

Minimizing Analogy Errors with the Help of Fuzzy

D.Manikavelan

*Department of Computer Science and Engineering, Research scholar at Sathyabama University,
Working as an Associate Professor in Dhanalakshmi College of Engineering, Chennai, India,*

R.Ponnusamy

*Department of Computer Science and Engineering, Professor,
CVR college of Engineering, Telangana, India.*

Abstract

The current study is about minimizing errors in software cost estimation when analogy technique is used for the estimation. Project and product based companies face lot of problems during cost estimation because of requirement changes from time to time which in-turn changes the utility cost. Overestimation and underestimation lead to complete collapse and failure of the project at earlier stages itself. Analogy based estimation is considered to be the best methodology in algorithmic modeling. Analogy is to create accurate estimates for the proposed project by comparing the proposed project to similar projects from the past. While doing a comparison in analogy we face lot of problems at runtime. To overcome analogy errors such as observational error, trapezoidal membership function is used. When an estimator is new to analogy, the person is prone to become suspicious about choosing the right set of projects to derive a comparison. In this paper, we analyze how S-Membership function could be used to deal with such scenarios.

Keywords: Analogy, Trapezoidal membership function, S-Membership function, Kilo lines of code

INTRODUCTION

Software cost estimation method is comparing the features of the method based on clustering abilities and also useful for selecting the best set of parameters for each project. There are two approaches in cost estimation, algorithmic approach and non-algorithmic approach. Constructive Cost Model (COCOMO) and Function Point Model (FPO) are non-algorithmic approaches. COCOMO is an extensively used software cost estimation model which estimates cost based on effort and schedule [2]. It is derived from statistical regression of data from past projects. Advantage of COCOMO model is that it is transparent, one can see how it works unlike other models such as SLIM and it is repeatable and easy to use. Disadvantage of COCOMO is it ignores customer skills, cooperation and knowledge [5,7]. FPO is one of the best methods for measuring functional size of software. Function Point metrics is a standardized method for measuring the functions of a software application. This method measures the complexity and size of a software system based on the

methods developed in the project, because each user requirement maps to an end-user function. An important aspect of functional point is that most of the clients are already familiar with it; however, it ignores the output quality, also it is very tough to automate and is quite complex to compute. In algorithmic approach, cost estimation is done based on product, project and process attributes which are equated in a mathematical function. The product attribute more commonly used is code size. Algorithmic model is developed using cost information of projects developed in the past relating some software metric with the project cost [8,9]. The main advantage of algorithmic cost modelling is that it produces repeatable estimations and easily adjustable input data; also it is easy to refine and customize the formulae. The disadvantage of algorithmic cost modelling is that some factors cannot be measured and algorithms may be proprietary. Skillful judgment involves seeking software cost estimation experts'. Knowledge and experience to derive the estimate for the proposed project. Amateurs to the field of cost estimation could make use of the expertise and skill of the experts to provide estimates based upon all the projects that the expert had contributed earlier. This is because experts can comfortably factor in differences between past projects and requirements of the proposed project. The estimate of the effort for a new software project is made by analogy with one or more previously completed projects. Application of the concept used in bottom-up estimates over analogy provides better results. This is because bottom-up estimates use a comprehensive project design to list all the resources and effort required to finish the project. Costs for each module or activity is done until the bottommost level and then augmented up to topmost levels for reporting and follow-up. The other advantage is that it permits the software group to handle an estimate in an almost traditional fashion and to handle estimate components for which the group has a feel.

Artificial Intelligence (AI) is the branch of science which deals with machine acts like a human because the machine also finds solution to complex problems like Robots [3,4]. CI is the study of intelligent agents. An intelligent agent is a system that acts intelligently which is flexible to changing environments and changing goals. The purpose is to understand how intelligent behaviour is possible. It is about the ability to study, understand and compute based on

changing necessities[10,11]. Many techniques are given in Computational intelligence; here we apply fuzzy logic technique which implements analogy to produce accurate cost estimation[1]. Fuzzy system is two-valued or Boolean, well defined and used in this theory. Boolean logic is important for implementation in computing system. Boolean logic plays an important role for development in AI reasoning system and associated with the two value set theory, where an element either belongs to a set or not. While some success is achieved using a two-valued logic [12], it is however not possible to solve all problems by mapping the domain into two-valued variables. Most of the problems are owing to incomplete process or uncertain information. While two valued logic and set theory failed in such environments, fuzzy logic and fuzzy sets give the formal tools to reason about uncertain information [13].

Membership function for cost estimation

In our research on analogy, fuzzy value is included as a parameter in the dataset against each project. Datasets contain more than 72 parameters for every project. Fuzzy value is manipulated for the projects to help classify based on its parameters. For instance, projects with lines of code ranging up to 1000 are grouped under a single fuzzy value. Successively, the ranging for the next set of thousand projects and the fuzzy values for every set are done based on the same parameter.

Algorithm

- Step1: Assume each row as a single project.
- Step2: Assume each column as the parameter for the project.
- Step3: Check whether the values in the database are not null. If it is not null, retrieve the result from the database using some search algorithm.
- Step4: Introduce a variable name for every input parameter
- Step5: Create a fuzzy value based on the input parameter using fuzzy formula.
- Step6: Find out trapezoidal membership functions using fuzzy logic.

$$f(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases}$$

Figure1: Trapezoidal formula

Step 7: Find out S membership function using fuzzy logic

$$f(x; a, b) = \begin{cases} 0, & x \leq a \\ 2 \left(\frac{x-a}{b-a} \right)^2, & a \leq x \leq \frac{a+b}{2} \\ 1 - 2 \left(\frac{x-b}{b-a} \right)^2, & \frac{a+b}{2} \leq x \leq b \\ 1, & x \geq b \end{cases}$$

Figure2: S-Membership formula

Step 8: Generate the fuzzy number for each project.

Algorithm shows create a database and initialize the values in the database and step1, 2 explained that we assume each row as a single project and column as the parameters for the project. step3, 4 shows that We check whether the input parameters are not null and retrieve result using optimization technique and use a variable name for every input parameter, say, ‘a’ for effort, ‘b’ for people, ‘c’ for time, ‘d’ for volume and ‘x’ for KDSI. step5 confirmed that Create the fuzzy value by calculating based on the input parameters using the fuzzy formula to generate the triangular membership function. step6 shows that Create the fuzzy value by calculating based on the input parameters using the fuzzy formula to generate the triangular membership function and Trapezoidal membership function value is observed and analyzed to identify failed estimates in data set. Step7 explained that S-Membership function values are used by new estimators (in field of analogy) to avoid indolent observation in data set.

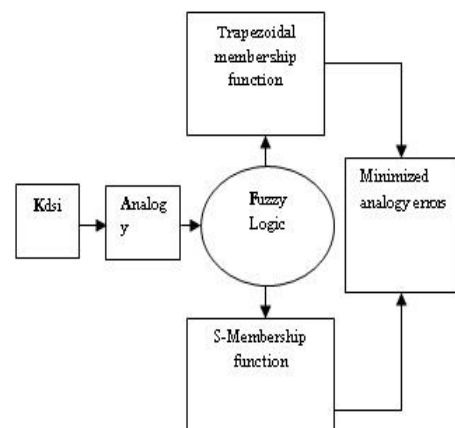


Figure 3: Architecture Diagram

Architecture diagram in figure 3 represents the concept used to minimize analogy errors which is discussed in this paper. KLOC is an important parameter in cost estimation this is considered primary key in analogy based estimations .the architecture diagram shows KLOC and analogy parameters (effort, people, time, complexity) being fed as inputs to fuzzy membership functions. Trapezoidal membership function minimizes analogy errors by helping the estimator retrieve

only successful estimations from the dataset thereby helping the estimator ignore failed estimations in the dataset. S-Membership function could be used by fresh estimators in analogy to get rid of possible errors that the estimator is prone to make. Analogy based estimation produces an accountable number of manipulative errors, two of such errors and the ways to overcome the errors has been discussed in this paper.

Evaluation criteria

In analogy based estimation fails as many reasons such as lack of transparency which analogy estimator using hand calculations .last minute requirement changes is the important parameters of failed estimated projects because of many project requirements is not stable. Failing to allocate enough resources in estimators and estimator not creating risk factors correctly .In this paper we will eliminate some analogy estimate errors.

Create the fuzzy value based on the input parameter using the fuzzy formula to generate each membership function and calculate it. The membership functions are, Triangular Membership Function, Trapezoidal Membership Function and S- Membership Function.To implement the fuzzy logic using membership functions and Generate the fuzzy value for each project. After that, compare the best estimation then to get the minimum value.

Table I: Retrieved dataset values

S.No	Kdsi	Trapezoidal membership	S-membership
1	2828	0.935730681	0.993826
2	1221	0.942005568	0.993811
3	1522	0.912090563	0.995659
4	1408	0.936941982	0.993902
5	504	0.93926773	0.994245
6	404	0.934695624	0.994911
7	293	0.930642638	0.994094
8	381	0.928965526	0.994383
9	242	0.922408793	0.994142
10	232	0.921875929	0.994257
11	268	0.918587025	0.994877
12	218	0.836187895	0.999986
13	132	0.948113942	0.993823
14	649	0.970638145	0.993792
15	384	0.966037058	0.997146

Comparison of algorithmic cost modeling

In our paper, we apply the rule of Trapezoidal Membership function that all the side values (input parameters) are different and that none of the values matches with the other. In analogy, observational error is the difference between the calculated value of quality and the actual value. Measured value differs from the actual value when random errors are present. By getting devoid of random errors, we could avoid wrong observation thereby increasing the degree of accuracy.

$$\text{Trapezoidal Membership function} = \text{Volume KDSI} / \text{Volume-complexity}$$

Our study uses fuzzy approach in retrieving those projects in the data set that have minimal observational/random errors[6]. For the dataset represented in Table 1 we apply the rule of trapezoidal membership function as figure 1 that all input parameters be different from each other. We assume that the parameters KDSI, no. of people, Effort and volume mentioned in table 1 are measured values and Trapezoidal membership function values are the true values. Referring the projects with approximately nearer KDSI values 232 and 242, the corresponding trapezoidal membership function values are closer as well, 0.921 and 0.922 respectively. Thus taking into account the TMF value for majority of projects with similar KDSI, the estimator could conclude to ignore a project if the TMF value deviates significantly .S-Membership function as figure 2 is used to avoid indolent observation in data set. A new estimator is prone to make errors in choosing the right set of projects from the data set. Errors could be caused when the estimator uses a software to perform the operation of choosing the apt projects from data set, does manual calculation, fails to include all the necessary resources in the estimate, does not take into account requirement changes, allocates inaccurate cost for labors, excludes risk factors in project cycle, lacks knowledge on relevant parameters etc. SMF helps a new estimator overcome above errors they are bound to make while choosing projects from dataset. Our study uses the SMF formula on the projects in the data set shown in table 1. When the KDSI value is lesser than or equal to volume, the SMF value is taken to be zero. If this case is false, we compute whether KDSI falls between effort and the average value of effort and time. If this case is again not true, we check whether KDSI falls between the above said average value and the time itself. Under conditions when none of the above cases are satisfied and KDSI is greater than time, we take it to be 1. The outcome of the substitution of SMF formula on all projects in the dataset in table 1 shows that all projects have similar values of SMF and that is approximately 0.99. A new estimator could therefore conclude that the dataset is reliable.

CONCLUSION

Software cost estimation plays a major role in driving a project to success. An imperfect estimation produces as much an impact that the outcome may completely fail to meet the actual project expectations. In order to overcome this, it is important for an estimator to choose the best estimation technique from all available methodologies. Our research is about cost estimation which is analogy based and which

compares data from projects executed in the past. However, there are lots of parameters to be taken into account for estimation using analogy. Some of those influencing factors while performing analogy based estimation have been analyzed and solutions to overcome the challenges are discussed in this paper. An analogy based estimator is always left with the challenge of exactly retrieving/locating only those projects from data set that had truly met with the estimates. Here, we deploy trapezoidal membership function to retrieve only the successful projects from data set and ignore the failed ones. An amateurish estimator is always left with lots of challenges when using analogy based estimation. S- Membership function helps new estimators overcome any errors they are prone to make while implementing analogy for cost estimation.

REFERENCES

- [1] 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.
- [2] 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
- [3] Adriano L.I. Oliveira, Petronio L. Braga, Ricardo M.F. Lima, Márcio L. Cornélio "GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation" Information and Software Technology 1155–1166 Elsevier 2010
- [4] Marappagounder Krinamoorithy Shanker Ganesh, Keppangounder Thanushkodi, "An Efficient Software Cost Estimation Technique using Fuzzy Logic with the Aid of Optimization Algorithm", International journal of innovative computing ,587-597,2015
- [5] Barry W. Boehm, Ricardo Valerdi "Achievements and Challenges in Cocomo-Based Software Resource Estimation" IEEE Software, Vol.25, Issue.5, PP. 74 – 83, 2008
- [6] 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.
- [7] Asli Celikyilmaz, Member, IEEE, and I. Burhan Turksen, "Enhanced Fuzzy System Models With Improved Fuzzy Clustering Algorithm," IEEE transactions on fuzzy systems, vol. 16, no. 3, June 2008.
- [8] Ekrem kocaguneli, and Ayse basar bener, "Exploiting the Essential Assumptions of Analogy-Based Effort Estimation" IEEE transactions on software Engineering vol. 38, no. 2, April 2012.
- [9] VahidKhatibi, Dayang N. A. Jawawi, "Software Cost Estimation Methods: A Review", Journal of Emerging Trends in Computing and Information Sciences, Vol.2, No. 1, PP.21-29, 2010.
- [10] Anurag Kumar, AmitSinha, BhupendraVerma, "A Novel Technique of Optimization for Software Metric Using PSO", International Journal of Soft Computing and Software Engineering (JSCSE), Vol.3, No.10, PP. 2013.
- [11] Attarzadeh, I. Siew Hock Ow, "Proposing a New Software Cost Estimation Model Based on Artificial Neural Networks", IEEE International Conference on Computer Engineering and Technology (IC CET) , Vol.3, PP.V3, PP.487-491 2010.
- [12] 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.
- [13] Nikolaos Mittas and Lefteris Angeliis, "Ranking and Clustering Software Cost Estimation Models through a Multiple Comparisons Algorithm," IEEE Transaction on Software Engineering, Vol. 39, No. 4, April 2013.