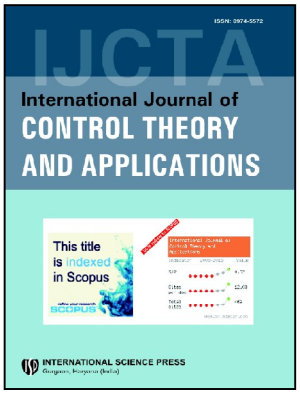

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 10 • 2017

Eliminating Analogy Cost Estimation Errors Using Fuzzy Logic

D. Manikavelan[1] and R. Ponnusamy[2]

[1]Dept of CSE, Research scholar at Sathyabama University
[1]Working as an Associate Professor in Dhanalakshmi [1]college of Engineering, Chennai, India, E-mail: manickavelan@yahoo.com
[2] Professor, Sri Lakshmi Ammal Engineering College, Chennai, India, E-mail: r_ponnusamy@hotmail.com

Abstract: It is quite usual in industries that estimations are done based on the complexity of requirements, number of people, etc. Not all such estimations made turn out to be accurate because there is always a deviation between actual and estimated values. The actual and targeted expenditure at the end each major milestone differs to a great extent. In order to overcome such erroneous estimates, we use analogy method since it is fast and reliable. Analogy is an algorithmic model to create accurate estimates for any proposed project by comparing the proposed project against similar projects from the past. In this paper, we reduce random errors in analogy estimation using fuzzy logic technique which makes it convenient to extract projects with minimal random error from a set of inputs. Random errors occur when the requirements for a project continuously change during development. Triangular function helps in mapping fuzzy values to real numbers and thereby eliminates random errors in projects.

Keywords: Fuzzy Logic Algorithm, Computational Intelligence, Analogy-based estimation.Fuzzy membership function

I. INTRODUCTION

Software cost estimation method is comparing the features of the method based on clustering abilities it is also useful for selecting the best of the each project. There are two approaches to deal with software cost estimation, one is algorithmic approach and another one is non- algorithmic approach. Non algorithmic approaches are constructive cost model (COCOMO) and Function Point model. COCOMO is most widely used in software cost estimation [2]. COCOMO based estimations use cost, effort and schedule for estimating the project. It is derived from statistical regression of data from past projects[3]. Functional Point Analysis is one of the best methods for measuring functional size of software[4]. Functional Point metrics is a standardized method for measuring the functions of a software application. Algorithmic cost modeling is a mathematical function of product, project and process attributes. The most commonly used product attribute for cost estimation is code size. This model is developed using historical cost information that relates some software metric to the project cost[6]. It helps to generate the repeatable estimations and the formulae can be customized. Expert judgment techniques involve consulting with software cost estimation expert or a group of the experts to use their experience and understanding of the proposed project to arrive at an estimate of its cost [7,8]. The main limitation of expert judgment techniques is that the estimate is only a good expert's opinion and is hard to document the factor used by the experts. The estimate of the effort to complete a new software project is made by analogy with one or more previously completed projects. Bottom-up estimates use a detailed project design to list all the materials and labor required to complete the project. In top-down estimating method, an overall cost estimation for the project is derived from the global properties of the software project, and the project is partitioned into various low-level components. COSYSMO (Constructive Systems Engineering Cost Model) more accurately estimates the time and effort associated with performing the system engineering tasks in complex systems. This model is to estimate the Systems Engineering effort for large-scale systems (both software and hardware). In order for COSYSMO to be effective, it must be calibrated to the organization and types of projects being targeted. The COSYSM Data Collection Form gives an idea of the type of information needed for calibration.

AI is a branch of science which deals with machine acting like a human because the machine also finds solution to complex problems. Computational intelligence is the study of intelligent agents. Here we apply fuzzy logic technique on analogy to produce accurate cost estimation. Fuzzy logic is a logical system, which is an extension of multi-valued logic. However, in a wider sense refers to all of the theories and technologies that employ fuzzy sets, which are classes with imprecise boundaries. Fuzzy logic ranges between 0 and 1. Fuzzy sets are an extension of crisp sets to handle the concept of partial truth,which enables the uncertainities of natural language. The memership function is the essence of fuzzy set. A membership function is a characteristic function of the fuzzy set. Fuzzy set associated with the degree of membership function. Two valued sets are also characterized by a membership function.This function must be bounded from below by 0 and from above by 1.

Software design continuously changes because requirements is not stable. In this stage we can't find any fixed estimation cost results .To avoid this scenario we can not pick the projects with random error in dataset. When we apply analogy in software cost estimation, three parameters such as complexity, KDSI and deveopment time are important to rectify errors. This paper is organized as follows- section 2 describes research methodology and how fuzzy system is helpful in estimation. Section 3 elaborates the experimental results with fifteen projects using fuzzy system. Section four extends comparison of algorithmic cost modeling. Finally the paper concludes in section five.

II. FUZZY SYSTEM FOR COST ESTIMATION

The Artificial Intelligence is the branch of science which deals with machine acts like a human because the machine also finds solution to complex problems like Robots[8]. The Fuzzy logic algorithm is a better solution to implement in analogy to get accurate results in software cost estimation[1].Software project managers require reliable methods for estimating software project costs, and it is especially important at the early stages of software cycle. For this purpose, analogy based software cost estimation has been considered as a suitable alternative to regression-based estimation methods. Analogy based estimation is another technique for early life cycle macro-estimation. Analogy based estimation involves selecting one or two completed projects that more closely match the characteristics of our planned project. The chosen project or analogues are then used as the base for our new estimate. This leads to accurate software cost estimation. In our paper, we derive fuzzy values based on a set of input parameters using membership function and comparing it with the membership function values for COCOMO and functional point models. There are three membership functions – Triangular [5], trapezoidal and Gaussian. In our research we have chosen triangular membership function.

Fig 1 shows that fuzzy system getting the input parameter with the help of datasets and processed using triangular membership function and produced minimal random error data from dataset. Software engineering repository data set has been made publicly available in order to use it repeatedly and is certified set of data. We follow the acknowledgment guidelines posted on the PROMISE repository website-http://promise.site.uottawa.ca/ SE Repository. Fuzzy logic is a logical system, which is an extension of multi-valued logic .It gets, the input from various datasets and uses the fuzzy rule to eliminate unnecessary parameters from dataset. Three important parameters are involved in membership function calculation, namely, KDSI, complexity and development time.

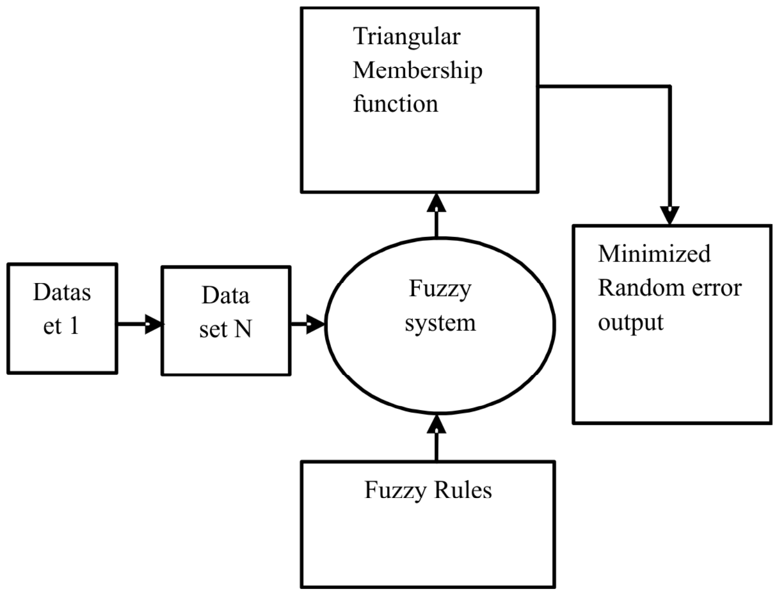

Figure 1: Project Architecture diagram

The calculated value is used to eliminate projects with random errors in the dataset. The proposed algorithm are given below

STEP 1: Consider each row as a single project and every column as parameter for the project.

STEP 2: Check whether the values for the input parameters are not null in the database and if not null, retrieve the non-null data using some search technique.

STEP 3: Assign a variable name for every input parameter. In our case, let x denote KDSI, a denote the number of people, b denote development time and c denote complexity.

STEP 4: Create the fuzzy value based on the input parameter using fuzzy formula.

STEP 5: Find out triangular membership function using the formula:

$$f(x;a,b,c) = \begin{Bmatrix} 0, & x \le a \\ \frac{x-a}{b-a}, & a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{Bmatrix}$$

STEP 6: Calculate the triangular membership function for every project (each row in the dataset).

STEP 7: Finally, compare the values against those for COCOMO and functional point models.

Figure 2: Triangular membership function formula

We create a table considering each row as a project and the various columns as the inputs for the projects. The table is arrived at by using a suitable search technique on the dataset ensuring non-null values for KDSI, number of people and development time. We assign variable names for the input parameters such as 'a' for number of people, 'b' for development time, 'c' for complexity and 'x' for KDSI. We then derive fuzzy values using triangular membership function f(x:a,b,c) for every project and compare them against the triangular membership values for COCOMO and functional point models.

III. EVOLUTION CRITERIA

Fuzzy values derived using triangular membership function serve as unique values for every project in the dataset. At any stage during software cost estimation, these fuzzy values can be used to uniquely identify a project from the dataset. Triangular membership function is a fast method of computation of fuzzy values for a data set. Also it requires three input parameters which match easily to our research requirements. The output from a triangular membership function is always a fuzzy number which is a mandate to process the idea in our paper. Cost estimation can be done using algorithmic and non-algorithmic methods. Non-algorithmic approach is supported by COCOMO and functional point models, while algorithmic approach is used by analogy based estimation. To be able to overcome random errors in non-algorithmic models, agile requirement change management process is implemented. However, for algorithmic models such as analogy, the identification and manipulation of random errors in projects of any promised dataset is still under research. Sometimes analogy fails due to many reasons like estimator knowledge, similarity measure and measurement error. Fuzzy logic is a better solution to implement in analogy to eliminate the random error. Triangular membership function helps to map fuzzy values to real numbers using Fig 1, thereby providing fuzzy values as reference values in the dataset. Dataset represents the data which is collected from promised data. The original data contains about 106 data with 94 attributes, but for research purpose, we have taken three attributes such as KLOC, complexity and time for fifteen projects in the dataset. We calculate triangular membership function using the above three attributes on the below formula:

Triangular Membership function = (complexity-kdsi) / (complexity-time)

In table 1 below the triangular membership function shows values between 0 and 1, thereby conforming the range defined by fuzzy logic, except for the value in the 15th entry (-4.85437) which is beyond the reference range {0,1}. This implies that the 15th row in the dataset is bound to have more random errors.

Table 1
Fuzzy value of triangular membership function for each project

S.No	*Kdsi*	*Time*	*Comp*	*Volume*	*Triangular*
1	2828	50969.11	371	38600.75	0.048559
2	1221	22480.91	183	18081.27	0.046551
3	1522	7318.61	152	15736.22	0.191164
4	1408	22958.05	149	20114.74	0.055197
5	504	5723.73	82	7030.53	0.0748
6	404	2836.53	50	5470.77	0.12704
7	293	3844.61	43	3647.52	0.065762
8	381	3845.5	46	4762.02	0.088169
9	242	3026.06	36	2690.94	0.068895
10	232	2606.7	35	2556.63	0.076603
11	268	1919.8	29	2964.65	0.126402
12	218	228.34	27	1192.97	0.948644
13	132	2391.13	12	2324.76	0.050439
14	649	12268.3	87	19227.48	0.046136
15	10	3.97	5	11.22	-4.85437

IV. COMPARISON OF ALGORITHMIC COST MODELING

We apply the fig 2 formula in original dataset and then we derive triangular membership function value using fuzzy logic. The graphical view of triangular membership table is represented below:

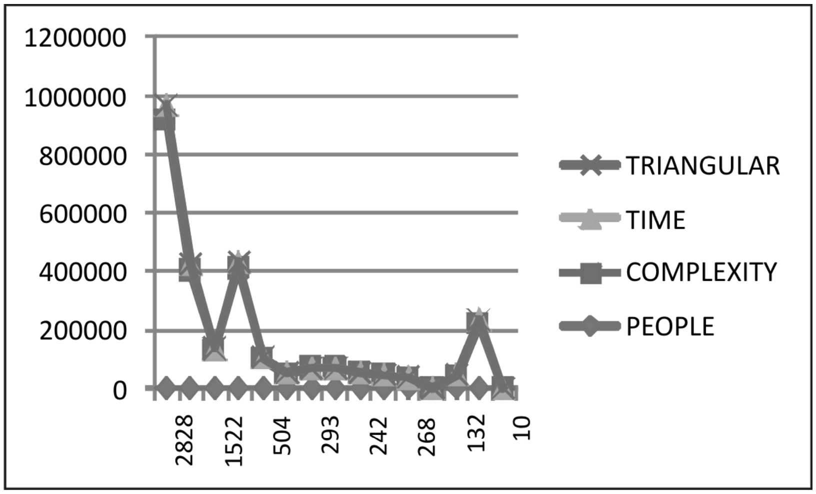

Figure 3: **Kdsi Vs Triangular Membership Function**

From the graph above(Fig 3), we understand that for higher values of KDSI, the values of time, number of people and complexity is high as well and vice-versa. Therefore, analogy based estimation requires KDSI as a parameter of mandate as all other parameters (time, complexity, number of people) are directly dependent on it. In the below table 2, we compare the value of triangular membership function as applied on COCOMO model, functional point and analogy dataset. Triangular membership function is applied for the same value of KDSI (649) and differing values of time, complexity and number of people. We observe that the fuzzy value of triangular membership function for analogy based estimation is within the bounds of a fuzzy value. Triangular membership function for COCOMO and functional point model because the former supports estimation during early stages of the project while the later supports estimation at the middle of software development.

Table 2
Comparison between algorithmic and non-algorithmic model

Model	*Kdsi*	*Time*	*People*	*Tri*
COCOMO	649	45.27503268	188.4244969	0.928853
Analogy	649	12268.3	87	0.046136
Fun point	649	51.0595531	0.0125	0.782774

V. CONCLUSION

In this paper, we have used the data set from many organizations that investigate the impact of various factors on software development outcomes. We found that software size was a major factor that affected development effort, people, and time. In this paper, we can deal with software cost estimation using fuzzy logic to derive accurate effort estimation; our tool uses triangular membership function to extract projects with more random errors from the data set. Random errors occur when the requirements to a project undergo continuous change. In this paper, triangular membership function has been used to identify those projects which have been undergoing continuous requirement changes through the development.

REFERENCES

[1] Mohammad Azzeh, Daniel Neagu, Peter I. Cowling, "Analogy-based software effort estimation using Fuzzy numbers," The Journal of Systems and Software 84 (2011) 270–284, Elsevier – 2010

[2] M. Kazemifard, A. Zaeri, N. Ghasem-Aghaee, M.A. Nematbakhsh, F. Mardukhi , " Fuzzy Emotional COCOMO II Software Cost Estimation (FECSCE) using Multi-Agent Systems," Applied Soft Computing 11 (2011) 2260–2270, Elsevier - 2010.

[3] Seok-Beom Roh, Sung-KwunOh, WitoldPedrycz, "Design of fuzzy radial basis function-based polynomial neural networks," Fuzzy Sets and Systems 185 (2011) 15–37. Elsevier - 2011.

[4] *http://www.ifpug.org*.

[5] Ali F. Jameel, Ahmad Izani Md. Ismail and Amir Sadeghi, "Numerical Solution of Fuzzy IVP with Trapezoidal and Triangular Fuzzy Numbers by Using Fifth Order Runge-Kutta Method," World Applied Sciences Journal, 17 (12): 1667-1674, 2012.

[6] Jacky Wai Keung, Barbara A. Kitchenham and David Ross Jeffery, "Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation," IEEE transactions on software engineering, vol. 34, no. 4, july/august 2008.

[7] Deepa Gangwani1, Saurabh Mukherjee2,"Analyzing the Impact of Different Factors on Software Cost Estimation in Today's Scenario" Journal of Software Engineering and Applications, 2015, 8, 245-251.

[8] Farhad Soleimanian Gharehchopogh, Awat Maroufi, "Approach of software cost estimation with hybrid of imperialist competitive and artificial neural network algorithms" Journal of Scientific Research and Development 1 (1): 50-57, 2014.