Chapter 107
General Regression Neural Network for Software Effort Estimation of Small Programs Using a Single Variable

S.K. Pillai and M.K. Jeyakumar

Abstract Software development effort estimation always remains a challenging task for project managers. New techniques are applied to estimate effort. Predicting effort for small programs in educational setting is a difficult task. Minimum number of independent variables should be used to reduce data collection effort. Evaluation of accuracy is a major activity as many methods are proposed in the literature. Here, we have applied General Regression Neural Network (GRN) and compared the results with Linear Least Squares Regression (LSR) for one and two independent variables. Results are evaluated using statistical tests and effect size. The results show that accuracy of GRN and LSR with one and two variables are not different for small programs.

Keywords Software development effort estimation · Least squares regression · Statistical tests · Effect size · Neural network

107.1 Introduction

One of the major activities in software project management is software development effort estimation (SDEE). Recently machine learning methods and data mining techniques are getting more attention [1, 2]. Among the model based techniques regression is most frequently used. In comparing different methods regression is used as a default method. Problems of comparing one method with another arise as there

S.K. Pillai (✉)
Electrical and Electronics Department, Noorul Islam University,
Kumaracoil, Tamil Nadu, India
e-mail: skpillai50@gmail.com

M.K. Jeyakumar
Computer Applications Department, Noorul Islam University,
Kumaracoil, Tamil Nadu, India
e-mail: jeyakumarmk@yahoo.com

C. Kamalakannan et al. (eds.), *Power Electronics and Renewable Energy Systems*,
Lecture Notes in Electrical Engineering 326, DOI 10.1007/978-81-322-2119-7_107

are many criteria for accuracy evaluation. Accuracy also depends on the data used for evaluation as well as the criteria. Generally one can classify SDEE into four groups:

(i) Analogy based methods
(ii) Expert estimation, Delphi and Wideband Delphi
(iii) Model based such as COCOMO, SLIM, etc.
(iv) Artificial Intelligence (AI) methods such as neural networks, fuzzy logic, genetic algorithms or combinations there of

Past projects data are used directly or indirectly in all the methods. Analogy based methods compare the current project with past project which is close to it. In expert estimation, opinion of experts is sought for effort values. In model based methods, relationship between effort and project parameters are obtained using historical data. Among the AI methods neural networks are most commonly used [3]. Here we have used General Regression Neural Network (GRN) which is easy to implement with only one parameter to be tuned and its performance compared with classical Linear Least Squares Regression (LSR).

Section 107.2 gives related work followed by estimation problem and measurement data in Sect. 107.3. LSR and GRN are explained in Sects. 107.4 and 107.5 respectively. Comparative results are provided in Sect. 107.6. Section 107.7 provides conclusions and future research. References are listed at the end. We have followed the empirical software engineering approaches given in [4, 5].

107.2 Related Work

Software development effort estimation continues to be a hot topic in spite of many persons across many countries doing research. The major problems are related to input data, algorithm, and accuracy evaluation criteria. One needs to consider all these three factors to arrive at a conclusion. Boehm et al. [6] suggest that no one technique should be relied upon for SDEE. Instead multiple methods should be compared for decision making. Also when students learn estimation and apply in a college setting it becomes easy for any organization to implement SDEE. As discussed earlier, SDEE is a function of input where size of software projects play an important role. For small projects effort required is also small. Lopez-Martin [7] used fuzzy logic model based on two independent variables New and 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 [8]. The evaluation of these methods with linear regression showed that Takai-Sugeno fuzzy system performs better. Here only New and Changed code is used as independent variable. GRN is used to predict effort of industrial projects [9]. It is proved using statistical tests ANOVA and Kruskal-Wallis that GRN is an alternative to regression model. None of the above works compares SDEE using one and two independent variables. It is suggested to report effect size in statistical testing of all randomized algorithms [5].

107.3 Estimation Problem and Measurements

SDEE generally consists of two stages viz. model building and model evaluation. These are also known as verification or training and validation or testing. A part of the measurements is used to build the model and the remaining data are used to validate the model. Here we have used the data for verification and validation given in [7]. This consists of Actual Effort (AE), N&C code (N&C) and Reused code (R) for small projects in an academic setting. Effort 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 107.1 summarizes both training and testing data (N&CT, RT, AET). Pearson correlation coefficients of different variables are given in Table 107.2. It can be observed that the linear correlation of Reused code with Actual Effort is small compared with New and Changed code correlation. More details of the data are available in [7].

The estimation problem aims at finding a relationship between dependent and independent variables using training data. Then the test data is used to validate the developed model. We have used General Regression Neural Network (GRN) [10] and compared the results with Linear Least Squares Regression (LSR) for one and two independent variables. The accuracy is evaluated by magnitude of error relative to prediction (MER) with respect to each project and for each model. It has been strongly suggested not to use magnitude of error with respect to prediction (MRE) [11]. Also the error, (Actual effort$_i$ − Predicted effort$_i$), known as residual in statistics literature is uncorrelated to Predicted effort. For each project:

$$\text{MER}_i = \text{abs}(\text{Actual effort}_i - \text{Predicted effort}_i)/\text{Predicted effort}_i$$

The aggregate for all the n projects is: MMER = (1/n) Σ MER$_i$.

Table 107.1 Characteristics of training and testing data

Variable	Mean	Stdev	Minimum	Median	Maximum	Skewness	Kurtosis
N&C	35.56	26.60	10.00	27.00	137.00	1.67	2.45
R	41.82	30.86	4.00	34.00	149.00	1.41	1.88
AE	77.07	37.81	19.00	67.00	195.00	0.85	0.04
N&CT	44.93	21.28	12.00	41.00	104.00	0.54	−0.10
RT	35.43	23.71	1.00	30.00	100.00	0.95	0.29
AET	79.16	26.47	11.00	78.00	144.00	0.22	−0.14

Table 107.2 Pearson correlation coefficients of different variables

N&C versus R	N&C versus AE	R versus AE	N&CT versus RT	N&CT versus AET	RT versus AET
0.114	0.747	−0.032	−0.175	0.307	0.190

107.4 Linear Least Squares Regression (LSR)

107.4.1 Using Two Independent Variables (N&C, R)

We have used MINITAB® to obtain the following results. The least squares method fits training data in the two variables as AE = 44.7 + 1.08 N&C − 0.146 R.

The contribution of Reused code is one tenth of New and Changed code. The coefficient signs are intuitively correct.

$$\text{R - Sq} = 57.2\ \% \quad \text{R - Sq(adj)} = 56.6\ \% \quad \text{R - Sq(pred)} = 55.43\ \%$$

The R-Sq value indicates that the predictors explain 57.2 % of the variance in Actual Effort. The R-Sq(adj) is 56.6 %, which accounts for the number of predictors in the model. The R-Sq(pred) value is 55.43 %. Because the predicted R value is close to the R-Sq and adjusted R-Sq(adj) values, the model does not appear to be over fit.

The *P*-value, 0.000, in the Analysis of Variance table (Table 107.3) shows that the model estimated by the regression procedure is significant at an α-level of 0.05. This indicates that at least one coefficient is different from zero.

The *P*-values for the estimated coefficients of N&C and R are both less than 0.05, indicating that they are significantly related to AE. The residuals plots for model validation are shown in Fig. 107.1. The normal probability plot shows an

Table 107.3 Analysis of variance for two variables

Source	DF	SS	MS	F	P
Regression	2	132,382	66,191	106.74	0.000
Residual error	160	99,214	620		
Total	162	231,596			

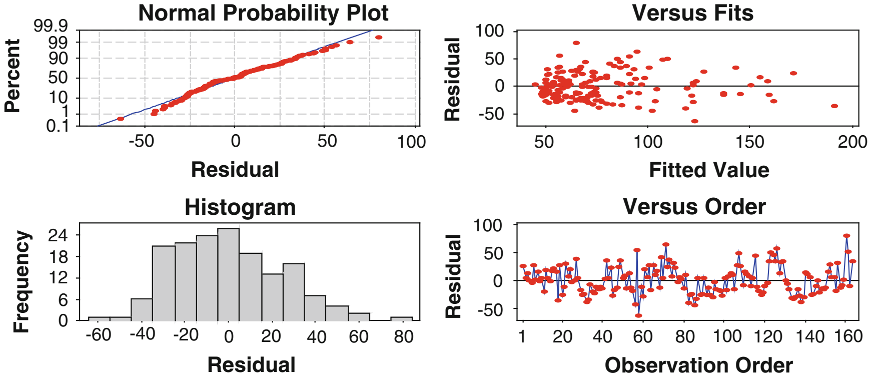

Fig. 107.1 Residual plots for training data for two variables

Table 107.4 Coefficients table for two variables

Predictor	Coef	SE Coef	T	P
Constant	44.723	4.032	11.09	0.000
N&C	1.08075	0.07403	14.60	0.000
R	−0.14568	0.06382	−2.28	0.024

approximately linear pattern consistent with normal distribution. The plot of residuals versus the fitted values shows that the residuals are distributed on both sides of the reference line. The graphs do not show any abnormality. From both graphical and tabular data analysis, we can accept the regression equation for two variables. The accuracy in term of MMER for training and testing are 0.274 and 0.287 respectively.

107.4.2 *Using One Independent Variable (N&C)*

Since the correlation of Reused code with AE is small we have used only N&C. The regression equation is

$$\text{AE} = 39.3 + 1.06\ \text{N\&C}$$

$$\text{R - Sq} = 55.8\,\% \quad \text{R - Sq(adj)} = 55.5\,\% \quad \text{R - Sq(pred)} = 54.63\,\%$$

These values are not much different from two variables results. Both ANOVA and coefficient table (not shown) (Table 107.4) indicate the significance of regression and coefficient. Residuals plots also do not show any problem. One interesting observation no residual is more than three sigma value where as for two variables case one observation (160) is outside three sigma. The accuracy in term of MMER for training and testing are 0.272 and 0.276 respectively.

107.5 General Regression Neural Network (GRN)

The GRN is a type of neural network which can be used to perform regression on a continuous data [10]. It can learn fast compared to standard back propagation multilayer perceptron as the output is obtained using a single pass. Also GRN needs only one parameter, spread, to be tuned. This network can be used for any regression problem including assumption of no linearity. We have used MATLAB® for GRN application. Figure 107.2 gives the architecture of GRN. The network contains one radial basis layer (Hidden) and one linear layer (Output). Radial basis layer activation function is exponential and its spread needs to be adjusted empirically for a particular problem. We need to find two optimal spreads one for two variable case and another for one variable input.

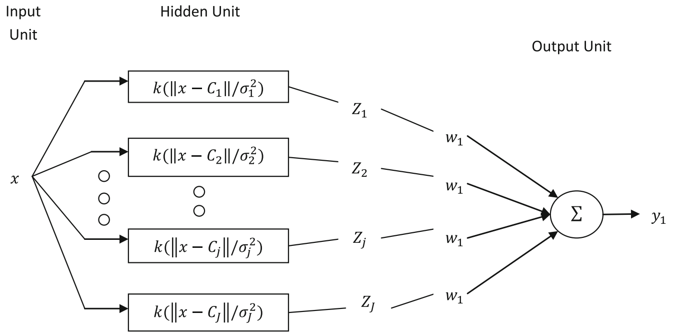
Input Unit
Hidden Unit
Output Unit
x
$k(\|x - C_1\|/\sigma_1^2)$
$k(\|x - C_2\|/\sigma_2^2)$
$k(\|x - C_j\|/\sigma_j^2)$
$k(\|x - C_J\|/\sigma_J^2)$
Z_1
Z_2
Z_j
Z_J
w_1
w_1
w_1
w_1
Σ
y_1

Fig. 107.2 Generalized regression neural network architecture

The spread is varied in steps of 0.01 from 0.01 to 1.0. The input and output are scaled from 0 to 1. For two variables, N&C and R, input it is found that at spread equal to 0.22, training and testing MMER are nearly equal. MMER for training is 0.2976 and for testing is 0.2986. For one variable case, N&C, input it is found that at spread equal to 0.08, training and testing MMER are nearly equal. MMER for training is 0.2725 and for testing is 0.2733. Although the accuracy is better (reduced error) for one variable, we need to validate statistically.

107.6 Comparative Analysis

We have used MINITAB® for testing the equality of two means and two medians. Two sample t test is used for the former as it is robust to moderate departures from normality assumptions. A non-parametric test, which does not require any distribution assumption, Mann-Whitney (M-W) test, is applied for comparing medians. As suggested in [5], we give *P* values for both tests. It is also required to give effect size as it is possible to obtain statistically significant results for large samples with *t*-test and M-W test. Here we report non-parametric effect size measure A_{12} given in [5].

$A_{12} = \{R1/m - (m + 1)/2\}/n$, where R1 is the rank sum of the first data group, m is the number of observations in first data sample, n is the number of observations in second data sample. Given a performance measure, M, the A_{12} statistics measures the probability that running algorithm one yields higher M values than running second algorithm. Table 107.5 provides comparison of mean, median and standard deviation of MER for LSR and GRN for one variable and two variables for testing and training data. The differences are small and we want to confirm by statistical tests. Results, *P* values, for different statistical tests for LSR and GRN are given in Table 107.6. It can be seen from t test and M-W test that for all cases *P* values are more than 0.05. We can conclude that GRN performs equal to LSR for one or two variables and for training and testing. Effect size for all the four cases is near 0.5, we

Table 107.5 Comparison of statistical parameters for LSR and GRN

	LSR				GRN			
	One variable (N&C)		Two variables (N&C, R)		One variable (N&C)		Two variables (N&C, R)	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Mean	0.272	0.277	0.274	0.287	0.272	0.273	0.298	0.299
Median	0.251	0.271	0.249	0.287	0.262	0.269	0.288	0.279
Stdev	0.193	0.184	0.194	0.186	0.192	0.190	0.217	0.182

Table 107.6 Comparison of statistical tests for LSR and GRN

	One variable (N&C)		Two variables (N&C, R)	
	Training	Testing	Training	Testing
t-test, *P* values	0.965	0.921	0.312	0.715
Mann-Whitney test, *P*	0.923	0.833	0.513	0.726
Effect size	0.497	0.511	0.479	0.482

Table 107.7 Comparison of statistical tests for one (N&C) and two variables (N&C, R)

	LSR		GRN	
	Training	Testing	Training	Testing
t-test, *P* values	0.891	0.743	0.269	0.431
Mann-Whitney test, *P*	0.835	0.729	0.410	0.418
Effect size	0.507	0.517	0.526	0.540

can conclude that LSR and GRN perform equally well. Table 107.7 compares the performance of one and two variables. As expected *P* values for *t*-test and M-W test are greater than 0.05. Also effect size values are close to 0.5. We can conclude that there is no performance difference between one or two variables for LSR or GRN.

107.7 Conclusions

Software effort estimation accuracy of GRN is equal to LSR for small projects for the MMER criteria. Software effort estimation accuracy using one variable is equal to two variables for small projects for the MMER criteria for both algorithms. It is recommended to use one variable, N&C, for estimation of small projects. Research needs to be carried out to theoretically justify the same performance of GRN and LSR.

References

1. Dejaeger K, Verbeke W, Martens D, Baesens B (2012) Data mining techniques for software effort estimation: a comparative study. IEEE Trans Softw Eng 38:375–397
2. Wen J, Li S, Lin Z, Hu Y, Huang C (2012) Systematic literature review of machine learning based software development effort estimation models. Inf Softw Technol 54:41–59
3. Dave VS, Dutta K (2012) Neural network based models for software effort estimation: a review. In: Artificial intelligence review, Springer, Berlin (online)
4. Kitchenham BA, Pfleeger SL, Pickard LM, Jones PW, Hoaglin DC, Emam KE, Rosenberg J (2007) Preliminary guidelines for empirical research in software engineering. IEEE Trans Softw Eng 28:721–733

5. Arcuri A, Briand L (2011) A practical guide for using statistical tests to assess randomized algorithms in software engineering. ICSE'11, Honolulu, USA, 21–28 May 2011
6. Boehm B, Abts C, Chulani S (2000) Software development cost estimation approaches: a survey. Ann Soft Eng 10:177–205
7. Lopez-Martin C (2011) A fuzzy logic model for predicting the development effort of short scale programs based upon two independent variables. Appl Soft Comput 11:724–732
8. Garcia-Diaz N, Lopez-Martin C, Chavoya A (2013) A comparative study of two fuzzy logic models for software development effort estimation. Procedia Technol 7:305–314
9. Lopez-Martin C, Isaza C, Chavoya A (2012) Software development effort prediction of industrial projects applying a general regression neural network. Empirical Softw Eng 17:738–756
10. Specht DF (1991) A general regression neural network. IEEE Trans Neural Netw 2:568–576
11. Foss T, Stensrud E, Kitchenham B, Myrtveit I (2003) A simulation study of the model evaluation criterion MMRE. IEEE Trans Softw Eng 29:985–994