International Journal of Information Technology and Knowledge Management
July-December 2010, Volume 2, No. 2, pp. 361-364

INNOVATIVE SCENARIO IN SOFTWARE DEVELOPMENT EFFORT ESTIMATION BASED ON A NEW FUZZY LOGIC MODEL

Rajender Bathla[1], Sukhdev Singh[2], Mittar Vishav[3] & Rahul Gupta[4]

Software developing has always been characterized by some metrics. In this area, one of the greatest challenges for software developers is 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. Many of the problems of the existing effort estimation models can be solved by incorporating Fuzzy Logic. This paper described an enhanced Fuzzy Logic model for the estimation of software development effort and proposed a new approach by applying Fuzzy Logic for software effort estimates. Results show 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.

Keywords: Software Engineering, Software Metric, Software Effort Estimation, Fuzzy Logic, COCOMO Model

1. INTRODUCTION

1.1. Software Development Effort Estimation

Software metric and especially software estimation is base on measuring of software attributes which are typically related to the product, the process and the resources of software development [1]. This kind of measuring can be used as parameters in project management models [2] 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 researched areas of software measurement is software effort estimation. Software effort estimation models divided into two main categories: algorithmic and non-algorithmic.

The most popular algorithmic estimation models include Boehm!s COCOMO [3], Putnam!s SLIM [4] and Albrecht!s Function Point [5]. 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 models also have difficulty in modeling the inherent complex relationships between the contributing factors, are unable to handle categorical data as well as lack of reasoning capabilities [6]. The limitations of algorithmic models led to the exploration of the non-algorithmic techniques which are soft computing based.

Deptt. of Computer Science & Engg, Hiet, Kaithal, Haryana, India
Email: sukhdev_kuk@rediffmail.com, mittarvishav@gmail.com, ron_btech@yahoo.co.in

These include artificial neural network, evolutionary computation, fuzzy logic models, case-based reasoning, combinational models and so on. Artificial neural network are used in effort estimation due to its ability to learn from previous data [7][8]. It is also able to model complex relationships between the dependent (effort) and independent variables (cost drivers) [7][8]. In addition, it has the ability to generalise 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 Perceptron, Backpropagation algorithm and sigmoid function [7]. Selecting good models for software estimation is very critical for software engineering. In the recent years many software estimation models have been developed [4, 5, 6, 7 , 8, 9].

Gray and MacDonell compared 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 [10] 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 [11] 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. Idri

[7, 8] further defined 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 [13] 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.

2.2. Fuzzy Logic Approach

Since fuzzy logic foundation by Lotfi Zadeh in 1965, it has been the subject of important investigations [12]. It is a mathematical tool for dealing with uncertainty and also it provides A technique to deal with imprecision and information granularity [11]. The fuzzy logic model uses the fuzzy logic concepts introduced by Lotfi Zadeh [12]. Fuzzy reasoning consists of three main components [11, 12, 13, 14]: fuzzification process, inference from fuzzy rules and defuzzification process. Fuzzification process is where 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. Inferencing 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.

2.3. Parameters ANALYSIS

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

$$MRE_i = \frac{\left|\text{Actual Effort}_i - \text{Predicted Effort}_i\right|}{\text{Actual Effort}_i} \quad (1)$$

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

$$MMRE = \frac{1}{N}\sum_{i}^{N} MRE_i \quad (2)$$

2. METHODOLOGY

The empirical study carried out here is based on the empirical study done by Lopez-Martin et al [10, 11]. 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 [5, 9].

3. RESULTS

A subset of projects, 10 projects, is selected and statistics and a brief description related to each module are depicted in Table 1 which is prepared by Lopez-Martin et al. [13, 14]. Table 1) Modules description and metrics, MC: McCabe Complexity, DC: Dhama Coupling, LOC: Lines of Code,

Table 1
DT: Development Time (Minutes)

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	minimum and	maximum values from a	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	0.167	8	18

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 a " b " c. Their scalar parameters (a, b, c) are defined as follows:

MF(x) = 0 if x < a

MF(x) = 1 if x = b

MF(x) = 0 if x > c

Six rules are suggested [6]:

1. If Complexity is low and Size(LOC) is small then DT is low;
2. If Complexity is average and Size(LOC) is medium then DT is average;
3. If Complexity is high and Size(LOC) is big then DT is high;
4. If Coupling is low then DT is low;
5. If Coupling is average then DT is average;
6. If Coupling is high then DT is high.

The membership function plots corresponding to Table 1 are shown in Figures 1, 2, 3 and 4

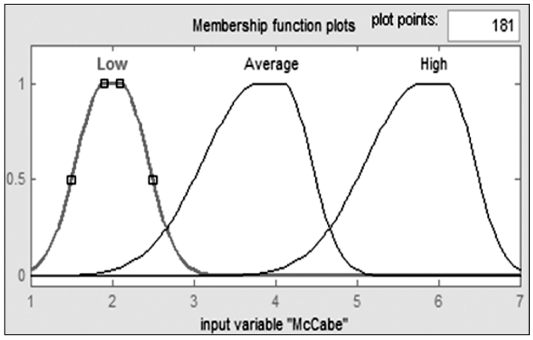

Fig. 1: McCabe Complexity Plot (Input)

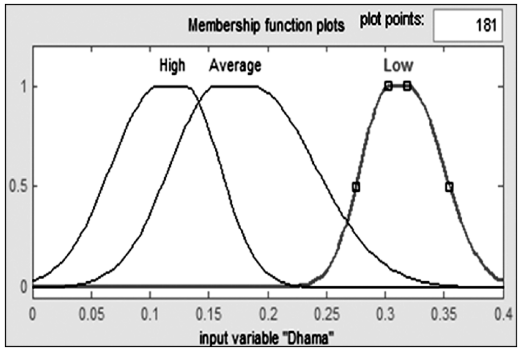

Fig. 2: Dhama Coupling Plot (Input)

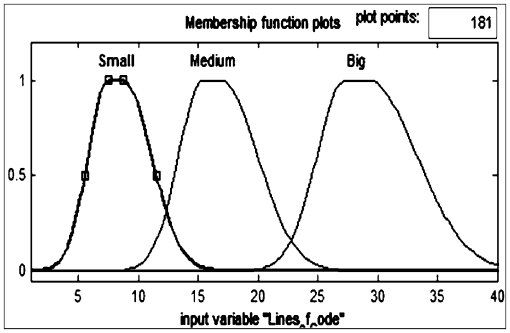

Fig. 3: Physical Lines of Code Plot (Input)

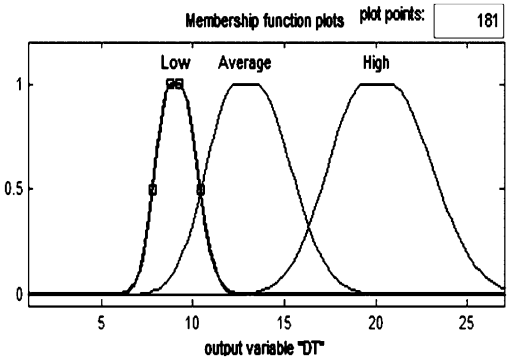

Fig. 4: Development Time Plot (Output)

The calculation of MRE and MMRE are shown in Table 2.

Table 2
Calculation of MRE, MMRE

Table 2 is not found

The result 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.

4. CONCLUSION

The paper suggests a new approach for estimating of software projects development time. 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. Here, the advantages of fuzzy logic and good generalization are obtained. The main benefit of this model is its good interpretability by using the fuzzy rules and another great advantage of this research is that it can 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] N. E. Fenton, S. L. Pfleeger, "Software Metrics, A Rigorous and Practical Approach", 2nd Edition, PWS Publishing Company, Thomson Publishing, Boston, 1997.

[2] A. R. Gray, S. G. MacDonell, "Applications of Fuzzy Logic to Software Metric Models for Development Effort Estimation". Fuzzy Information Processing Society 1997 NAFIPS! 97, Annual Meeting of the North American, 21 & 24 September 1997, pp. 394 & 399.

[3] B. W. Boehm, Software Engineering Economics, Englewood Cliffs, NJ, Prentice Hall, 1981.

[4] L. H. Putnam, "A General Empirical Solution to the Macrosoftware Sizing and Estimating Problem". IEEE Transactions on Software Engineering, SE-4(4), 1978, pp 345-361.

[5] Attar Software. 2002. "Fuzzy Logic in Knowledge Builder", White Paper. http://www .i n t ellicraf t ers.com / fuzzy.htm.

[6] M. O. Saliu, M. Ahmed and J. AlGhamdi. "Owards Adaptive Soft Computing based Software Effort Prediction", Fuzzy Information, 2004. Processing NAFIPS '04. IEEE Annual Meeting of the North American Fuzzy Information Processing Society, 27-30, June 2004, 1, pp. 16-21.

[7] A. Idri, T. M. Khoshgoftaar, A. Abran. "Can Neural Networks be Easily Interpreted in Software Cost Estimation", IEEE Trans. Software Engineering, 2, 2002, pp. 1162 & 1167.

[8] A. Idri,, A. Abran,, T.M. Khoshgoftaar. #Estimating Software Project Effort by Analogy based on Linguistic Values" in.Proceedings of the Eighth IEEE Symposium on Software Metrics, 4-7 June 2002, pp. 21 & 30.

[9] C. J. Burgess, M. Lefley. "Can Genetic Programming Improve Software Effort Estimation, A Comparative Evaluation", Information and Software Technology, 43, No.14, 2001, pp. 863-873.

[10] Fei Z, Liu X (1992) f-COCOMO: :Fuzzy Constructive Cost Model in Software Engineering". In: IEEE International Conference on Fuzzy Systems, pp 331&337.

[11] Roger JS (1993) ANFIS: "Adaptive Network based Fuzzy Inference Systems". IEEE Trans Syst Man Cybern, 3(3):665&685.

[12] L. A. Zadeh. "Fuzzy Sets". Information and Control, 8, 1965, pp. 338-353.

[13] Xu Z. and Khoshgoftaar, T. M. Identification of Fuzzy Models of Software Cost Estimation, 2004.

[14] S. G. MacDonell, A. R. Gray, J. M. Calvert. #FULSOME: Fuzzy Logic for Software Metric Practitioners and Researchers$ in 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.