Survey: Non Algorithmic Models for Estimating Software Effort

Ridhika Sharma
Aravali college of Engineering and Manegment
ridhikasharma@gmail.com

Abstract*:*

Software cost and effort estimation is the most vulnerable issue in software development technology. Due continuous changing resources and customer involvements, software effort estimation has become more challenging issue in software management. To accurately estimate software cost and effort is difficult for both developers as well as customers. It has become very difficult to establish relationship between attributes of projects and its effort estimates through algorithmic models like COCOMO. Although non algorithmic models like those of neural networks are good in building relationships and pattern in the data. This paper will present a bird eye view on various tools to predict and analyze different techniques in cost and effort estimation.

Keywords: *Genetic Algorithms,* COCOMO, *Neural Network learning techniques*

1. Introduction

Software cost and effort estimation is one of the most challenging issues in software project management. Several estimates are involved to effectively manage the software cost. It has become objective of every software engineering community to develop useful models that can accurately estimate the software effort. One of the most widely used technique is COCOMO (constructive cost model) introduced by Barry Boehm in 1981, and is still in use by software engineering community.

Software development efforts estimation is the process of predicting the most realistic use of effort required to develop or maintain software based on incomplete, uncertain and/or noisy input. Effort estimates may be used as input to project plans, iteration plans, budgets, investment analyses, pricing processes and bidding rounds. Most of the research has focused on the construction of formal software effort estimation models. The early models were typically based on regression analysis or mathematically derived from theories from other domains. COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. The first level, Basic COCOMO is good for quick, early, rough order of magnitude estimates of software costs, but its accuracy is limited due to its lack of factors to account for difference in project attributes (Cost Drivers). Intermediate COCOMO takes these Cost Drivers into account and Detailed COCOMO additionally accounts for the influence of individual project phases.

Basic COCOMO computes software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of source lines of code (SLOC)

COCOMO applies to three classes of software projects:

1. Organic projects - "small" teams with "good" experience working with "less than rigid" requirements.
2. Semi-detached projects - "medium" teams with mixed experience working with a mix of rigid and less than rigid requirements.
3. Embedded projects - developed within a set of "tight" constraints. It is also combination of organic and semi-detached projects.(hardware, software, operational, ...)

The basic COCOMO equations are:
Effort Applied (E) = ab (KLOC) bb [man- months]
Development Time (D) = cb (Effort Applied) db [months]
People required (P) = Effort Applied / Development Time [count]

Where, KLOC is the estimated number of delivered lines (expressed in thousands) of code for project.
Intermediate COCOMO computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes. This extension considers a set of four "cost drivers", each with a number of subsidiary attributes:-

1. Product attributes
   a. Required software reliability
   b. Size of application database
   c. Complexity of the product
2. Hardware attributes
   a. Run-time performance constraints
   b. Memory constraints
   c. Volatility of the virtual machine environment
   d. Required turnabout time
3. Personnel attributes
   a. Analyst capability
   b. Software engineering capability
   c. Applications experience
   d. Virtual machine experience
   e. Programming language experience
4. Project attributes
   a. Use of software tools
   b.Application of software engineering methods
   c.Required development schedule

The Intermediate Cocomo formula is:

E=ai (KLoC) (bi).EAF

Where E is the effort applied in person-months, KLoC is the estimated number of thousands of delivered lines of code for the project, and EAF is the factor calculated above.

Detailed COCOMO 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.

The detailed model uses different effort multipliers for each cost driver attribute. These Phase Sensitive effort multipliers are each to determine the amount of effort required to complete each phase.

In detailed COCOMO, the effort is calculated as function of program size and a set of cost drivers given according to each phase of software life cycle.

A Detailed project schedule is never static.

The five phases of detailed COCOMO are:-

(i) Plan and requirement. (ii) System design, (iii) detailed design, (iv) module code and test, (v) integration and test.

In order to make accurate estimates cost estimation techniques are divided into two main categories (1) Parametric Models or Algorithmic Models that are derived from numerical analysis of historical projects data (2) Non Parametric or Non Algorithmic Models based on set of artificial intelligence techniques like neural networks , genetic Algorithm, rule based induction, etc. This paper discusses various non parametric cost estimation techniques.

2. Literature Review

Many researchers used their different non algorithmic models and different data sets to predict the software effort more correctly. Their comes out to 3 major categories of non algorithmic models a brief overview of related work is discussed in this paper.

2.1. Neural network based models to estimate software development effort:

Most of the work in the application of neural network to estimate effort use backpropogation algorithm and cascade correlation network. [10]. ANN is a network of non linear computing elements called neurons which model the functionality of human brain. Anjana Bawa [10] proposed a general ANN architecture composed of 8 basic components. (i) Neurons, (ii) activation function, (iii) signal function, (iv) pattern of connectivity, (v) activity aggregation rule, (vi) activation rule, (vii) environment. Figure 1 [10] shows the architecture of an artificial neuron.

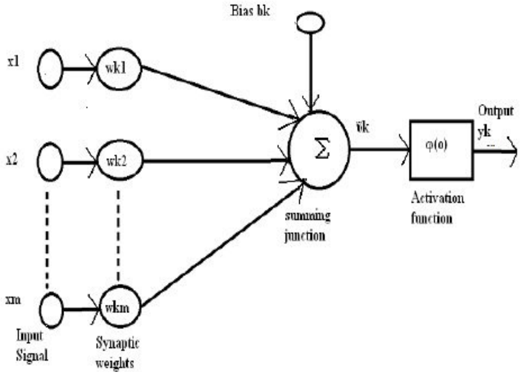

Figure-1 Architecture of an artificial neuron.

A.R.Venkatchalam [1] used COCOMO data set and back propogation algorithm to model the software cost estimation and the results are compared with COCOMO model. ANN used in this paper uses sigmoid activation function and selection of hidden layers are based on heuristic knowledge. The model implemented by Anupama Kaushik, et al. [5], is trained using perceptron learning algorithm. The test results from the trained neural network are compared with COCOMO model. Nasser Tadayon [3] explained the use of expert judgment and machine learning technique using neural network as well as referencing COCOMO II approach to predict the cost of software. Figure 2 [3] shows the network connection used in this paper.

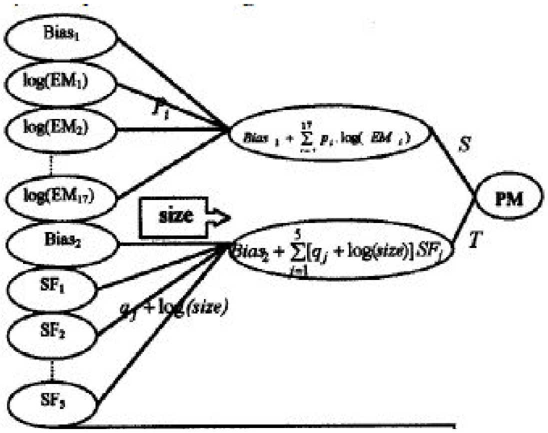

Figure-2 Network connection

In a paper by Ali Idri, et al. [2], COCOMO’81 data set is used to train and test the Radial basis feed forward network. It was observed in the paper that the accuracy of RBFN depends entirely on parameters of middle layer, especially no. of hidden neurons and value of width. Iman Attarzadeh, et al. [4], used COCOMO I, NASA 93 data set to produce accurate software estimate. The NN used back propogation algorithm to test

156 projects and 8.36% improvement in cost estimation was observed in the modal as compared to original COCOMO II. Ali Bou Nassif [11] proposed a novel Cascade Correlation Neural Network Model to predict software effort from use case design. The input of ANN is software size, productivity and project complexity. In this paper multiple linear regression and CCNN model were trained using 168 projects and evaluated using 72 projects. Y-S Huang et al. [15], in his paper described a method to construct an RBFN classifier efficiently and effectively. The method determined the middle layer neurons by a fast clustering algorithm and computes the optimal weights between the middle and the output layers statistically using a statistical approach called linear discriminant function to determine the weights. Ch.Satyananda Reddy [6] adopted feed forward multilayer perceptron with linear activation function to avoid slow convergence problem that is a drawback of sigmoid activation function. Fig 3 [6] shows the model architecture.

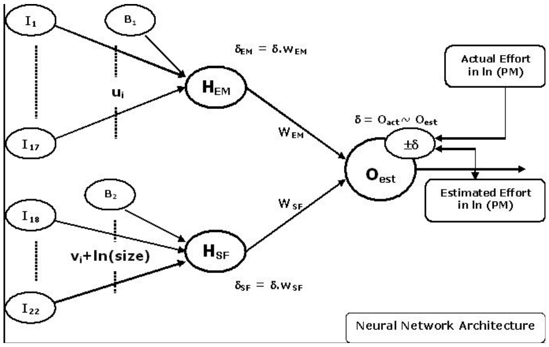
I₁
B₁
$\delta_{EM} = \delta.w_{EM}$
Actual Effort in ln (PM)
u_i
H_{EM}
I_{17}
$\delta = O_{act} \sim O_{est}$
W_{EM}
±δ
O_{est}
B₂
I_{18}
W_{SF}
Estimated Effort in ln (PM)
v_i+ln(size)
H_{SF}
$\delta_{SF} = \delta.w_{SF}$
I_{22}
Neural Network Architecture

Figure-3 Architecture of the proposed neural network model.

ANN model proposed by Prasad Reddy, et al. [17], is created using Radial Basis and Generalized Regression. The model is based on COCOMO'81 database and the results are compared with intermediate COCOMO and it is observed that the RBF provides better results. In the paper by Manpreet Kaur et al.[16], efficiency of Bayesian regulation back propogation based cost estimation model is studied The dataset of NASA is used in the research and the results showed that the Bayesian Regulation Back propogation system has the lowest MMRE to the actual. Ch.Satyananda Reddy et al. [18] trained and tested the RBF network using COCOMO'81 dataset. It had been observed that the RBFN designed with K-means clustering algorithm performs better in reference to software effort estimation. B.Tirimula Rao et al. [25], proposed an efficient Functional Link ANN to reduce the computational complexity so that the network becomes suitable for on line applications.

2.2. Fuzzy Logic Based Models to calculate software effort:

A fuzzy model usually is more apt when the data is uncertain, inaccurate and vague. In the paper presented by Prasad Reddy et al. [21], software development effort prediction using Fuzzy triangular Membership Function and Gbell Membership function was implemented and compared with COCOMO.NASA93 data sets were used to model these functions and intermediate COCOMO model was used for developing the Fuzzy Inference System. Pooja Jha et al. [24] focused on comparative analysis of COCOMO 81 using various fuzzy membership functions. It was observed that fuzzy based COCOMO model was better in estimating effort as compared to COCOMO. Abeer Hamdy [22] incorporated fuzzy component to enhance the accuracy and sensitivity of COCOMO intermediate model. In this COCOMO NASA 2 data set was used to evaluate the proposed fuzzy model and the results showed that the sensitivity of the proposed fuzzy model was superior to the COCOMO intermediate model. Due to the drawback of trapezoidal function where high degree of compatibility values were assigned to attributes instead of lower degree values, Ch.Satyananda Reddy et al. [23], proposed to use Gaussian Membership function for cost drivers and it was observed that the latter function demonstrated smoother transition in its intervals and results were very much closer to actual effort. Sandeep Kad, et al. [27], explored a soft computing based technique to overcome the problems of uncertainty and imprecision resulting in improved process of software development effort estimation. Iman Attarzadeh et al. [28], aimed to propose a fuzzy logic realistic model to achieve more accuracy in software effort estimation. This model is validated by two approaches using COCOMO data set using 63 projects and artificial data set consisting of 100 sample projects.

2.3. Genetic Algorithm approach for effort estimation:

Alaa F.Sheta [26] used Genetic Algorithm on NASA software project data set to estimate effort of software. Brajesh Kumar Singh et al. [20] investigated the effect of crisp inputs and genetic algorithm technique on NASA data set applied on 18 projects. Kavita Chaudhary [19], explored the inter relationship among different dimensions of software models. It was observed that with the use of genetic algorithm, effective effort could be done. Recardo de A.Aroujo et al. [29] presented a hybrid intelligent model to design the Morphological Rank Linear perceptrons to solve the software cost estimation problem using modified Genetic Algorithm with Gradient Descent method to optimize the model.

3. Conclusion

It has been a challenging task to efficiently estimate the cost and effort of software projects in its early phases of development. Results from various research papers discussed shows that when neural network techniques are used MMRE and PRED values for RBFN and CCNN comes out to be relatively better [30]. In case of Fuzzy techniques, when Gaussian membership function is used improvement in MMRE and PRED values are observed [24].

4. Future Scope

The neural network model can be combined with fuzzy technique and a neuro-fuzzy model can be analyzed for effectively estimating software cost. The RBFN model [18] can be extended by integrating Fuzzy C-means clustering algorithm.

References

1. A.R. Venkatachalam, Software Cost Estimation Using Artificial Neural Networks. Proceedings of 1993 International Joint Conference on Neural Networks.
2. Ali Idri , Samir Mbarki , Alain Abran ,Validating and Understanding Software Cost Estimation Models based on Neural Networks
3. Nasser Tadayon, Neural Network Approach of Software Cost Estimation. Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC’05) 0-7695- -2315-3/05
4. Iman Attarzadeh , Amin Mehranzadeh , Ali Barati , Proposing an Enhanced Artificial Neural Network Prediction Model to Improve Accuracy in Software Effort Estimation , 2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks
5. Anupama Kaushik , Ashish Chauhan , Deepak Mittal, Sachin Gupta , COCOMO Estimates using Neural Networks , *I.J. Intelligent Systems and Applications,* 2012, 9, 22-28 Published Online August 2012 in MECS
6. Ch. Satyananda Reddy, KVSVN Raju, a Concise Neural Network Model for Estimating Software Effort, International Journal of Recent Trends in Engineering, Issue. 1, Vol. 1, May 2009
7. Xishi Huang, Luiz F. Capretz, Jing Ren, Danny Ho, Neuro Fuzzy Model for Software Cost Estimation Proceedings of the Third International Conference On Quality Software (QSIC’03)
8. Ali Idri , Alain Abran , Samir Mbarki ,An Experiment on the Design of Radial Basis Function Neural Networks For Software Cost Estimation
9. Harish Mittal , Pradeep Bhatia ,Optimization Criteria for Effort Estimation using Fuzzy Technique , CLEI ELECTRONIC JOURNAL, VOLUME 10, NUMBER 1, PAPER 2, JUNE 2007.
10. Anjana Bawa , Mrs.Rama Chawala, Experimental Analysis of Effort Estimation Using Artificial Neural Network
11. Ali Bou Nassif , Luiz Fernando Capretz, Danny Ho ,Software Effort Estimation In Early Stages of the Software life Cycle Using a Cascade Correlation Neural Network Model , 2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing

12. L.A. Zadeh, 2002, From Computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions, Int. J. Appl. Math.Comut.Sci., 2002, Vol.12, No.3, 307-324.
13. Vahid Khatibi, Dayang N. A. Jawawi , Software Cost Estimation Methods: A Review , Journal of Emerging Trends in Computing and Information Sciences , Volume 2 No. 1, ISSN
14. A. Idri, T. M. Khoshgoftaar and A. Abran. “Can Neural Networks be easily interpreted in Software Cost Estimation?” FUZZ-IEEE, Hawaii, 2002, pp. 1 162-1 166.
15. Y-S Hwang and S-Y Bang. “An Efficient Method to Construct a Radial Basis Function Network Classifier.” Neural Networks, vol. 10, no. 8, 1997, pp. 1495-1 503.
16. Mapreet Kaur, Sushil Garg.”Using Bayesian regulation back propogation algorithm for Software effort and comparing with COCOMO. IRACST – Engineering Science and Technology: An International Journal (ESTIJ), ISSN: 2250-3498, Vol.2, No. 4, August 2012.
17. Prasad Reddy P.V.G.D, Sudha K.R, Rama Sree P and Ramesh S.N.S.V.S.C. “Software Effort Estimation using Radial Basis and Generalized Regression Neural Networks. JOURNAL OF COMPUTING, VOLUME 2, ISSUE 5, MAY 2010, ISSN 2151-9617
18. Ch. Satyananda Reddy, P. Sankara Rao, KVSVN Raju, V. Valli Kumari.” A New Approach for Estimating Software Effort Using RBFN Network”. IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.7, July 2008.
19. Kavita Chaudhary “GA Based Optimization of Software Development Effort Estimation. IJCST Vol. 1, Issue 1, September 2010.
20. Brajesh Kumar Singh, A.K.Mishra “Software Effort Estimation by Genetic Algorithm Tuned Parameters of Modified Constructive Cost Model for NASA Software Projects” International Journal of Computer Applications (0975 – 8887) Volume 59– No.9, December 2012.
21. Prasad Reddy P.V.G.D, Sudha K.R, Rama Sree P “Application Of Fuzzy Logic Approach to software Effort Estimation” in (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, _o. 5, 2011.
22. Abeer Hamdy “Fuzzy Logic for Enhancing the Sensitivity of COCOMO Cost Model”, Journal of Emerging Trends in Computing and Information Sciences. VOL. 3, NO. 9, SEP 2012 ISSN 2079-8407.
23. Ch. Satyananda Reddy, KVSVN Raju “An Improved Fuzzy Approach for COCOMO’s Effort Estimation using Gaussian Membership Function” JOURNAL OF SOFTWARE, VOL. 4, NO. 5, JULY 2009.
24. Pooja Jha and K.S.Patnaik “Comparative Analysis of COCOMO81 using various Fuzzy Membership Functions” International Journal of Computer Applications (0975 – 8887).Volume 58-No.14, November2012.
25. B.Tirimula Rao , B.Sameet , G.Kiran Swathi , K.Vikram Gupta , Ch.Ravi Teja , S.Sumana “A Novel Neural Network Approach For Software Cost Estimation Using Functional Link Artificial Neural Network (FLANN).IJCSNS International Journal Of Computer Science And Network Security, Vol.9No.6,June2009.
26. Alaa F.Sheta “Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects” Journal of Computer Science 2 (2): 1118-1123, 2006 ISSN 1549-3636.
27. Sandeep Kad, Vinay Chopra “Fuzzy Logic based framework for Software Development Effort Estimation” Research Cell: An International Journal of Engineering Sciences ISSN: 2229-6913 Issue July 2011, Vol. 1.
28. Iman Attarzadeh, Siew Hock Ow “Proposing a new high performance model for Software Cost estimation “in 2009 Second international Conference on Computer and electrical Engineering.
29. Ricardo de A.Araujo, Adriano L.I.de Oliveira and Sergio Soares.”Hybrid Intelligent Design of Morphological Rank Linear Perceptrons for Software Development Cost Estimation. 2012 22nd International Conference on tools with Artificial Intelligence
30. Jagannath Singh , Bibhudatta Sahoo , “Software Effort Estimation with Different Artificial Neural Network” in proceedings of IJCA Special Issue on 2nd national Conference – Computing ,Communication and Sensor Networks.CCSN2011.