International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

Software Cost Estimation with the Help of Genetic Algorithm using Artificial Intelligence

Dileep Kumar
Dept of MCA
DRMLAU, Ayodhya, India
Dileep_k_2000@yahoo.com

Abstract: Software cost estimation is one of important activity of software development. Software cost estimation plays an important role in software engineering practice, often determining the success or failure of contract negotiation and project execution. Cost estimation's deliverables such as effort, schedule, and staff requirements are valuable information for project formation and execution. The more software becomes important in almost every human activity, the more it becomes complex and difficult to implement. Even if modern software technologies render easier the development of certain types of software products, increased user demands and new application domains produce additional problems. It is not surprising that software project management activities are becoming increasingly important. Effective software project estimation is one of the most challenging and important activities in software development. Estimation is one of the cornerstones of effective project planning: effective project planning and control is not possible without a sound and reliable estimate.

Keywords: AI, Cost Estimation, GA, Optimization.

1. Introduction:

It has been surveyed that nearly one-third projects overrun their budget and late delivered and two-thirds of all major projects substantially overrun their original estimates. The accurate prediction of software development costs is a critical issue to make the good management decisions and accurately determining how much effort and time a project required for project managers as well as system analysts and developers. Without reasonably accurate cost estimation capability, project managers cannot determine how much time and manpower cost the project should take and that means the software portion of the project is out of control from its beginning; system analysts cannot make realistic hardware-software trade-off analyses during the system design phase; software project personnel cannot tell managers and customers that their proposed budget and schedule are unrealistic. This may lead to optimistic over promising on software development and the inevitable overruns and performance compromises as a consequence. But, actually huge overruns resulting from inaccurate estimates are believed to occur frequently.

The overall process of developing a cost estimate for software is not different from the process for estimating any other element of cost. There are, however, aspects of the process that are peculiar to software estimating. Some of the unique aspects of software estimating are driven by the nature of software as a product. Other problems are created by the nature of the estimating methodologies. Software cost estimation is a continuing activity which starts at the proposal stage and continues through the lift time of a project. Continual cost estimation is to ensure that the spending is in line with the budget.

Cost estimation is one of the most challenging tasks in project management. It is to accurately estimate needed resources and required schedules for software development projects. The software estimation process includes estimating the size of the software product to be produced, estimating the effort required, developing preliminary project schedules, and finally, estimating overall cost of the project.

It is very difficult to estimate the cost of software development. Many of the problems that plague the development effort itself are responsible for the difficulty encountered in estimating that effort. One of the first steps in any estimate is to understand and define the system to be estimated. Software, however, is intangible, invisible, and intractable. It is inherently more difficult to understand and estimate a product or process that cannot be seen and touched. Software grows and changes as it is written. When hardware design has been inadequate, or when hardware fails to perform as expected, the "solution" is often attempted through changes to the software. This change may occur late in the development process, and sometimes results in unanticipated software growth.

After 20 years research, there are many software cost estimation methods available including algorithmic methods, estimating by analogy, expert judgment method, price to win method, top-down method, and bottom-up method. No one method is necessarily better or worse than the other, in fact, their strengths and weaknesses are often complimentary to each other. To understand their strengths and weaknesses is very important when you want to estimate your projects.

2. Related Work:

N. Veeranjaneyulu, S.Suresh, Sk.Salamuddin and Hye-jin Kim, (2014), a new classical methodology has been proposed here to estimate the normal software project effort. This methodology is focused around thinking by relationship, fuzzy logic and phonetic quantifiers, which can be viably utilized when the software activities are portrayed by all out as well as numerical information. The new approach enhances the traditional relationship technique while utilizing the clear cut data. In the fuzzy analogy approach, both unmitigated and numerical data are spoken to by fuzzy sets. The preference of this system is that it can deal with the imprecision and the instability distinctively while portraying the software project. From the execution of the

results, it is watched that the proposed strategy has successfully evaluated the normal effort for the software project datasets.

Stein Grimstad, et. al, (2006)A real-world example where common analysis of estimation error lead to a flawed conclusion, together with a review of published estimation error analyses in research studies, suggest that there is a need for better analyses of software cost estimation error. A framework that can be useful tool to improve the estimation error analysis is presented here. In particular, the checklist to identify non-studied factors with a potential biasing impact on the measured estimation error, the emphasis on proper estimation terminology, and the support on isolation strategies are useful. The framework does, however, not replace good analysis skill.

Lalit V. Patil, et. Al., (2014), there is so many models available categorized into algorithmic and non-algorithmic model each of their strengths and weakness. The authors proposed a hybrid approach, which consists of Functional Link Artificial Neural Network (FLANN) and COCOMO-II with training algorithm. FLANN reduces the computational complexity in multilayer neural network. It does not have any hidden layer, and it has fast learning ability.

FarooqAzam, et. al. (2014),proposed a technique for software size estimation. The basis for estimates, when using this method, is the available knowledge of the considered system. In order to capture and represent this knowledge, an object oriented functional model has been adopted. This functional model provides for a disciplined methodology for decomposing system complexity. This methodology is the key in the process of detailing the functionality of the system in order to enable estimators to achieve more reliable estimates. In this theory they investigated the effect of value prerequisites, specifically security and convenience on assesses made utilizing and master estimation. The objective was to explore the favorable circumstances and confinements of formal estimation systems vs. master estimation with respect to quality necessities.

MogiliUmamaheswara Rao, et. al. (2014),their main goal is to estimate the effort of various software projects using Class Point Approach. The parameters are optimized using various artificial intelligence (AI) techniques such as Multi-Layer Perceptron (MLP), KNearest Neighbour Regression (KNN) and Radial Basis Function Network(RBFN), fuzzy logic with various clustering algorithms such as the Fuzzy C-means (FCM) algorithm, K-means clustering algorithm and Subtractive Clustering (SC) algorithm, such as to achieve better accuracy. Furthermore, a comparative analysis of software effort estimation using these various AI techniques has been provided. However, the CPA is one of the different cost estimation models that have been widely used because it is simple, fast, and accurate to a certain degree. Fuzzylogic technique is further used to find out the complexity level of the class and to calculate optimized class point. Then the calculated class point values are being normalized and used to optimize the effort estimation result. The optimization is achieved by implementing different artificial (AI) techniques such as ANN, KNN, RBFN, and fuzzy logic system with different clustering algorithm using normalized class point value. Finally, the generated minimum results of different have been compared for estimating the performance of different models.

3. Methodology:

Today's Software development cost estimation models are based on soft computing techniques as neural network, genetic algorithm, the fuzzy logic modeling etc. for finding the accurate predictive software development effort and cost estimation. As there is no clear guideline for designing neural networks approach and also fuzzy approach is hard to use. Genetic Algorithm can offer some significant improvements in accuracy and has the potential to be a valid additional tool for software effort estimation. It is a non-parametric method since it does not make any assumption about the distribution of the data and derives equations according only to the fitted values. Genetic Algorithm is one of the evolutionary methods for the effort estimation. The solution is achieved by means of a cycle of generations of candidate solutions that are pruned by criteria, "survival of the fittest" [10]. Genetic Programming (GP) is a global search technique which makes it less likely to get stuck in the local optimum. This is different from other techniques such as neural networks and gradient descent which is prone to the local optimal values. It is particularly well suited for hard problems where little is known about the underlying search space and the concept is easy to understand [11].

The genetic algorithm is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution. You can apply the genetic algorithm to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear.

The genetic algorithm uses three main types of rules at each step to create the next generation from the current population:

Selection rules select the individuals, called parents that contribute to the population at the next generation.

Crossover rules combine two parents to form children for the next generation.

Mutation rules apply random changes to individual parents to form children.

GA modeling for estimation of cost has been carried out in this work using the MATLAB software [2]. Mat-lab codes were developed for solving multivariable minimization problem using optimization method "ga – Genetic algorithm" solver [2]. Some of the options available within the GA solver are described below in brief

4. Result and Discussion:

After optimization of the fitness function using MATLAB command, the optimized function value and the optimal parameter values are obtained. Using different parameter options for GA algorithm functions solutions obtained are as follows:

$S_{estimated} = S_{observed} = \alpha + b*x(1) + c*x(2) + d*x(3);$ *(1)*

Where

α = 143.654790554716, a constant term

b = 0.802930099303848

c = 0.212133114686023,

d = 0.0811574301282941

The lower and upper bound values used for the parameters are as follows:

lb = [0.75 0.15 0.07];
ub = [0.85 0.25 0.10] ;

Further the figure 1 below shows optimized parameter values of all the datasets using GA optimization.

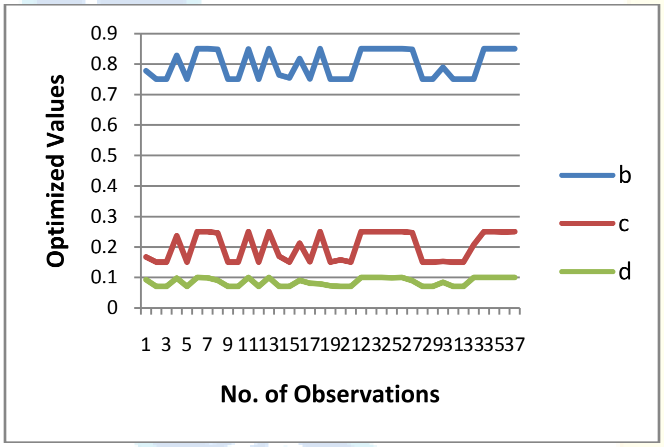

Fig. 1: Plot of optimized parameters "b", "c" and "d" for datasets

On further analysis of the above equations (2) and (3), it was seen that equation (3) for the model was found to be the best developed model, resulting in low RMSE value of 61.66 as compared to that of regression model for the same datasets having RMSE value of 96.31. Also, MAE values for RR and GA based models are 0.17188 and 0.098818 respectively, which again demonstrates the superiority of GA over other techniques (Table 4.5 and Fig. 4.8 & 4.9 below). Further, it clearly demonstrates that genetic algorithm optimization techniques have been successful in developing a better prediction model by lowering the RMSE value. It is shown in figure 4.7 below. The MATLAB plot of the various functions used in the optimization of the model has been shown in figure 4.8 below. Further, the various fitness function values of parameters which are to be optimized have been plotted, as shown in figure below.

Table 1: RMSE and MAE values using RR and GA

	Using Regression	Genetic Algorithm
RMSE	96.31	61.66
MAE	0.17188	0.098818

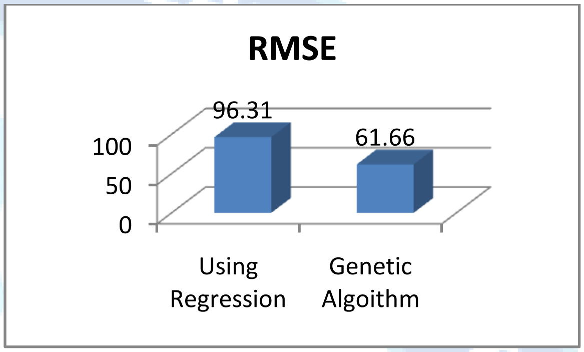

Fig. 2: Plot of RMSE Value using Regression and GA techniques

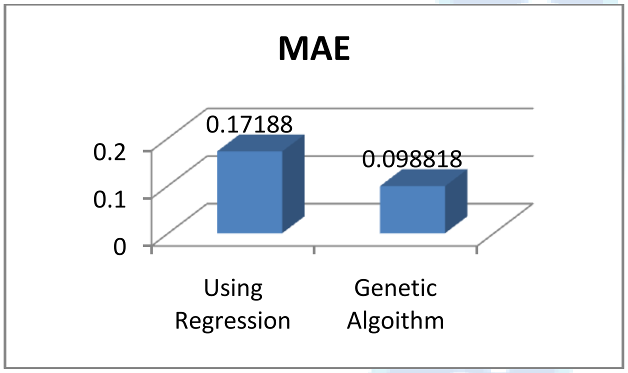

Fig. 3: Plot of MAE Value using Regression and GA techniques

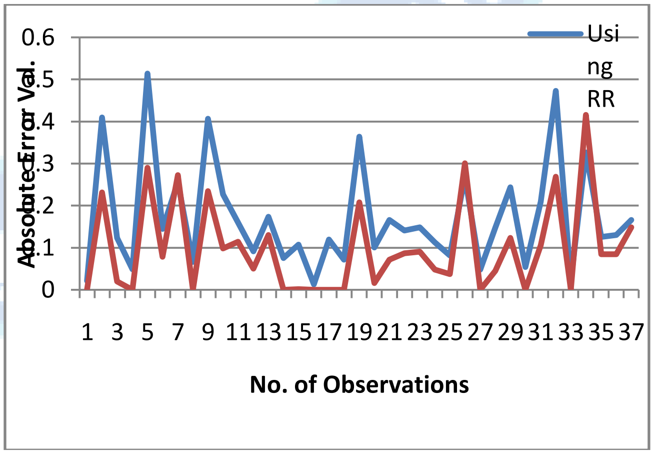

Fig. 4: Plot of Absolute Error Values using Regression and GA techniques for all datasets

5. Conclusion:

Software cost estimation is one of important activity of software development. Software cost estimation plays an important role in software engineering practice, often determining the success or failure of contract negotiation and project execution. Cost estimation's deliverables such as effort, schedule, and staff requirements are valuable information for project formation and execution. The more software becomes important in almost every human activity, the more it becomes complex and difficult to implement. Even if modern software technologies render easier the development of certain types of software products, increased user demands and new application domains produce additional problems. It is not surprising that software project management activities are becoming increasingly important.

References:

[1]. N. Veeranjaneyulu, S.Suresh, Sk.Salamuddin3 and Hye-jin Kim, (2014), " Software Cost Estimation on e-Learning Technique using A Classical Fuzzy Approach", International Journal of Software Engineering and Its Applications Vol. 8, No. 11 (2014), pp. 217-222

[2]. Stein Grimstad, et. al, (2006), "A Framework for the Analysis of Software Cost Estimation Accuracy", ISESE'06, ACM 1-59593-218-6/06/0009

[3]. Lalit V. Patil, et. Al., (2014), " Develop Efficient Technique of Cost Estimation Model for Software Applications", International Journal of Computer Applications (0975 – 8887) Volume 87 – No.16, February 2014

[4]. Jyoti G. Borade, (2013), " Software Project Effort and Cost Estimation Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, pp 730-739.

[5]. K. Subba Rao, et. al. (2013), "Software Cost Estimation in Multilayer Feed forward Network using Random Holdback Method", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 10, pp 1309-1328.

[6]. Vahid Khatibi, Dayang N. A. Jawawi, (2011), " Software Cost Estimation Methods: A Review", Journal of Emerging Trends in Computing and Information Sciences, Volume 2 No. 1, pp 21-29.

[7]. Swati Waghmode1, Dr.Kishor Kolhe, (2014), "A Novel Way of Cost Estimation in Software Project Development Based on Clustering Techniques", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 4, PP. 3892-3899.

[8]. Isa Maleki, Laya Ebrahimi, Saman Jodati, Iraj Ramesh,(2014), "ANALYSIS OF SOFTWARE COST ESTIMATION USING FUZZY LOGIC", International Journal in Foundations of Computer Science & Technology (IJFCST), Vol.4, No.3,, PP. 27-41.

[9]. Hasan Al-Sakran, (2006), "Software Cost Estimation Model Based on Integration of Multi-agent and Case-Based Reasoning", Journal of Computer Science 2 (3): 276-282.

[10]. Soumyabrata Mukherjee, Bishnubrata Bhattacharya, Suvajit Mandal, (2013), " A SURVEY ON METRICS, MODELS & TOOLS OF SOFTWARE COST ESTIMATION", International Journal of Advanced Research in Computer Engineering &Technology (IJARCET) Volume 2, Issue 9,, PP. 2620-2625.

[11]. Geetika Batra, Kuntal Barua, (2013), "A Review on Cost and Effort Estimation Approach for Software Development", International Journal of Engineering and Innovative Technology (IJEIT) Volume 3, Issue 4,, PP. 290-293.

[12]. Narendra Sharma, Ratnesh Litoriya, (2012), " Incorporating Data Mining Techniques on Software Cost Estimation: Validation and Improvement", International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 3,, pp. 301-309.

[13]. Zulkefli Mansor, Saadiah Yahya, Noor Habibah Hj Arshad , (2011), " Review on Traditional and Agile Cost Estimation Success Factor in Software Development Project", International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(3): 942-952.

[14]. Martin Shepperd and Chris Schofield, (1997), "Estimating Software Project Effort Using Analogies", IEEE Transactions On Software Engineering, Vol. 23, NO. 12,, PP. 736-743.