ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 9, December 2015

A Software Cost Estimation Approach Using Least Square-Support Vector Machine Technique

V.R.Arulmozhi[1], B.Vijaya Nirmala[2], N.Deepa[3]
[1, 2, 3] Assistant Professor/Department of CSE, RVS Educational Trust's Group of Institution Dindigul, Tamilnadu, India.
[1] arulmozhiram@gmail.com
[2] bvijayanirmalacse@gmail.com
[3] deepanatrayan@gmail.com

Abstract— **In software development, software cost estimation is one of the important process. Real estimation requires cost and effort factor in producing software by using data mining techniques. In the area of software cost estimation various methods have been proposed to estimate the cost of software projects. Constructive Cost Model (COCOMO) is the first and algorithmic cost estimation model which was developed by Boehm. Basic COCOMO is suitable for quick and early estimation of required effort in producing software, but its accuracy is limited due to outliers. Although most of the proposed methods produce point estimates, in practice it is more realistic and useful for a method to provide interval predictions. Data mining techniques improve the accuracy of the estimation models in many cases. In our proposed system, we explore the possibility of using such a method, known as Least Square Support Vector Machine (LS-SVM) to improve the accuracy of software cost estimation.**

Keywords- COCOMO, Data Mining, LS-SVM, Software Cost Estimation.

I. INTRODUCTION

Software cost estimation is the discipline that attempts to foresee the effort required for the completion of a software development project. Typically software cost estimation process identifies the characteristics of a project such as prediction of the size of code and other project artifacts. The outcome of such analysis is an effort estimation value. Based on this value important decision is made by software managers. Software cost estimation is safer to produce category estimation i.e. to predict the project actual cost. By knowing the estimated cost of a particular software project manager can approve or reject a project proposal or to manage the development process more effective.

Furthermore, accurate cost estimates would allow organization to make more realistic bids on external constructs. In recent years, SCE method used for determining requirement effort has been a significant issue. These are applied based on experts experiences. Variety of available methods can be point to the algorithmic models. One of non-linear methods used for estimating SCE is COCOMO 81 Model.

Nowadays, by confecting COCOMO models and using AI techniques, new approach has been created for approximation and calibration new software in great companies and business applications. This problem has direct relationship with success or failure in performed new projects Theses COCOMO method is also known as parametric method because they predict software development effort using a formula of fixed form that is parameterized from historical data. The algorithmic methods require as input attributes such as experience of the software development team, the required reliability of the particular software, the language in which the software is to be written, source line of code (SLOC), complexity and so on which are difficult to obtain during the early stage of a software development life cycle (SDLC). They have also difficulty in modeling the inherent complex relationships .The limitations of algorithmic methods compel us to the exploitation of non-algorithmic methods which are soft computing based. The remaining of this paper is structured as follows: First, section 2 presents a related research of software cost estimation. Then, in section 3 presents about the data mining technique (least square support vector machine). In, section 4 presents about proposed methodology and experimental discussion and results. This paper is concluded by section 5 with conclusion and future research.

II. RELATED RESEARCH

In the field of cost estimation, the cost required to develop a new software project is estimated by taking the details of the new project. That selected project data set is compared with the historical dataset i.e. set of past projects which contains attributes like lines of code, language used and experience of development team. First Zeynab Abbasi Khalifelua [11], compared the different data mining techniques with COCOMO model to estimate software costs and then the results of each technique are evaluated and compared. And showed the comparison of the estimation accuracy of COCOMO model with data mining techniques.

The result shows that the use of Data mining techniques improve the estimation accuracy of the software. Karel Dejaeger, Wouter Verbeke, David Martens, and Bart Baesens [6], proposed that a predictive model is required to be accurate and comprehensible in order to inspire confidence in a business setting. He addresses this issue by reporting on the results of a large scale benchmarking study. He considers different types of techniques, including techniques inducing tree/rule-based models like M5 and CART, linear models such as various types of linear regression, nonlinear models and estimation techniques that do not explicitly induce a model.

The results are subjected to rigorous statistical testing and indicate that ordinary least squares regression in combination with a logarithmic transformation performs best. Ali Bou

Nassif and Luiz Fernando Capretz [1], proposed a Artificial Neural Network (ANN) technique which is used to predict the software effort from use case points (UCP) model. The ANN model was evaluated using the MMER and PRED criteria against the regression model as well as the UCP model that estimates effort from use cases.

Andreas S. Andreou and Efi Papatheocharous [3], developed a technique for software cost estimation using fuzzy decision tree for resource allocation and control. Two algorithms were used namely CHAID and CART are applied on empirical software cost recorded in the ISBSG repository. Stephen Mac Denel [10], has proposed a budgetary constraints are placing increasing pressure on project managers to effectively estimate development effort requirements at the earliest opportunity. Estimates the development effort on the basic of data collected from organization, which captures environmental factors and various difference among the given projects. It is also a constant factor to arrive at final effort.

Jack E. Matson, Bruce E. Barrett, and Josep M. Mellichamp [5], proposed an assessment of several published statistical regression models that relate software development effort to software size measured in function points. The principal concern with published models has to do with the number of observations upon which the models were based and in attention to the assumptions inherent in regression analysis. It also focuses on a problem with the current method for measuring function points that constrains the effective use of function points in regression models and also suggests a modification to the approach that should enhance the accuracy of prediction models based on function points in the future.

Shi-Gan Deng1 and Tsung-Han Yeh2 [9], proposed an estimate of the costs of carbon steel pipe material, steel pipe bending, pressure vessel manufacturing, and pump purchasing. The performance of numerous cost estimation models, including regression analysis, neural networks, and support vector regression, established in the previous articles, are compared with that of the LS-SVM model. The test results verified that the LS-SVM model can provide more accurate estimation performance and outperforms other methods.

S.M. Mousavi, Seyed Hossein Iranmanesh [7], has proposed an approach for managing costs in new product development (NPD) projects is a difficult practice that requires much effort and experience. Least squares support vector machine (LS-SVM) combined with genetic algorithm (GA) are proposed to estimate cost data in these projects. It is proved that the LS-SVM can overcome some shortcoming in the traditional techniques, and the GA is used to tune the LS-SVM parameters automatically.GA enhances the efficiency and the capability of estimation.

Amanjot Singh Klair and Raminder PreetKaur [2], proposed a Software Effort Estimation approach for software development. SVM and KNN are computer-based techniques to estimate effort. KNN and SVM based approach could serve as an economical, automatic tool to generate ranking of software by formulating the relationship based on its training. When there are some data points that belong to one of two classes then the goal is to decide about the class of new data point.

Haifeng Wang DejinHu[4], proposed a Support Vector Machines (SVM) algorithm has been widely used in classification and nonlinear function estimation.However, the major drawback of SVM is its higher computational burden for the constrained optimization programming. This disadvantage has been overcome by LeastSquares Support Vector Machines (LS-SVM), which solves linear equations instead of a quadratic programming problem. Finally he concluded that LS-SVM is preferred especially for large scale problem, because its solution procedure is high efficiency and after pruning both sparseness and performance of LS-SVM are comparable with those of SVM. Estimating software effort is probably the biggest challenge facing software developers. Estimates done at the proposal stage has high degree of inaccuracy, where requirements for the scope are not defined to the lowest details, but as the project progresses and requirements are elaborated, accuracy and confidence on estimate increases. It is important to choose the right software effort estimation technique for the prediction of software effort.

Prabhakar, and Maitreyee Dutta[8], proposed an Artificial Neural Network and Support Vector Machine (SVM) which have been used using China dataset for prediction of software effort. However, Boehm describe new approach to improve SCE accuracy. Pahariya proposed new computational intelligence sequential hybrid architectures involving programming and Group Method of Data Handling (GMDH). This includes data mining methods such as Multi-Layer Regression (MLR), and Radial Basis Function (RBF).

Andreou used Fuzzy Decision Trees (FDTs) for predicting required effort and software size in cost estimation as if strong evidence about those fuzzy transformations of cost drivers contributed to enhancing the prediction process. Reddy et al improved fuzzy approach for software effort of the COCOMO using Gaussian membership function which Loading and preprocessing dataset Training data Testing data Initial Parameter Build LS SVM model Calculate Error Estimating cost Calculate Effort performs better than the trapezoidal function to presenting cost drivers. By means of using these methods the estimated value is not too accurate. So data mining technique where used to estimate the software. To develop an effective cost estimation Least Square Support Vector Machine (LS-SVM) method is used. The quality of cost data directly influences the accuracy of cost estimation model.

III. LEAST SQUARE SUPPORT VECTOR MACHINE

SVM is a nonlinear machine learning technique based on recent advances in statistical learning theory. SVMs have recently become a popular machine learning technique, suited both for regression and classification. A key characteristic of SVM is the mapping of the input space to a higher dimensional feature space. This mapping allows for an easier construction of linear regression functions. LS-SVM for regression is a variant of SVM in which the goal is to find a linear function f(xi) in a higher dimensional feature space minimizing the squared error. The function f(xi)takes the following form: F (xi) = <w,(xi)>+b, with w Rn the weight vector in the input space, a nonlinear function providing the mapping from the input space to a higher (possibly infinite) dimensional feature space, and b R a bias parameter.

The function f(xi) is determined by solving the following convex optimization problem: Minimize W T W + γ ½ Σ ri2 Subject to ei = WT (xi) + b + ri , i=1….N. As can be seen from the objective function, a tradeoff is made between the minimization of the squared error,r2i, and minimizing the dot product of the weight vectors, wTw, by an optimization parameter . The Lagrangian of the problem takes the following form: WT W + Σ ri2-Σαi {wT (xi) + b + ri – ei} Where αi R are the Lagrange multipliers.

The problem is reformulated in its dual form giving way to the following equation: ei=Σαi< (x), (xi) > + b. At this point, the kernel function is applied which will compute the dot product in the higher dimensional feature space by using the original attribute set. In this study, a Radial Basis Function kernel was used since it was previously found to be a good choice in case of LS-SVMs SVM are a popular technique which has been applied in various domains. Since this is a rather recent machine learning technique, its suitability in the domain of software effort estimation has only been studied to a limited extent. System Architecture for Software Cost Estimation.

IV. OVERVIEW OF PROPOSED SYSTEM

A. Preprocessing Dataset:

A dataset is a collection of data; it lists value for each of the variables, such as size of the project. The query used to generate a particular dataset from the selected connection or flat file profile. Dataset pre-processing is the process of deleting, adding, transforming and discrete building variables, which should have discrete value such as actual effort, EM variables and size of the projects. First, from the cocomo data set, the data used to learn and validate the models are selected; only attributes that are known at the moment when the effort is estimated are taken into account (e.g., duration or cost are not known and therefore not included in the data set).

An assumption is made in most software cost estimation studies is that size-related attributes are taken for granted. However, such attributes are often a result of an estimation process on their own. Snap shot for After Preprocessing dataset. Second, since some of the techniques are unable to cope with missing data, an attribute is removed if more than 25% of the attribute values are missing. Since missing values in data set is often occur in the same observations, the number of discarded projects turned out to be low.

B. SVM Model Construction:

The datasets is split into the testing dataset and training dataset. Major pa\rt of data set has been devoted to training data, minor part of it is applied for testing data. We are comparing our trained dataset.

For predicting software costs, SVM applies a linear model to implement non-linear class borders. It maps nonlinear input vectors (consisting of EM and Size of the projects) into a high dimensional attributes space by means of kernels. In this topic, kernel is composed of poly kernel.

Then, the support vectors are applied to invent an optimal linear separating hyper plane (in a case of pattern recognition) or a linear regression function (in the case of regression) in this feature space SVMs can efficiently perform a non-linear classification using what is called the kernel trick implicitly mapping their inputs into high-dimensional feature spaces.

C. Training and testing with Intermediate Dataset:

All the datasets were trained and tested in WEKA data mining tool. Major part i.e., 80% of the data set is created as training data set and minor part i.e., 20% is selected as testing data set. After this process ls-svm method is selected in WEKA tool.

As the result of this all the errors in the datasets were calculated. The comparison between the outputs of dataset applied in the LS-SVM with COCOMO shows that AI method in many cases having efficiency in producing the correct prediction of COCOMO method. At the end of this process magnitude of relative error (MRE) was calculated. MRE = * 100

D. Estimating Software Cost:

The accuracy measures are computed by removing all the errors data. After removing errors effort was calculated, based upon the effort calculated, accuracy of the software cost is estimated.

V. RESULTS AND DISCUSSION

Least Square Support Vector Machine (SVM) machine learning techniques have been used for predicting the software efforts using COCOMO dataset. Some performance indices have been used in order to compare the results obtained from these models.

These indices are Mean-Absolute-Error (MAE), Root-Mean- Square-Error (RMSE), Relative-Absolute-Error (RAE), Relative-Root- Square-Error (RRSE). After calculating this errors all the errors were removed from the datasets. By means of using this values effort is calculated. Effort = A*sizeE* EMi

Where A is the multiplicative constant factor that is related to local organization processes, E is the type of software and EM are project attributes. EMi is an EM made by combining process product and development attributes. From these effort value cost of an overall project where estimated.

VI. CONCLUSION AND FUTURE WORK

Data mining techniques can make a valuable contribution to the set of software effort estimation techniques, but should not replace expert judgment. Instead, both approaches should be seen as complements. Depending on the situation, either expert driven or analytical methods might be preferable as first line estimation. In case the experts possess a significant amount of contextual information not available to an analytical method, expert-driven approaches might be preferred. An automated data mining technique can then be adopted to check for potential subjective biases in the expert estimations. Additionally, various estimation scan be combined in alternative ways to improve the overall accuracy.

REFERENCES

[1] Ali Bou Nassif and Luiz Fernando Capretz, (2012) 'Efficient software Effort using an ANN model based on Use Case Points', Proceedings of 11thIEEE International Conference on Machine Learning and Applications, pp. 42-47.

[2] Amanjot Singh Klair, and Raminder Preet Kaur, (2012) 'Software Effort Estimation using SVM and KNN', International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012) Pattaya (Thailand), pp. 146-147.

[3] Andreas S. Andreou, and Efi Papatheocharous, (2008) 'Software Cost Estimation using Fuzzy Decision Trees', Automated Software Engineering, 23rd IEEE/ACM International conference, pp. 371-374.

[4] Haifeng Wang Dejin Hu. (2005) 'Comparison of SVM and LS-SVM for regression', IEEE International conference on Neural Network and Brain, pp. 279-283.

[5] Jack E. Matson, Bruce E. Barrett, and Josep M. Mellichamp, (1994) 'Software Development Cost Estimation Using Function Points', IEEE Transactions of Software Engineering, Vol.20, Issue 4, pp. 275-287.

[6] Karel Dejaeger, wouter Verbeke, David Martens, Bart Baesens, (2012) 'Data Mining Techniques for Software Effort Estimation: A Comparative Study', IEEE Transactions on software engineering, Vol. 38, No. 2, pp. 375-397.

[7] Mousavi, S.M. and Seyed Hossein Iranmanesh, (2011) 'Least Square Support Vector Machine with genetic algorithm for estimating cost in NPD projects', IEEE 3rd international conference on Communication Software and Networks, pp. 127-131.

[8] Prabhakar, and Maitreyee Dutta, (2013) 'Prediction of Software Effort Using Arificial Neural Network And Support Vector Machine', International Journal of Advanced Research in Computer Science and Software Engineering Vol.3, Issue.3, pp.40-46.

[9] Shi-Gan Deng, and Tsung-Han Yeh, (2011) 'Using Least Squares Support Vector Machines to the Product cost Estimation', Proceedings of the journal of C.C.I.T, pp. 1-16.

[10] Stephen Mac Denel, (1994) 'Comparative Review of Functional Complexity Assessment Methods for Effort Estimation', Software Engineering Journal, pp. 107-116.

[11] Zeynab Abbasi Khalifelua, and FarhadSoleimanianGharehchopogh, (2012) 'Comparison and evaluation of data mining techniques with algorithmic models in software cost estimation', Proceedings of the Elsevier international journal on Expert Systems with Applications, pp. 01-11.