International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:9, No:1, 2015

Evaluation of Model Evaluation Criterion for Software Development Effort Estimation

S. K. Pillai, M. K. Jeyakumar

Abstract**—**Estimation of model parameters is necessary to predict the behavior of a system. Model parameters are estimated using optimization criteria. Most algorithms use historical data to estimate model parameters. The known target values (actual) and the output produced by the model are compared. The differences between the two form the basis to estimate the parameters. In order to compare different models developed using the same data different criteria are used. The data obtained for short scale projects are used here. We consider software effort estimation problem using radial basis function network. The accuracy comparison is made using various existing criteria for one and two predictors. Then, we propose a new criterion based on linear least squares for evaluation and compared the results of one and two predictors. We have considered another data set and evaluated prediction accuracy using the new criterion. The new criterion is easy to comprehend compared to single statistic. Although software effort estimation is considered, this method is applicable for any modeling and prediction.

Keywords**—**Software effort estimation, accuracy, Radial Basis Function, linear least squares.

I. Introduction

MODELING of a system is critical to understand and to predict its behavior. In software development due to intangible nature of software and there is no manufacturing, each software produced is unique. We only make copies of the software which is done in a short time. As the software engineering field is not yet matured like conventional engineering fields there is no established hand book. There is no standards certification for all the software. The problem becomes more complicated as the size measurement is also not universally standardized. In spite of all these problems managers and software engineers have to develop a plan using estimation techniques. Generally Lines of Code (LOC) or Function Point (FP) is used as basic size measure. Methods of varying complexity are proposed for software effort estimation. They are expert based [1], analogy based [2], analytical [3], and machine learning based [4]. Among the machine learning methods, neural networks play a major role in Software Development Effort Estimation (SDEE) [5]. One can design Radial Basis Function network (RBF) by changing only one parameter, function width (spread) which is also known as impact factor [6]. RBF is frequently used for Software Development Effort Estimation and it is shown that RBF performs better [7]–[9]. This motivates the authors to use RBF for estimating small projects. The estimate is essential at early stages of a project to plan manpower, schedule and cost. Underestimates may lead to poor quality and reducing the scope or even may lead to cancellation of the project. This can happen even to fit the project to budget due to management pressure. On the other hand overestimation can lead to underutilization of staff or an organization may lose the project in bidding itself. Both the cases are deterrent to an organization. One has to estimate effort as accurately as possible. Here lies the real problem, the definition of accuracy [10]. A new method of evaluation of accuracy based on linear least squares is proposed. A linear relationship between actual effort and predicted effort for test data is made. We have used mainly the data given in [11] for our studies. The paper is organized as follows: The next section reviews the related work followed by description of radial basis function neural network. Experimental evaluations using the new method are provided in the next section. Conclusions are given at the end followed by references.

S.K.Pillai is with the Electrical and Electronics Engineering Department, Noorul Islam University, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India (Mob: +919840783711; Fax: +914651257266; e-mail: skpillai50@gmail.com).

M.K. Jeyakumar is with the Computer Applications Department, Noorul Islam University, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India (Mob: +919443281133; Fax: +914651257266; e-mail: jeyakumarmk@yahoo.com).

II. Related Studies

SDEE or any prediction (forecasting) accuracy depends on the input data, algorithm used, and criteria used for accuracy computation. Generally historical data is divided into training (verification) set and testing (validation) set. Training data is used to build the model. This model is used for validation using test data. SDEE is a function of input where size of software projects plays an important role. For small projects effort required is also small. Lopez-Martin [11] used fuzzy logic model based on two independent variables New & Changed (N&C) code and Reused (R) code. He has compared the performance of fuzzy model with multiple regression model. The results indicate that there is no difference between these two models. Two fuzzy logic models Mamdani and Takai-Sugeno are studied in [12]. The evaluation of these methods with linear regression showed that Takai-Sugeno fuzzy system performs better. None of these works compares SDEE using one and two independent variables. We have used error characteristics to compare the performance of the two models as explained in [10]. We have followed the guidelines suggested in the literature to conduct statistical tests [13].

Commonly used accuracy evaluation criteria are Mean Magnitude of Relative Error (MMRE), PRED which are defined as below [10], [14].

$$\text{MRE} = \text{abs}(\text{actual} - \text{predicted})/\text{actual} \quad (1)$$

Magnitude of relative error is calculated for each project. This is added for each project and average is calculated.

$$\text{MMRE} = \text{sum}(\text{MREi})/\text{n} \quad (2)$$

$$\text{PRED(l)} = \text{k/n} \quad (3)$$

where k is the number of projects that have a relative error MRE less than l.

If the actual value is 100 and predicted value is 10 then MRE is 90%. On the other hand if the predicted value is 100 and the actual is 10 then MRE is 900%. Although in both cases, the error is 90, MRE favors lower estimate. To avoid this, Mean Magnitude of Error Relative (MMER) is introduced where the denominator is replaced with predicted instead of actual.

$$\text{MER} = \text{abs}(\text{actual} - \text{predicted})/\text{predicted} \quad (4)$$

$$\text{MMER} = \text{sum}(\text{MREi})/\text{n} \quad (5)$$

This statistic favors over estimation. Another reason to support (4) is that the error (actual-predicted) is correlated with actual. To avoid the above two problems it is suggested to use balanced relative error

$$\text{BRE} = \text{abs}(\text{actual} - \text{predicted})/\min(\text{actual}, \text{predicted}) \quad (6)$$

Also mean of the errors or standard deviation is affected by extreme values. The problem with all of these is we are looking for a summary statistic. Instead we have proposed to fit a linear least squares curve between actual and predicted values. Ideally, this equation should have intercept zero and slope one. The major advantage of this is we are comparing with the exact values instead of looking for minimum in MMRE/MMER or maximum of PRED.

III. Measurements

We have used the data given in [11] and [15] for our experimentation. LOPEZ1 data consists of Actual Effort (AE), N&C code (N&C) and Reused code (R) for small projects in an academic setting [11]. Effort in minutes is the dependent variable or response and the two independent variables or predictors are N&C code and R code. For training 163 projects are used and for testing 68 projects are used. Table I summarizes both training (N&C, R, AE) and test data (N&CT, RT, AET). Pearson correlation coefficients of different variables are given in Table II. It can be observed that the linear correlation of R code with Actual Effort is small compared with N&C code correlation. More details of the data are available in [11].

LOPEZ2 data consists of three independent variables, McCabe Complexity (MC), Dhama Coupling (DC), Lines of Code (LOC), and a dependent variable Development Time (DT) in minutes [14]. It has a total of 41 observations. We have randomly selected eight observations for test and the rest for training. As the sample size is not large we have provided summary statistics in Table III for the total data. Correlation coefficients of different variables are given in Table IV. It can be seen that all the correlations are significant.

TABLE I
Characteristics of Lopez1 Data

Variable	Mean	Stddev	Minimum	Median	Maximum
N&C	35.56	26.60	10.00	27.00	137.00
R	41.82	30.86	4.00	34.00	149.00
AE	77.07	37.81	19.00	67.00	195.00
N&CT	44.93	21.28	12.00	41.00	104.00
RT	35.43	23.71	1.00	30.00	100.00
AET	79.16	26.47	11.00	78.00	144.00

TABLE II
Pearson Correlation Coefficients for Lopez1 Data

N&C, R	N&C, AE	R, AE	N&C, RT	N&C, AET	RT, AET
0.114	0.747	-.032	-0.175	0.307	0.190

TABLE III
Characteristics of Lopez2 Data

Variable	Mean	Stddev	Minimum	Median	Maximum
MC	2.707	1.006	1.000	3.000	5.000
DC	0.169	0.058	0.077	0.167	0.333
LOC	13.610	5.563	4.000	13.000	31.000
DT	16.634	3.673	9.000	16.000	25.000

TABLE IV
Pearson Correlation Coefficients for lopez2 Data

MC, DC	MC, LOC	DC, LOC	DT, MC	DT, DC	DT, LOC
-0.386	0.765	-0.435	0.708	-0.705	0.583

IV. Radial Basis Function Neural Network

Neural networks are popular in applications where we are not able to specify the exact relationship between input and output or the relationship is nonlinear. Feed forward neural networks require many parameters to be specified and are iterative in nature. However, RBF networks are iteration free and its output is determined in a straight forward manner when the output layer is linear [6]. Reference [14] concludes that for the software industry RBF network is best suited to effort prediction compared to back propagation neural network. The architecture of RBF is shown in Fig. 1 which consists of input layer, hidden layer and output layer. Hidden layer has h neurons and uses radial basis function

$$(\text{x}) = \exp\left[\frac{\|\text{x}-\text{cj}\|^2}{\sigma\text{j}}\right] \quad (7)$$

cj is the center and σj is the radial distance or spread.

The output is given by

$$\text{F(x)} = \sum_{\text{j}=1}^{\text{h}} \beta_\text{j}\, \varphi(\text{x}) \quad (8)$$

βj is the output layer weights.

The output layer weights are determined using generalized inverse. In our study we have used MATLABR2010a® Neural Network toolbox function.

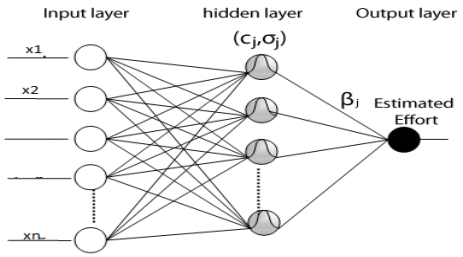

Fig. 1 Radial basis function neural network

V. Experimental Results

A. LOPEZ1Data

Studies were made for LOPEZ1 training data containing 163 observations. In a RBF neural network, the RBF (7) has two constants cj and σj. The center, cj, is selected from the input and the user can only specify, σj, the spread. The spread is varied from 0.1 to 10 for the two input N&C and R and one output effort. RBF performance, mean square error (MSE), 0.0194 is lowest when spread is 1.0 and number of hidden neurons is seven. For the single input N&C minimum MSE, 0.0190, is achieved when spread is 1.0. The trained network is used for evaluating the prediction capability of the RBF network for 68 projects. The box plot of training errors and test (prediction) errors is given in Fig. 2 for both single (RBF1) and two variables (RBF2) cases. Mean, median and inter quartile range (IQR) for the error (actual-predicted) data are given in Table V. It can be observed that the difference between one and two variables is not much. We want to validate this observation using statistical tests. The resulting p-values for t-test and Mann- Whitney nonparametric tests are given in Table VI. We have also given effect size as suggested in the literature [13]. It is clear that statistically there is no significant difference between usages of one or two variables.

We want to fit a linear least squares equation between actual and predicted effort.

$$\text{actual effort} = a * \text{predicted effort} + b \quad (9)$$

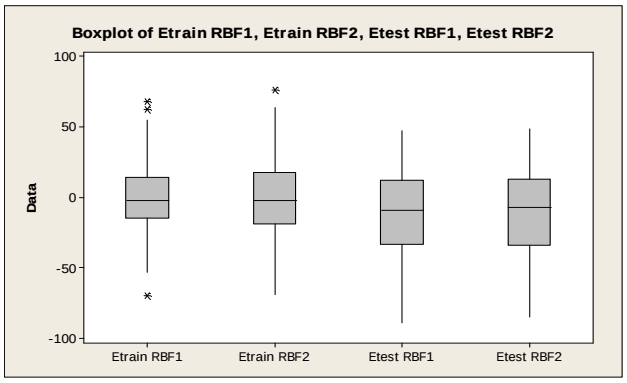

Fig. 2 Box plot of training and test errors for one and two variables for LOPEZ1 data

TABLE V
Training and Test Errors for One and Two Variables LOPEZ1 Data

Variable	One Variable (N&C)		Two Variables (N&C, R)	
	Training	Test	Training	Test
Mean	0.000	-10.620	0.000	-10.36
Median	-2.378	-9.244	.2.223	-6.669
Inter Quartile Range	28.902	45.589	36.446	46.755

TABLE VI
Statistical Tests for One and Two Variables LOPEZ1 Data

	Training	Test
t-test, p values	1.000	0.961
Mann-Whitney test, p values	0.961	0.901
Effect Size	0.507	0.494

TABLE VII
Coefficients for One and Two Variables LOPEZ1 Data

Variable	One Variable (N&C)		Two Variables (N&C, R)	
	Training	Test	Training	Test
Intercept (b)	0.000	48.438	0.000	52.376
Slope (a)	1.000	0.342	1.000	0.299

If the actual effort and predicted effort are equal, the intercept (b) should be zero and slope (a) should be unity. The coefficients obtained for LOPEZ1 data are shown in Table VII. This result indicates that there is some bias in prediction for test data as given by the intercept. RBF estimates well for training data. The one variable test data gives slightly lower intercept and higher slope. This shows that single input is better than two inputs for prediction for LOPEZ1 data set.

B. LOPEZ2 Data

Studies were made for LOPEZ2 training data containing 33 observations. By varying the spread parameter from 0.1 to 1.0 for the three inputs McCabe Complexity, Dhama Coupling and LOC and one output Design time. RBF performance, mean square error 0.01329 is lowest when spread is 0.40 and number of hidden neurons is five. The trained network is used for evaluating the prediction capability for eight projects. The box plot of training errors and test (prediction) errors is given in Fig. 3. Mean, median and inter quartile range (IQR) for the errors are given in Table VIII.

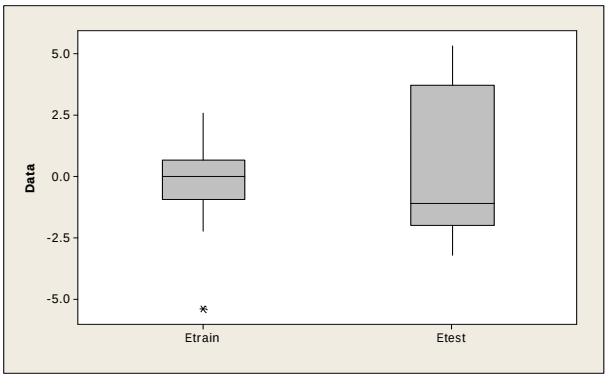

Fig. 3 Box plot of training and test errors for LOPEZ2 data

TABLE VIII
CHARACTERISTICS OF TRAINING AND TEST ERRORS LOPEZ2 DATA

	Training	Test
Mean	0.000	0.27
Median	-0.017	-1.10
Inter Quartile Range	1.592	5.67

We want to fit a linear least squares equation between actual and predicted effort. If the actual effort and predicted effort are equal, the intercept (b) should be zero and slope (a) should be unity. The coefficients obtained for LOPEZ2 data are shown in Table IX. This indicates that the prediction is not as good as training.

TABLE IX
COEFFICIENTS FOR TRAINING AND TEST LOPEZ2 DATA

Coefficients	Training	Test
Intercept (b)	0.000	-3.0942
Slope (a)	1.000	1.2059

VI. CONCLUSION

Based on this study one may choose a single variable N&C for effort prediction of small programs as the statistical tests do not show much difference between the two cases of one and two variables for LOPEZ1 data. The new evaluation criteria of linear least squares curve fitting and checking for intercept and slope also favors one variable for effort estimation. For LOPEZ2 data RBF training is good but the accuracy of prediction is not good. Future studies should aim to reduce the intercept and make the slope of linear least squares fit between actual and predicted effort close to one. The goal of the paper is to demonstrate the use of the new evaluation criterion; we have not tried to compare different models. However, we have used two different data sets. Two major conclusions of the present study are i) The use of linear correlation, as a preprocessing step helps to select independent attributes for effort estimation, ii) The use of linear regression for evaluation of prediction capability of a model. Although, we have used effort estimation problem to demonstrate the new criterion, this method can be used for any model evaluation. This method compares with the expected value of slope one and intercept zero of a straight line compared to other summary statistic looking for a relative value of minimum or maximum.

REFERENCES

[1] M. Jørgensen, "A Review of Studies on Expert Estimation of Software Development Effort," The Journal of Systems and Software, vol. 70, pp. 37–60, 2004.
[2] M.J. Shepperd, C. Schofield, " Estimating Software Project Effort Using Analogies," IEEE Trans. Software Eng., vol. 23, pp. 736-743,1997.
[3] B. Boehm, E. Horowitz, R. Madachy, D. Reifer, B.K. Clark, B.Steece, A.W. Brown, S. Chulani, C. Abts, Software Cost Estimation with COCOMO II, Prentice Hall, 2000.
[4] J. Wen, S. Li, Z. Lin, Y. Hu, C. Huang, "Systematic literature review of machine learning based software development effort estimation models," Information & Software Technology, vol. 54, pp. 41-59, 2012.
[5] V. S Dave, K. Dutta," Neural Network based Models for Software Effort Estimation: A Review," Artificial Intelligence Review, Springer, online 06 May 2012.
[6] Simon Haykin, Neural Networks and Learning Machines, PHI learning Private Limited, New Delhi, 3rd edition, 2010.
[7] A. Adri, A. Zakrani, "Design of Radial Basis Function Neural Networks for Software Effort Estimation," International Journal of Computer Science Issues, vol. 4, pp. 11-17, 2010.
[8] P.V.G.D. Prasad Reddy, K.R. Sudha, P. Rama Sree, S.N.S.V.C. Ramesh, "Software Effort Estimation using Radial Basis and Generalized Regression Neural Networks," Journal of Computing, vol. 2, pp. 87-92, 2010.
[9] V.S. Dave, K. Dutta, "Comparison of Regression model, Feed-forward Neural Network and Radial Basis Neural Network for Software Development Effort estimation," ACM SIGSOFT Software Engineering Notes, vol. 36, *pp*. 1-5, 2011.
[10] B.A.Kitchenham, L.M.Pickard, S.G.MacDonell and M.J.Shepperd, "What accuracy statistics really measure," IEE Proc. Software, vol. 148, pp. 81-85, 2001.
[11] C. Lopez-Martin, "A Fuzzy Logic Model for predicting the Development effort of Short Scale Programs based upon Two Independent Variables," Applied Soft Computing, vol.11, pp.724-732, 2011.
[12] N. Garcia-Diaz, C. Lopez-Martin, A. Chavoya, "A Comparative study of Two Fuzzy Logic Models for Software Development Effort Estimation," Procedia Technology, vol. 7, pp. 305-314, 2013.
[13] A. Arcuri, L. Briand, "A Practical Guide for Using Statistical Tests to Assess Randomized Algorithms in Software Engineering," ICSE'11, Honolulu, USA, May 21-28, 2011, pp. 1-10.
[14] E. Praynlin, P. Latha, "Software Effort Estimation Models Using Radial Basis Function," International Journal of Computer, Information, Systems and Control Engineering, vol. 8, no. 1, World Academy of Science, Engineering and Technology, pp. 248-253, 2014.
[15] C. Lopez-Martin, J. Leboeuf Pasquier, Cornelio Yanez.M, Augustin Gutierrez, T, "Software Development Effort Estimation Using Fuzzy Logic: A case study," Proceedings of the sixth International Conference on Computer Science, (ENC'05), IEEE Computer Society, 26-30 Sep. 2005, pp.113-120.

S. K. Pillai received B.E. in Electrical Engineering in 1971 from Madurai University, Madurai, Tamilnadu, India. He obtained M.Tech. from IIT Madras, Chennai, Tamilnadu, India, in 1973. After his masters, he joined Indian Space Research Organization and worked for 22 years. Then he joined NeST, Trivandrum, India as President and worked for eight years. Afterwards, he was Vice-President at HCL Technologies for six years. He was a Six Sigma Black Belt from American Society for quality. He holds Six Sigma Black Belt from Quality Assurance Institute and Six Sigma Master Black Belt from Indian Statistical Institute. Currently, he is working as Professor in the Department of Electrical and Electronics Engineering. Concurrently, he is also perusing his Ph.D. in Computer Science and Engineering. He has published more than 30 papers in peer reviewed journals and conferences. He is a senior member of IEEE and member of ACM. He is also a life member of Computer Society of India and National Institution for Quality and Reliability. He is Fellow of Institution of Engineers, India. His interests include soft computing and software engineering,

Dr. M. K. Jeyakumar received his Post Graduation Degree in Master of Computer Applications from Bharathidasan University, Trichirappalli, Tamilnadu, India in 1993. He fetched his M.Tech degree in Computer Science and Engineering from Manonmaniam Sundarnar University, Tirunelveli, Tamilnadu, India in 2005. He completed his Ph.D. degree in Computer Applications from Dr.M.G.R Educational and Research Institute University, Chennai, Tamilnadu, India in 2010. He is at present working as Professor in the Department of Computer Applications and Additional Controller of Examinations, Noorul Islam Centre for Higher Education, Kumaracoil, Tamilnadu, India. He has twenty years of teaching experience in this reputed institution. He has published thirty six research papers in International and National Journals. He has also presented more than twenty research papers in International and National Conferences conducted by esteemed organizations. His research interests are Mobile Computing and Network Security.