International Journal of Advanced Research and Development
ISSN: 2455-4030
Impact Factor: RJIF 5.24
www.advancedjournal.com
Volume 3; Issue 1; January 2018; Page No. 125-127

A fuzzy logic approach to software development effort estimation

Mandeep Kaur
Department of Computer Science, Khalsa College for Women, Civil Lines, Ludhiana, Punjab, India

Abstract
Accurate effort estimation is an ongoing challenge and is the most important activity in software project management. The data available in the initial stages of software development is not clear and incomplete which makes the software effort estimation a challenging task. Many models do exist which help in estimating the effort but each has its own pros and cons. This paper presents a comprehensive survey on the use of fuzzy logic in the field of software development effort estimation as fuzzy logic is effective in dealing with uncertainty in data.

Keywords: fuzzy logic, effort estimation, COCOMO, membership functions, mean magnitude of relative error

1. Introduction

Software is an end product obtained after the completion of a set of activities ranging from requirement gathering, analysis, design, coding and testing and the continuous process of maintenance. For the effective management of the software project, various estimates need to be made accurately and efficiently in the very early stages of software project development. These estimates include:

1. Software Size Estimation which can be measured in terms of Lines of Code (LOC) and Function Points [1],
2. Software Effort Estimation which is measured in terms of man-hour required to complete the project [2],
3. Software Time Estimation which is made on basis of software size and effort,
4. Software Cost Estimation which is based on the size, hardware, software, personnel, etc. requirement of the software.

Software effort estimates helps in estimating the schedule and staffing requirements which in turn collectively form the basis for project bidding, budgeting, risk analysis, project planning, progress monitoring and control. Over past many decades different models have been proposed and used for estimation of effort but each of them has their own pros and cons. The models can be broadly classified into two categories:

1. Algorithmic Models are based on some mathematical equation. For example: COCOMO [3], Albrecht's Function Points, Putnam's SLIM [4].
2. Non-Algorithmic Models include Expert Judgment, Delphi, Fuzzy Logic, Neural networks.

Due to some limitations in algorithmic models [5], non-algorithmic models for effort estimation were explored out of which fuzzy logic approach was able to deal with the inaccurate, vague, imprecise data for effort estimation.

2. Fuzzy Logic Approach

The concept of fuzzy logic was introduced by Prof. Lofti Zadeh in 1965 [6, 7]. It is a form of multi-valued logic that takes the values between 0 and 1. It easily deals with uncertain, inaccurate and vague data, hence is one of the reasons for use in software effort evaluation [8]. It imitates the human decision making process. It is used in the field of aerospace, defense, automotive, finance, business, electronics, manufacturing, marine, pattern recognition, classification, medical, securities, etc.

Fuzzy logic system consists of four main components:

1. **Fuzzification**: It transforms the crisp input values into fuzzy values using membership functions where a membership function is defined as $\mu_A(x) \rightarrow [0,1]$ for a fuzzy set 'A' on the universe of discourse 'x'. Some of the membership functions are:

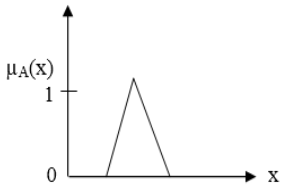

Fig 1: Triangular Function

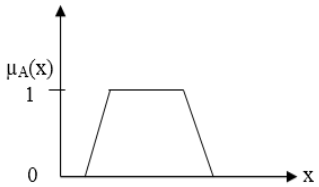

Fig 2: Trapezoidal Function

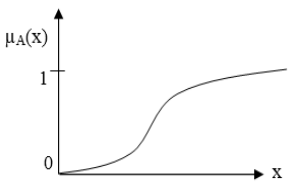

Fig 3: Logistic Function

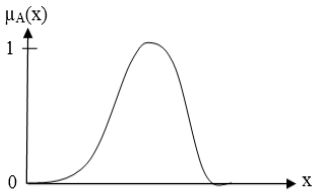

Fig 4: Guassian Function

2. **Knowledge Base**: It consists of fuzzy sets and rule base that maps the input set into fuzzy output set. Fuzzy rule base contains the if-else rules.
3. **Inference Engine**: It uses the rules in the knowledge base to derive the output.
4. **Defuzzification:** It is the process of conversion of fuzzy outputs to crisp outputs.

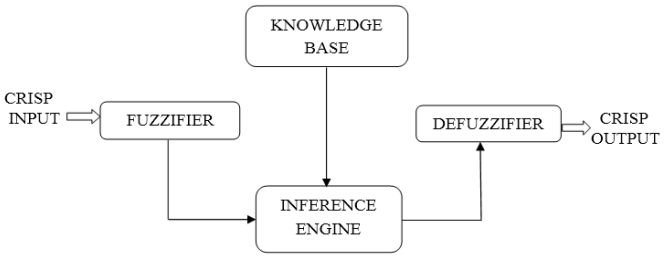

Fig 5: Fuzzy Logic System

In estimation of software development effort, size, cost drivers, scale factors, effort multipliers, etc are given as an input to fuzzy system which are then fuzzified and passed to the inference engine which uses the rules specified in the knowledge base to perform the required conversion of fuzzy inputs to fuzzy outputs. Finally the defuzzifier converts the fuzzy output to crisp output which is used for estimating the effort. The next section presents a review of literature on the use of fuzzy logic in estimation of software development effort.

3. Related Work

Sandhu *et al*. [9] proposed the use of Neuro-Fuzzy technique for software effort estimation on NASA software project data. The performance of the proposed model was compared with the Halstead, Walston-Felix, Bailey-Basili and Doty Models. The results obtained showed that the Neuro-fuzzy system had the lowest MMRE and RMSSE values as compared to other effort estimation models.

Attarzadeh and Ow [10] proposed an enhanced Fuzzy Logic model for the software development effort estimation. The value of MMRE after applying enhanced Fuzzy Logic was observed to be lower than MMRE obtained by the application of other Fuzzy Logic models. Hence enhanced fuzzy logic was considered to be much better than other fuzzy logic model.

Kad and Chopra [11] applied fuzzy logic to different parameters of Constructive Cost Model (COCOMO) II. The results obtained by application of fuzzy logic showed that the value of MMRE and pred was much better than MMRE of algorithmic model. The validation of results was carried out on COCOMO dataset.

Malathi and Sridhar [12] proposed reasoning by analogy and fuzzy approach to estimate software effort for projects represented by categorical or numerical data. The results obtained by proposed model were more accurate in comparison to the dataset analyzed with the earlier methodologies.

Reddy *et al*. [13] proposed the use of Fuzzy Triangular Membership Function and GBell Membership Function to estimate software effort. NASA93 dataset was used and the results obtained were compared with the Intermediate COCOMO on the basis of different criterions like VAF, MARE, VARE, MMRE, Prediction and Mean BRE. It was observed that the Fuzzy Logic Model using Triangular Membership Function provided better results as compared to GBell membership function or Intermediate COCOMO.

Du *et al*. [14] proposed a combination of neuro-fuzzy technique with the System Evaluation and Estimation of Resource Software Estimation Model (SEER-SEM). The performance of the proposed model was accessed by designing and conducting evaluation with published projects and industrial data. The estimation obtained using proposed neuro-fuzzy model containing SEER-SEM was found to be better in comparison to the estimation results obtained using only SEER-SEM algorithm.

Kumar *et al*. [15] proposed a Fuzzy logic based model for software effort prediction. This model was developed and validated upon student data. The model had good interpretability by using fuzzy rules and it put together expert knowledge (Fuzzy rules) and project data into one general framework. When the comparison of multiple regression model was done with fuzzy logic model, the results supported the fuzzy logic model.

Batra and Trivedi [16] proposed the use of Gaussian Membership Function (GMF) for the COCOMO cost drivers. It was found that the Gaussian function performed better than the trapezoidal function, and the results achieved were closer to the actual effort.

Kumar [17] proposed two new models for effort estimation based on LOC and functional point analyses (FPA) using fuzzy logic techniques. Size was considered as a fuzzy number in LOC based model and the research was conducted on NASA software projects dataset. MARE obtained for proposed model was much lower than that of existing models. Fuzzy numbers were used to represent DET's, to get variation of function points. It was observed that the proposed model gave better results as compared to some earlier models.

Chawla and Ahlawat [18] compared the different membership functions of fuzzy logic in their ability to accurately measure the software development effort. It was observed that Trapezoidal Membership function gave more accurate estimate of Effort as compared to Triangular Membership Function. GBell membership function gave better result as compared to Gauss2 membership function but Gaussian Curve MF's were more accurate than the simple straight line membership functions. It was observed that out of five techniques used in the research GBell MF had the lowest value of MMRE. So, it had the highest accuracy.

4. Conclusion

Software effort estimation is one of the major issues that need to be addressed at the early stages of software development life cycle. Since in the initial stages of software development, the data required for effort estimation is incomplete, vague and imprecise, the estimates based on this type of data is not accurate which results in losses, improper schedules and staffing, etc. It is observed that fuzzy logic is one technique that helps to deal with this vagueness effectively. Many researchers have deployed fuzzy logic on COCOMO, COCOMO II, function points to estimate the effort and the results obtained were better as compared to other techniques. The literature review presented helps to conclude that fuzzy logic is widely used to efficiently and accurately estimate the software development effort.

5. References

1. Pressman RS. Software Engineering: A Practitioner's Approach, McGraw-Hill series III Computer Science,

New York, 2001.
2. Mall R. Fundamentals of Software Engineering, PHI Learning Private Limited, New Delhi, 2008.
3. Boehm BW. Software Engineering Economics, Engle woods Cliffs, NJ, Prentice-Hall, 1981.
4. Putnam LH. A general empirical solution to the macro software sizing and estimating problem, IEEE transactions on Software Engineering, 2:345-361.
5. Kocaguneli E, Menzies T, Bener A, Keung JW. Exploiting the Essential Assumptions of Analogy-based Effort Estimation, Journal of IEEE Transactions on Software Engineering. 2010; 34(4):471-484.
6. Zadeh LA. Fuzzy Sets, Information and Control. 1965; 8:338-353.
7. Zadeh LA. fuzzy Logic, Neural Networks and Soft Computing, Communication of ACM. 1994; 37(3):77-84.
8. Rojas R. Neural Network, Springer-Verlag, 1996.
9. Sandhu PS, Bassi P, Brar AS. Software Effort Estimation Using Soft Computing Techniques, World Academy of Science, Engineering and Technology, 2008, 46.
10. Attarzadeh I, Ow SH. Software Development Effort Estimation Based on a New Fuzzy Logic Model, International Journal of Computer Theory and Engineering, 2009; 1(4):473-476.
11. Kad S, Chopra V. Fuzzy Logic based framework for Software Development Effort Estimation, Research Cell: An International Journal of Engineering Sciences. 2011; 1:330-342.
12. Malathi S, Sridhar S. A Classical Fuzzy Approach for Software Effort Estimation on Machine Learning Technique International Journal of Computer Science 2011; 8(6):249-25.
13. Reddy PVGD, Sudha KR, Rama SP. Application of Fuzzy Logic Approach to Software Effort Estimation, International Journal of Advanced Computer Science and Applications. 2011; 2(5):87-92.
14. Du WL, Ho D, Capretz LF. Improving Software Effort Estimation Using Neuro-Fuzzy Model with SEER-SEM, Global Journal of Computer Science and Technology. 2010; 10(12):52-64.
15. Kumar S, Pal J, Bhattacherjee V. Software Effort Prediction - A Fuzzy Logic Approach, Global Journal of Computer Science and Technology Software & Data Engineering, 2013, 13(10).
16. Batra G, Trivedi M. A Fuzzy Approach for Software Effort Estimation, International Journal on Cybernetics & Informatics. 2013; 2(1):9-15.
17. Kumar J. A Fuzzy Model for Function Point Analysis for Software Effort Estimation, International Journal of Advanced Research in Computer Science and Software Engineering. 2014; 4(4):1372-1378.
18. Chawla R, Ahlawat D. Software Development Effort Estimation using Fuzzy Logic Framework- An Implementation, International Journal On Advanced Computer Theory And Engineering. 2015; 4(4):15-21.