INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 3, Issue 6)
Available online at www.ijariit.com

Predictive and Stochastic Methodology for Software Effort Estimation

V. Narsing Rao
Sphoorthy Engineering College, Hyderabad, Telangana
narsingrao.v@gmail.com

CH. Mahesh Reddy
CMR Engineering College, Hyderabad, Telangana
mahi015@gmail.com

I. INTRODUCTION

Software development is a systematic disciplined approach to the development, maintenance, and retirement of software. The main goal of software engineering is to produce high quality of software with a low cost and within a time and budget. A systematic project management is required to meet this goal. Software Project Management is a set of activities that are collectively called Project Planning. Before the project can begin, the project manager and software team must estimate the work to be done, the resources that will be required, and the time that will elapse from start to finish. If the customer will stay throughout the project, then that will be better for the software team and moreover, it is possible to get more quality and to completion of that project also within time.

The Software Project Management activities are divided into two phases-Project Planning and Project Monitoring and Control. Planning activities must be performed before starting the development work. Once the project is started, project control begins. Planning is the most important management activity and is the weakest activity in many software projects.The basic goal of planning is to identify the activities that need to be done to complete the project successfully and plan the scheduling and resource allocation for these activities.The input to the planning activity is the Requirement Specification Document and the output of this phase is Project Plan describing Cost Estimation, Schedule and Milestones, Personal Plan, Team structure, Software Quality Assurance Plans, Configuration Management Plans, Project Monitoring plans and Risk Management.

Cost estimation is the activity where the overall cost requirement for the project as well as the breakup of the cost for different phases is estimated. Human resources are the most important for software development; cost estimates are often given in terms of person-months (PM) which is also known as Effort. For estimating the cost, cost models are used, that estimate the cost as a function of various parameters. The initial cost estimation is obtained as a function of the coding size (measured in terms of kilo delivered lines of code-KDLOC) of the final product. Once the overall cost estimation is obtained, from this we can estimate project duration estimate and staffing requirement.

The software is the most expensive element in all computer based systems. For complex, custom systems, a large cost estimation error can make the difference between profit and loss. Cost overrun can be disastrous for the developer. To achieve reliable cost and effort estimates, there is a number of alternatives: delay estimation until late in the project, by considering the early projects which are similar, use divide and conquer method and use one or more empirical models for software cost and effort estimation.

The accuracy of the estimate will be depending on the amount of accurate information on the final product. Specification with Uncertainty represents a range of possible final products, and not one precisely defined product. Estimation carries an inherent risk and this risk leads to uncertainty. The accuracy of the actual cost estimates will depend on the effectiveness and the accuracy of the Cost/Effort model employed.

The cost for a project is a function of many parameters, the basic parameter is the size of the project, and other parameters are programmer ability, the experience of the developer's area, the complexity of the project and reliability requirements.

The software cost estimation problem deserves special attention because, the development of any software product is usually a unique undertaking, with the increased size of software projects, any estimation mistakes could cost a lot in terms of resources allocated to the project, and the uncertainty about cost estimation is usually quite high.

II. MOTIVATION

The software cost estimation problem deserves special attention because the development of any software product is usually a unique undertaking, with the increased size of software projects, any estimation mistakes could cost a lot in terms of resources allocated to the project, and the uncertainty about cost estimation is usually quite high. Customers often want a contract for a customer software development to have a fixed price, so they are not exposed to the risk of inaccurate estimates. On the other hand, Software developers want a contract to allow them to change for whatever resources they use; this protects the developers from inaccurate cost estimates but exposes the customers to all the risk. A solution to this conflict is to make a correct estimate project.

"Improving the COCOMO model using a Neuro-Fuzzy approach" which carries some of the desirable features such as learning ability and good interpretability while maintaining the merits of the COCOMO model. There are two major components in their model. The first component is Twenty-Two sub models for each sub model the input is the rating value of a cost driver, and the output is the corresponding multiplier value of a cost driver, which is used as the input of the COCOMO model. The second component is COCOMO model, the input is the size of software and output of sub model and the output is effort estimation. The sub model structure was constructed using Neural Networks; a fuzzy set is defined for each linguistic term of every cost driver. The membership functions were either triangular or other function, or the universe of discourse is the interval. The fuzzy rules framed are functionally equivalent to a Takagi and Sugeno Model. The distinguishable features of neuro fuzzy model are learning ability.

"Software Cost Estimation with Fuzzy models" as a Simple f-COCOMO model, augmented f-COCOMO model. "Optimization criteria for the effort estimation using fuzzy technique", there are two models based on fuzzy logic sizing. Rather than using a single number, the software size is regarded as triangular fuzzy number and extension principle. The fuzzy output is evaluated and then estimation is done by defuzzification technique. They optimize the estimated effort for any applications by varying arbitrary constants for these models.

"Estimation of COCOMO model parameters using Genetic Algorithms for NASA software projects". GA was used to estimate the parameters of a COCOMO type effort estimation model. These solutions are based on the fitness function. For hard optimization problems, it is often recommended to use probabilistic algorithms.

" Development of Software Effort and Schedule Estimation model using Soft Computing Techniques", one model use the Particle swarm Optimization to tune the parameters of the famous Constructive Cost Model (COCOMO). Another model by using Fuzzy logic, to build a set of linear models over the domain of possible software lines of code (LOC).

"Software Effort Estimation Using Particle Swarm Optimization with Inertia Weight", describes that software is the most expensive element of virtually all computer based systems. For complex custom systems, a large effort estimation error can make the difference between profit and loss. Cost (Effort) Overruns can be disastrous for the developer because uncertainty existing in the input information. Accurate software effort estimation is a challenge in Industry. In this paper, they are proposing three software effort estimation models by using soft computing techniques: Particle Swarm Optimization with inertia weight for tuning effort parameters.

Thesis on "Some soft Computing Techniques for Software Development Effort Estimation", outlined that accurate software effort estimation is a challenge in the software Industry. This thesis investigates the efficacy of soft computing methodologies such as Fuzzy logic, Interval Type-2 Fuzzy logic and Particle Swarm Optimization for software effort estimation. In this thesis Interval, Type-2 fuzzy sets are used for modeling uncertainty. Rather than using a single number, the software size when regarded as a fuzzy set yields the cost estimate in the form of a fuzzy set. The PSO with inertia weight and constriction factor can be a powerful tool when tackling the problem of optimized parameters and this thesis examined their suitability for Software Effort Estimation. It also examined the suitability of a Fuzzy PSO, where in the uncertainty in the input sizes are controlled by using fuzzy logic and the PSO with inertia weight for lining the parameters of the cost model to generate an optimal result.

Software Effort Estimation depends on the size of the source code written during the coding phase, which is not known before the development. With the increased size of the software projects, any estimation mistakes could cost a lot in terms of resources allocated to the project. The uncertainty about cost estimation is usually quite high. The development of any software product is usually a unique undertaking.

Software projects estimations can never be an exact science, but a combination of good historical data and systematic techniques can only improve estimation accuracy. With the increased size of software projects, any estimation mistakes could cost in terms of resources allocated to the project. Thus it has become a very important area for researchers experimenting with new methodologies to reduce uncertainties in estimation and this context gave motivation to the present research work. The hybrid method using the techniques-k-means for classification, particle swarm optimization for tuning parameters and Back propagation neural network for training can be tested to determine their feasibility to overcome the uncertainties at the input level, by tuning parameters to produce an optimal solution for software effort estimation. Thus the present work investigates the suitability of some soft computing strategies for effort estimation using multi objective particle swarm optimization, fuzzy logic, and a predictive-stochastic approach.

III. PROBLEM IDENTIFICATION

Studies on references from literature revealed that there is a number of challenges in providing parameter tuning for software effort estimation. Although many researchers, over the years have suggested various approaches to resolve them, still there are requirements for invention and improvements.

Accurate estimation of software development effort is critical in software engineering. Underestimates lead to time pressures that may compromise full functional development and thorough testing of software. In contrast, overestimates can result in noncompetitive contract bids and over the allocation of development resources and personnel. In response to industry demand, a myriad of estimation techniques has been proposed during the last three decades.

In this context, the present project work takes the following approach for effective estimation of development effort with the application of predictive and stochastic strategies. Develop a Predictive and Stochastic approach technique, and examine to apply k-means to bring the linearity among the data items, and optimize parameters tuning by using PSO with inertia weight and then Back Propagation Neural Networks for training and use the approach for software effort estimation.
As the efficacy of the techniques is always crucial in any study and in order to assess the suitability of these proposed techniques in the thesis, from a diverse selection, their performance and relative merits are compared in each chapter, with several existing models in the literature by conducting the experiments on NASA data set and COCOMO datasets.

IV. OBJECTIVES

The main objectives of the present project can be summarized as follows:

1. Develop a Predictive and Stochastic approach technique, and examine to apply k-means to bring the linearity among the data items, and optimize parameters tuning by using PSO with inertia weight and then Back Propagation Neural Networks for training and use the approach for software effort estimation.
2. To conduct experiments on these proposed models by using test data from NASA data set and COCOMO data set.
3. Investigate the feasibility of predictive and stochastic techniques for effort estimation and establish their robustness during uncertain conditions.

V. METHODOLOGY

Predictive and Stochastic approach for Software Effort Estimation

In the present model of Predictive-Stochastic Software Effort Estimation technique for estimating the COCOMO parameter values are implemented. The input data set is clustered using the K-means clustering algorithm in order to bring linearity among the data items and then PSO is applied to each cluster to find the optimized parameter values. Using these clusters and values, the neural network is trained which is then used to classify the testing data into the appropriate cluster. The cluster is further evaluated by PSO for software cost effort estimation.

VI. PLAN OF WORK AND TARGETS TO ACHIEVE

The project is divided into 4 Tasks:
Task-I of the project deals with a brief introduction of the problem understanding, the review of some software cost estimation methods, importance of software cost estimation, basics of swarm intelligence technique- particle swarm optimization, fuzzy logic, multi objective particle swarm optimization, bee colony optimization k-means clustering technique, and some basics of neural networks are to be experimented.

Task-II gives a comprehensive overview of software effort models, Fuzzy Logic Techniques for various applications, swarm Intelligence- Particle Swarm Optimization, multi objective particle swarm optimization Techniques and bee colony optimization for optimization problems.

Task-III of the project is concerned with the development of a Predictive and Stochastic approach for Software Effort Estimation. The behaviour of the K-means clustering for classification, the methodology used in the effort estimation using PSO for fine tuning parameters and classified data and tuned parameters training is done back propagation neural networks are discussed, followed by the model analysis and implementation details. The experimental results of the proposed method are to be presented.

REFERENCES

1. Bailey J.W and Basili V.R (1981), "A meta model for software development resource expenditures", proceedings of the Fifth International Conference on Software Engineering, CH-1627-9/81/0000/0107500.75@IEEE, pp:107-129,1981.
2. Kemerer C.F, (1987) "An Empirical Validation of Software Cost Estimation Models", Management of Computing-Communications of ACM, Vol: 30 No. 5, pp: 416-429, May 1987.
3. Briand L.C, Emam K.E, Frank B, (1997) COBRA: "A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment", International Software Engineering Research Network Technical Report ISERN-97-24 (Revision 2), pp: 1-24, 1997.
4. Hao Ying, (1998), The Takagi-Sugeno fuzzy controllers using the simplified linear control rules are nonlinear variable gain controllers, ELSEVIER: Automatica, Vol: 34, No.2, pp.157-167, 1998.
5. Nilesh N.K, Jerry M.M, and Qilian L (1999), Type-2 Fuzzy Logic Systems, IEEE Transactions On Fuzzy Systems, Vol. 7, No. 6, pp: 643-658 December 1999.
6. Frank C.H.R and Cheul .H (2002), "An Interval Type-2 Fuzzy Perception", 0-7803-7280-8/02@IEEE pp: 1331-1335, 2002.
7. Wu B, Zheng Y, Liu S, and Shi Z (2002), "CSIM: A Document Clustering Algorithm Based on Swarm Intelligence", 0-7803-7282-4/02@EEE pp: 477-482, 2002.
8. SHAO-YUAN L and YI-PENG Y (2003), On-Line Constrained Predictive Control Algorithm using Multi-Objective Fuzzy-Optimization and a Case Study, Fuzzy Optimization and Decision Making, Vol: 2, pp: 123– 142, 2003.
9. Tan K.C, Khor K.F, Lee T.H and Yang Y.H (2003), A Tabu-Based Exploratory Evolutionary Algorithm for Multi objective Optimization, Artificial Intelligence Review Vol: 19, pp: 231–260, 2003.

