004**
[15] Zadeh L.A., Fuzzy Sets, Information and Control, Vol. 8, **1965**, pp. 338-353