Volume 5, Issue 8, August 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Putnam's SLIM Model for Cost Estimation Effort using Fuzzy Logic Membership Functions

Swati Singh, Shubha Jain
Department of CSE, Kanpur Institute of Technology,
UPTU, U.P., India

Abstract*— *Software cost estimation is one of the most important software development activities; however it is also one of the most difficult activities. As software applications have grown in size and importance, the need for accuracy in software cost estimation has also grown. The major attribute that impacts the cost of software development is human effort. This research paper presented the outputs of various cost estimation models .On the basis of the results obtained the performance is compared on an evaluation techniques MMRE-Mean Magnitude Relative Error. The simulation has been done using the fuzzy logic toolbox MATLAB on 12 projects of NASA 93 Data Set which have their lines of code greater than 70 KLOC. Putnam's SLIM model has been devised using fuzzy inference system and results have been well presented and discussed here.

Keywords*— *Putnam's SLIM Model, COCOMO, Fuzzy Logic, Membership Functions, NASA 93 Data Set, MMRE.

I. INTRODUCTION

Software cost estimation has been an important but difficult task since the beginning of the computer era in the 1940s. As software applications have grown in size and importance, the need for accuracy in software cost estimation has also grown. The major attribute that impacts the cost of software development is human effort. The most cost estimation methods focus on this aspect and give estimates in terms of person -months. Software cost estimation predicts the likely amount of effort, time, and staffing levels required to build a software system at an early stage of project development. However, estimates at the preliminary stages of the project development are difficult to obtain because the primary source to estimate the cost comes from the requirement specification document [1].

1.1 Different Methods of Estimation

There exist various methods to perform software cost estimation, each having its own strengths and weaknesses. The methods are as follows:

1.1.1 Algorithmic Methods

The algorithmic methods are designed to provide some mathematical equations to perform software estimation. These mathematical equations are based on research and historical data and use inputs such as Source Lines of Code (SLOC), number of functions to perform, and other cost drivers such as language, design methodology, skill-levels, risk assessments, etc [5]. The various algorithmic methods are COCOMO model, Putnam model, function points based models etc.

1.1.1.1 COCOMO Model

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Boehm [6, 7]. The COCOMO model is a set of three models: Basic, Intermediate, and Detailed. In COCOMO the code size S is given in thousand LOC (KLOC) and Effort is in person-month.

i. Basic COCOMO

The basic COCOMO model computes software development effort (and cost) as a function of program size expressed in estimated lines of code (LOC).The basic COCOMO is given as in equation 1.1 :

$$\text{Effort Applied (E)} = A_a(\text{KLOC})B_b \quad (1.1)$$

where,

KLOC is the estimated number of delivered lines (expressed in thousands) of code for project. The A_a and B_b are the coefficients constant which given in the following table1.1:

Table 1.1: COCOMO mode coefficient values (Basic Model)

MODE	A_a	B_b
Organic	**2.4**	**1.05**
Semi-detached	**3.0**	**1.12**
Embedded	**3.6**	**1.2**

ii. Intermediate COCOMO

Idri and Abran [13] reported that intermediate COCOMO model is the most widely used version because the accuracy of an estimation obtained with the intermediate version is greater than that of simple COCOMO 81 model, and very similar to that obtained with detailed version. This model computes effort as a function of program size and a lot of cost drivers that includes subjective assessment of product attributes, hardware attributes and project attributes. These 15 attributes get a 6 point scale ranging from "very low" to "extra high". Based on the rating, effort multiplier is determined .The product of all effort multipliers result in "Effort Adjustment Factor" (EAF).Intermediate COCOMO model can be the following for:

$$E = a_i \, (KLOC)^{b_i} \times EAF \qquad (1.2)$$

Where,

E: Effort applied in terms of "person –months"

KLOC: Kilo lines of code for the project

EAF: It is the effort adjustment factor

The value of a_i and b_i for various software projects are shown in table 1.2:

Table 1.2: COCOMO mode coefficient values (Intermediate Model)

Mode	a_i	b_i
Organic	**3.2**	**1.05**
Semi-detached	**3.0**	**1.12**
Embedded	**2.8**	**1.2**

iii. Detailed COCOMO

Detailed COCOMO model incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc) of the software engineering process. While both basic and intermediate COCOMOs estimate the software cost at the system level, detailed COCOMO works on each sub-system separately and has an obvious advantage for large systems that contain non-homogeneous subsystems.

1.2.1.2 COCOMO II (Constructive Cost Model)

One of the problems with using COCOMO 81 today is that it does not match the development environment of the late 1990's and 2000's. It was created in a time when batch jobs were the norm; programs used to run on mainframes and compile times were measured in hours instead of seconds [8]. In 1997 COCOMO II was published and was supposed to solve most of these problems.

COCOMO II describes 17 cost drivers that are used in the Post-Architecture model. They are used in the same way as in COCOMO 81 to calculate the EAF. The cost drivers for COCOMO II are again rated on a scale from Very Low to Extra High in the same way as in COCOMO 81. COCOMO II post architecture model is given in following equation 1.3:

$$Effort = A \times (size^B) \times \prod_{i=1}^{17} Effort\ multiplier_i \qquad (1.3)$$

- Where, $B = 1.01 + 0.01 \times \sum_{i=1}^{5} Scale\ Factor_i$
- A: Multiplicative Constant
- Size: Size of the software measures in terms of KSLOC (thousands of Source Lines of Code)
- The scale factors (SF) are based on a significant source of exponential variation on a project's effort or productivity variation.

1.2.1.3 Putnam's Model and SLIM

Putnam derives his model based on Norden/Rayleigh manpower distribution and his finding in analysing many completed projects [11]. The central part of Putnam's model is called software equation as follows:

$$S = E \times (Effort)^{\frac{1}{3}} t_d^{\frac{4}{3}} \qquad (1.4)$$

Where, t_d is the software delivery time; E is the environment factor that reflects the development capability, which can be derived from historical data using the software equation. The size S is in LOC and the Effort is in person-year. Another important relation found by Putnam is-

$$Effort = D_0 \times t_d^{\,3} \qquad (1.5)$$

Where, D_0 is a parameter called manpower build-up which ranges from 8 (entirely new software with many interfaces) to 27 (rebuilt software). Combining the above equation with the software equation, we obtain the power function form:

$$Effort = \left(D_0^{\,4/7} \times E^{-9/7}\right) \times S^{9/7} \qquad (1.6)$$

and

$$T_d = \left(D_0^{\,-1/7} \times E^{3/7}\right) \times S^{3/7} \qquad (1.7)$$

Putnam's model is also widely used in practice and SLIM is a software tool based on this model for cost estimation and manpower scheduling.

II. SOFT COMPUTING TECHNIQUES

Soft Computing became a formal Computer Science area of study in early 1990s [12].Soft computing methodology is a group of methodologies centring in Fuzzy Logic (FL), Artificial Neural Networks (ANN) and Evolutionary Computation

(EC). Soft computing achieves robustness and provides low-cost solutions by exploiting tolerance for imprecision, uncertainty, approximate reasoning, and partial truth.

2.1 Fuzzy Logic

The theory of Fuzzy Logic was first raised by the mathematician Lotfi A. Zadeh [11] in 1965. This theory is a result of the insufficiency of Boolean algebra to many problems of the real world. As most of the information in the real world is imprecise , and one of humans' greatest abilities is to effectively process imprecise and "'fuzzy" information. Fuzzy systems are knowledge based or rule based systems. A fuzzy system is based on IF-THEN rule is an IF-THEN statement in which some words are characterized by continuous membership functions.

2.2 Membership Functions

The membership function $\mu_A(x)$ describes the membership of the elements x of the base set in the fuzzy set A, whereby for $\mu_A(x)$ a large class of functions can be taken. There are different forms of membership functions such as triangular, trapezoidal, piecewise linear, Gaussian, or singleton. The most common types of membership functions are triangular, trapezoidal, and Gauss shapes .The presented paper has implemented triangular and Gauss membership function which are discussed as follows in figure 2.1 and figure 2.2:

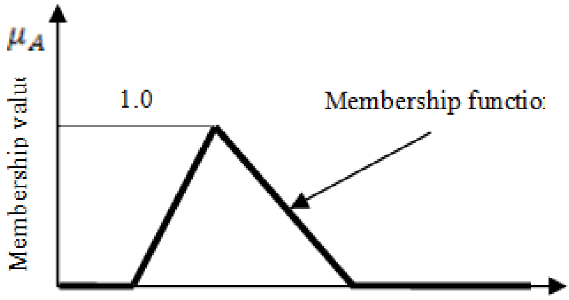

Figure 2.1: Membership Grades of a Fuzzy Set

A) Triangular MFs

A triangular MF is defined by a lower limit a, an upper limit c, and value b, where $a < b < c$ as follows:

$$f(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \dfrac{x-a}{b-a}, & a \leq x \leq b \\ \dfrac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases}$$

The parameters (a, b, c) determine the x coordinates of the three corners of the underlying triangular MF. Figure 3.3 illustrates a triangular MF defined by $f(x; 20,60,80)$.

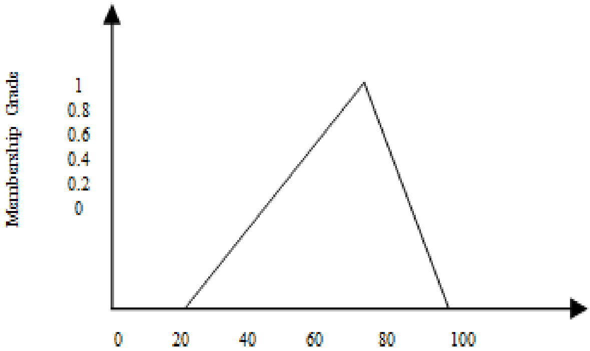

Figure 2.2: Triangular MF

B) Gauss MFs

A Gaussian membership function used to represent vague, linguistic terms and it is specified by two parameters:

$$f(x; c, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}$$

A Gauss MF is determined completely by c and σ; c represents the membership functions centre and σ (standard deviation) determines the membership functions width in fuzzy set A. It is quite popular in the fuzzy logic literature. Figure 3.4 plots a Gauss membership functions defined by $f(x; 50,20)$.

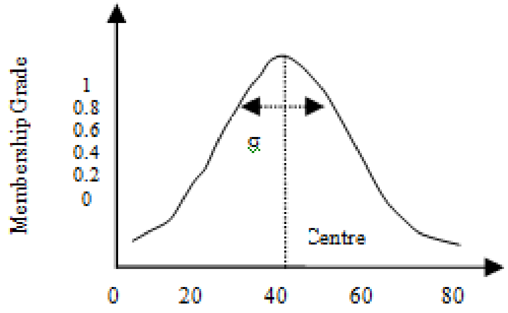

Figure 2.3: Gauss MFs

2.3 Fuzzy Inference System

The FIS formulates suitable rules and based upon the rules the decision is made. This is mainly based on the concepts of the fuzzy set theory, fuzzy IF–THEN rules, and fuzzy reasoning. FIS uses "IF. . . THEN" statements, and the connectors present in the rule statement are "OR" or "AND" to make the necessary decision rules. The basic FIS can take either fuzzy inputs or crisp inputs, but the outputs it produces are almost always fuzzy sets. Schematic view of FIS is shown in figure 2.4.

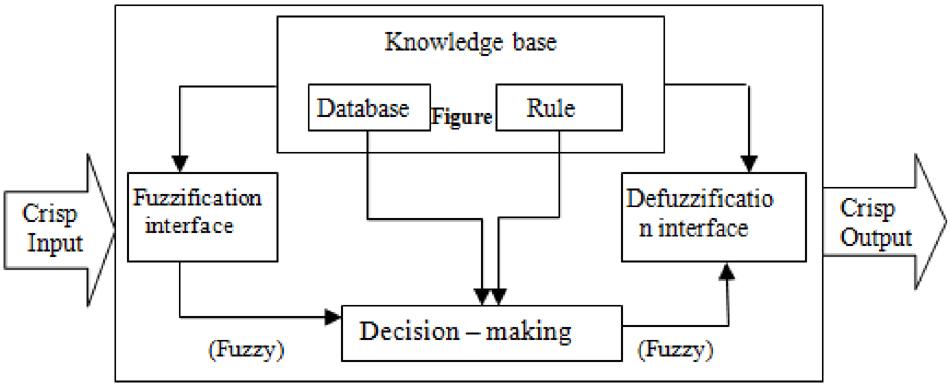

Figure 2.4: Fuzzy Inference System

III. PROPOSED MODEL USING FUZZY INFERENCE SYSTEM

A software cost estimation model based on fuzzy logic has been proposed for handling uncertainties in inputs for Putnam's SLIM model in the process of software development.

3.1 Software Cost Estimation using Fuzzy Logic

Inaccuracy is present in all parameters of the Putnam's SLIM model. It is important to stress that uncertainty at the input level of the Putnam's SLIM model yields uncertainty at the output. Figure 3.1 represents the proposed model.

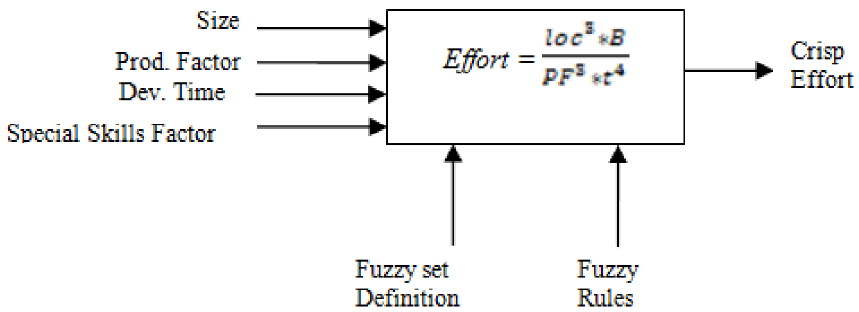

Figure 3.1: The Proposed Model: Inputs (Size, Prod. Factor, Dev. Time and Special Skill Factor) and Output: (Effort Estimation)

The proposed framework integrates the four components of the Putnam's SLIM model as given in Eq. (3.1): the size, productivity factor, development time and the special skills factor.

3.1.1 Fuzzy Putnam's SLIM model for Effort

The model has been derived from productivity data collected for over 4000 contemporary software projects. Based on these data, an estimation model of the form

$$Effort = \frac{loc^3 * B}{PF^3 * t^4} \quad (3.1)$$

Where E =Effort in person years

loc =Size
t = Development Time
B= Special skills factor (.39 for >= 70 KLOC)
p = Productivity parameter that reflects:
P= 2000 for real time embedded software
P=10000 for Tele. Communication and systems software
P=28000 for business systems applications

The following figure 3.5 and figure 3.6 shows the Putnam's SLIM Model and subsystem model designed in MATLAB Fuzzy Inference System.

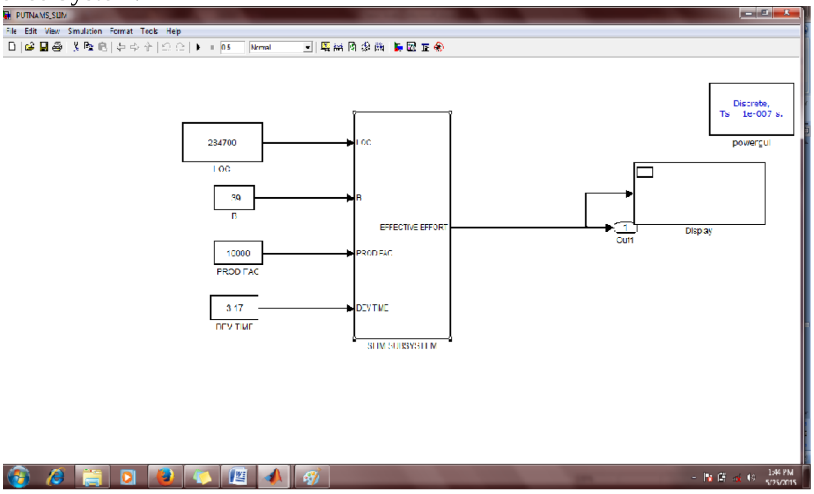

Figure 3.2 Putnam's SLIM Model using MATLAB Fuzzy Inference System

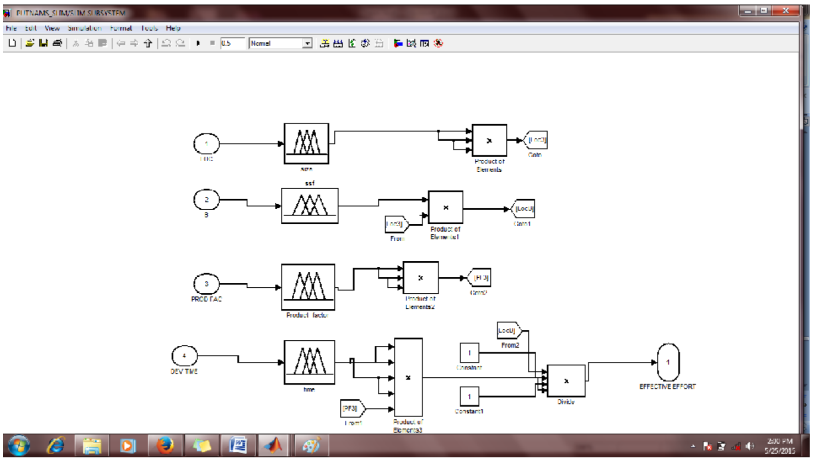

Figure 3.3: Putnam's SLIM SUBSYSTEM Model using MATLAB Fuzzy Logic

IV. RESULTS

The proposed model has been executing over 12 projects of NASA 93 data set. The given graph in figure 4.1 shows the estimated effort by using triangular membership function which is close to the actual effort as compared to COCOMO II and Putnam's SLIM. There are some cases where estimated effort differs from the actual. There are many reasons for this; one of them is outliers' data. Second Putnam's SLIM dependent on technology factor. It can be decided by analysing the data from previously completed projects which is then used to calibrate the model.

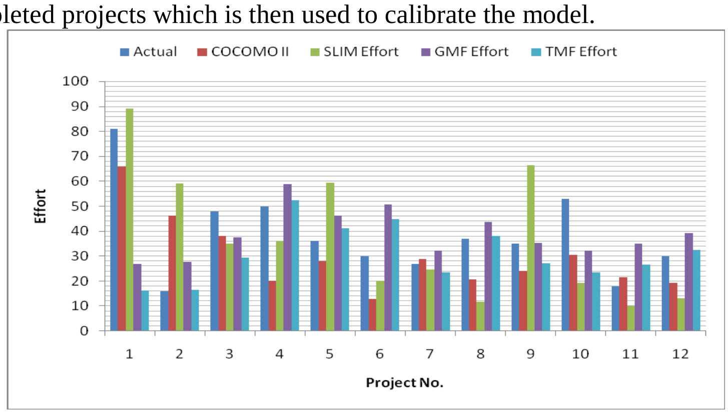

Figure 4.1: Estimated effort using various models versus actual effort

Depending upon the estimated effort in the graph Mean Magnitude Relative Error-MMRE and Prediction for each model has been calculated. The Magnitude of Relative Error- MRE value has been calculated for each observation say i through the equation (4.1).

$$MRE_i = \frac{Actual\ Effort_i - Estimated\ Effort_i}{Actual\ Effort_i} \times 100 \quad (4.1)$$

which is the effort estimated at that observation. The aggregation of magnitude of relative error over multiple observations (n) can be achieved through the equation (4.2) .

$$MMRE = \frac{1}{n} \sum_i^n MRE_i \quad (4.2)$$

The following graph in figure 4.1 shows the comparison of Mean Magnitude Relative error MMRE of described models for cost estimation and it has been found that using Fuzzy membership functions better results are observed.

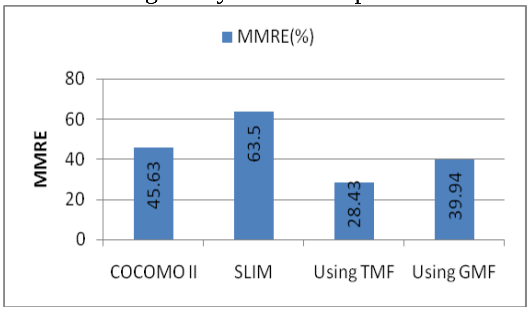

Figure 4.1: Comparison of MMRE of various models

V. CONCLUSION AND FUTURE SCOPE

On executing the Fuzzy membership functions on the proposed model discussed above, it has been observed that the MMRE for TMF and GMF has decreased values in comparison to other models. The uncertainty in outputs has been improved and yielded crisp values for 12 projects taken from NASA 93 Data Set. However, it may be added that these results are based upon the observations over 12 projects only and the results may vary when explore more projects. This proposed model can be extended when implemented using Fuzzy Neural Networks, Membership functions except gauss and triangular functions. In addition to this more real life projects can be calculated and compared for further enhancements and better precision in MMRE. This will further ensure the validity of the present work done over the 12 projects of NASA 93 Data Set.

REFERENCES

[1] Kirsopp, C. and Shepperd, M.J., *"Making Inferences with Small Numbers of Training Sets"*, 6th International Conference on Empirical Assessment & Evaluation in Software Engineering,Vol. 58, pp 123-130, 2002.

[2] Boehm (1981): *Software Engineering Economics*, Prentice Hall, 1981.

[3] G.N. Parkinson, *"Parkinson's Law and Other Studies in Administration",* Houghton Miffin, pp 151-163, 1991.

[4] Royce, W.*:"Software Project Management: A Unified Framework"*, Addison Wesley, 1998.

[5] J. Ryder, *"Fuzzy modeling of software effort prediction",* Proceedings of IEEE Information Technology Conference, 1998.

[6] R.K.D. Black, R. P. Curnow, R. Katz and M. D. Gray, *BCS Software Production Data, Final Technical Report,* RADC-TR-77-116, pp 2-2, 1977.

[7] B. W. Boehm, *Software engineering economics,* Englewood Cliffs, NJ: Prentice-Hall, 1981.

[8] Center for Software Engineering, *http://sunset.usc.edu/research/COCOMOII/cocomo _main .html.*

[9] Boehm, B., Clark, B., Horowitz, E., Madachy, R., Shelby, R., and Westland C.: *"Cost Models for Future Software Life Cycle Processes: COCOMO 2.0"*, Annals of Software Engineering, 1995.

[10] Adam A., Jaralla, A., and Ahmed, M.: *"Can Cohesion Predict Fault Density?"* The 4th ACS/IEEE International Conference on Computer Systems and Applications, March 8-11, 2006.

[11] Lotfi A. Zadeh: *Fuzzy Sets Information and Control 8(3):* pp 338-353, 1965.

[12] Lotfi A. Zadeh: *Fuzzy Logic, Neural Networks, and Soft Computing.* Commun. ACM 37(3): pp 77-84,1994.

[13] Allidri and Alain Arban, Laila Kjiri *"COCOMO Cost Model Using Fuzzy Logic"*7th International Conference On Fuzzy Theory & Technology, 2000.