A New Regression Based Software Cost Estimation Model Using Power Values

Oktay Adalier[1], Aybars Uğur[2], Serdar Korukoğlu[2], Kadir Ertaş[3]

[1]TUBITAK-UEKAE, National Research Institute of Electronics and Cryptology,
PK 74 Gebze KOCAELI - TURKEY,
{oadalier}@uekae.tubitak.gov.tr

[2]Ege University, Department of Computer Engineering, Bornova IZMIR – TURKEY,
{aybars.ugur, serdar.korukoglu}@ege.edu.tr

[3]Dokuz Eylül University, Department of Econometrics,
Buca, IZMIR – TURKEY,
{kadir.ertas}@deu.edu.tr

Abstract. The paper aims to provide for the improvement of software estimation research through a new regression model. The study design of the paper is organized as follows. Evaluation of estimation methods based on historical data sets requires that these data sets be representative for current or future projects. For that reason the data set for software cost estimation model the International Software Benchmarking Standards Group (ISBSG) data set Release 9 is used. The data set records true project values in the real world, and can be used to extract information to predict new projects cost in terms of effort. As estimation method regression models are used. The main contribution of this study is the new cost production function that is used to obtain software cost estimation. The new proposed cost estimation function performance is compared with related work in the literature. In the study same calibration on the production function is made in order to obtain maximum performance. There is some important discussion on how the results can be improved and how they can be applied to other estimation models and datasets.

Keywords: Software Cost Estimation, Regression Analysis, Software Cost Models

1 Introduction

Software development effort estimates are the basis for project bidding, budgeting and planning. These are critical practices in the software industry, because poor budgeting and planning often has dramatic consequences [1]. The common argument on the project cost overruns is very large. Boraso reported that [2] 60% of large projects significantly overrun their estimates and 15% of the software projects are never completed due to the gross misestimating of development effort.

Delivering a software product on time, within budget, and to an agreed level of quality is a critical concern for software organizations. Accurate estimates are crucial for better planning, monitoring and control [3]. Jones [4] proposes software quality as "software quality means being on time, within budget, and meeting user needs". On the other hand, it is necessary to give the customer or the developer organization an early indication of the project costs.

As a consequence, considerable research attention is now directed at gaining a better understanding of the software development process as well as constructing and evaluating software cost estimation tools. Therefore, there has been excessive focus of research on estimation methods from a variety of fields.

Statistical regression analysis is the most suitable technique to calibrate performance models in a black-box fashion. Statistical regression analysis is a technique which models the relation between a set of input variables, and one or more output variables, which are considered somewhat dependent on the inputs, on the basis of a finite set of input/output observations. The estimated model is essentially a predictor, which, once fed with a particular value of the input variables, returns a prediction of the value of the output. The goal is to obtain a reliable generalisation, that means that the predictor, calibrated on the basis of a finite set of observed measures, is able to return an accurate prediction of the dependent variable when a previously unseen value of the independent vector is presented. In other terms, this technique aims to discover and to assess, on the basis of observations only, potential correlations between sets of variables and use these correlations to extrapolate to new scenarios [5].

The paper is structured according to the Jorgensen's [6] software development estimation study classification. That is the research topic of the study is cost production function and its calibration. Estimation approach of the work is regression analysis and expert judgment. Research approach covers real-life evaluation and history-based evaluation data. The data set which is used in this study are obtained from the International Software Benchmarking Standards Group (ISBSG) data set Release 9. The data set records true project values in the real world, and can be used to extract information to predict new projects cost in terms of effort.

The rest of the paper is structured as follows: Section 2 starts with a discussion on related work. The evaluation criteria are presented in Section 3. Section 4 follows with the description of the data set and the data preparation. Section 5 explains the proposed estimation methods applied. Section 6 summarizes and discusses the results of the analysis. Finally, Section 7 presents the conclusions and discussion of practical implications.

2 Related Work

Regression-based estimation approaches dominate. Notice that regression-based estimation approaches include most common parametric estimation models. More than half of the estimation papers try to improve or compare with regression model based estimation methods. Statistical regression analysis is the most suitable technique to calibrate performance models in a black- box fashion [5]. The problem of cost modeling can be attacked within the framework of statistical regression. The idea is that statistical techniques can discover complex relations between input independent variables and output dependent variables. The estimated model is essentially a predictor, which once fed with a particular value of the input variables, returns a prediction of the value of the output. In other terms, this technique aims to discover and to assess, on the basis of observations, potential correlations between sets of variables and use these correlations to extrapolate to new scenarios.

In our case, we are interested in building a regression model based on the training data to use it subsequently to predict the total effort in man-months of future software projects.

Wieczorek [3] make four different studies with different data sets. Especially her third and fourth study with ISBSG dataset is valuable for our study. It covers 145 and 324 projects respectively. In her study system size (function points) was identified as the most important cost driver by all methods and for all datasets. Her evaluation criteria was the magnitude of relative error and as prediction level Pred(0.25) is used. He used in her study ordinary least squares regression and stepwise analysis of variance. She found that the descriptive statistics for the variables Organization type, System size in terms of function point and productivity.

Liu [7] used ISBSG data set in her study. She tests the correlation between the explanatory numerical variables and the response variable by running a stepwise regression analysis. At the end of her study she obtains adjusted R^2 –value 0.6275. Therefore her model represents whole ISBSG dataset not more than 65%. Stensrud [8] uses a dataset with 48 completed projects extracted from an internal database in Andersen Consulting. He calculates the adjusted R^2 –value by 80.1%. Hu [9] reviews software cost models. He presents the Minimum Software Cost Model (MSCM), derived from economic production theory and system optimization. The MSCM model is compared with COCOMO, SLIM estimation models. As known COCOMO is an empirical model derived from statistical regression. SLIM has its roots in the Rayleigh manpower distribution. For comparative purpose, two classic production models, the generalized Cobb-Douglas production (GCD) and the generalized Cobb-Douglas production with time factor (GCDT) are also included. He note that MSCM model also takes the general form of the Cobb-Douglas production function except that it requires α + β = 1. They used the magnitude of error (MRE) to measure the quality of estimation of each model. They obtain in their study the adjusted R^2 –value for MSCM 89%, for GCDT 56% and for GCD 54% respectively.

3 Evaluation Criteria

The validation of a model is also a persistent issue in the construction of software cost functions. Depending on the authors, different measures of the goodness of fit of the model have been considered. The coefficient of multiple determinations R^2, and adjusted R^2, obtained from regression analysis is being used as a measure of the capability of explanation of the model. Analysis of the residuals seems a necessary condition for examining the aptness of the model, and outlier examination has been suggested for examining the model stability. There is a wide consensus in using the magnitude of relative error (MRE) as an essential element for assessing a regression model [10].

The magnitude of relative error as a percentage of the actual effort for a project is defined as:

$$MRE = \left| \frac{Effort_{Actual} - Effort_{Estimated}}{Effort_{Actual}} \right| \qquad (1)$$

The MRE is calculated for each project in the data sets.

In addition the prediction level Pred(r), prediction at level r, is also used. This measure is often used in the literature and is a proportion of observations for a given level of accuracy:

$$\Pr ed(r) = k / N \qquad (2)$$

where, N is the total number of observations, and k the number of observations with an MRE less than or equal to r. Wieczorek [3] and Boraso [2] report that according to Conte, a value of Pred(0.25) ≥75% and $\overline{MRE}$ ≤25% are considered desirable for effort models. It seen that the accuracy of an estimation technique is proportional to the Pred(0.25) and inversely proportional to the MRE and the mean MRE, $\overline{MRE}$.

4 Data Set Description

Jorgensen state that the potential importance of researchers with a long-term focus on software cost estimation can be illustrated through an analysis of the papers covering the topics “measures of estimation performance” and “data set properties.” He concludes these topics are basic research on cost estimation necessary for meaningful analyzes and evaluations of estimation methods [6].

He states that most of the publications evaluate an estimation method by applying historical data. Therefore, he believes that the lack of publications on real-life use of estimation methods point at a potentially important shortcoming. According to the author, if the estimation context were made with real-life professional project dataset then the realism of the estimation study will increase.

The study in this paper is based on the International Software Benchmarking Standards Group (ISBSG, Release 9) data [11]. It contains data for 3.024 projects all around the world. It should be noted that the data in the Repository has come from twenty countries, with 70% of the projects being less than six years old. This is what makes the ISBSG Repository unique. A broad range of project types from many industries and many business areas are available for our study to use for estimating, awareness of trends, comparison of platforms and languages or benchmarking. 57% are enhancement projects, 41% are new developments, and 2% are re-developments. The number of projects according to their development type is shown in Table 1.

Table 1. ISBSG R9 data set number according to four development types is given.

Type of development	Projects
Enhancement	1711
New development	1246
Re-development	65
Other	2
Total	3024

The ISBSG project has developed and refined its data collection standard over ten year period based on the metrics that have proven to be most useful in helping to improve software development management and process. The ISBSG R9 data set records true project values in the real world, and can be used to extract information to predict new projects cost in terms of effort. In the data set the reliability of samples are rated as 4 levels from A to D where level 'A' represents the most reliable data. In ISBSG R9) data set, there are 734 'A' rated observations, each containing 50 explanatory variables (project size, time, programming language, etc.) and 2 response variables (Summary work effort and normalized work effort). After removing the observations that contain missing attribute values, from level 'A' data set, there are 112 project records remaining.

We choose normalized work effort as response variable or dependent variable. Each observation contains 4 variables 3 of which are the independent variables such as source lines of code (SLOC), adjusted function points, normalized productivity delivery rate. The primary motivation for choosing adjusted function points was the need to determine an estimate of system size and hence development effort early in the development lifecycle the same need is stated also by Finnie [12]. The SLOC is the number of the source lines of code produced by the project. Since this is only available for some projects in the data set. For this reason only 112 of the projects are chosen from the data set.

It is not a surprised that none of the chosen explanatory variables are normally distributed. Figure 1 shows the distribution of raw data of explanatory variables.

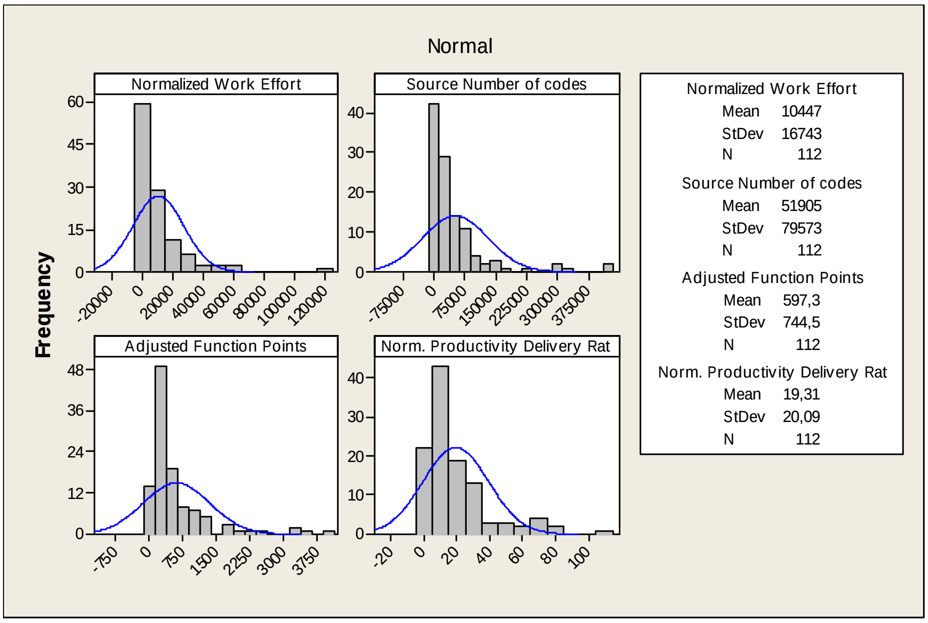

Figure 1. Histogram of original raw data of explanatory variables.

These variables are transformed to approximate a normal distribution. The natural logarithm to base e (natural logarithms) is applied to the data set. The correlation between explanatory variables is not high.

5 Proposed Model

Dolado [10] stated that the consumption of resources can be seen from two complementary viewpoints: as a cost function or as a production function. The effort is the input to the process and the output is the software product. This problem is the dual one of finding a cost function for the software product, although from the methodological point of view the issues are similar in both cases. Almost all works in the literature discuss the problem of project estimation from the perspective of the identification of the cost function, in which the problem under study is to identify the function which relates the size of the product to the man months employed. This is the approach followed here. The underlying idea for building the model is to consider that the software product is manufactured in some way, and the factors that affect the cost of the project is the size of the software which is given as source number of codes, adjusted function points and normalized productivity delivery rate. The equation is calibrated to fit to the actual data points of the projects which are given in the dataset. Since all the p-values are smaller then 0.01, there is sufficient evidence at alpha = 0.01 that the predictor variables are correlated with the response variable. As a result we can use the predictor variables source number of codes, adjusted function points and normalized productivity delivery rate to predict the effort of the project.

The use of the above stated predictor variables is supported by Delapy [13]. Which claims that one simple way of calculating the effect involved in the development of a software system requires a measure of size of the system (for example in lines of code or function points) and the expected productivity of the project team. Similarly, Shepperd [14] stated that the most straightforward method is to assuming a linear model which uses regression analysis to estimate the coefficients of the below shown equation.

$$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 \mathrm{x}_3 + \varepsilon \qquad (3)$$

where x_1 is Source Number of codes, x_2 is Adjusted Function Points and x_3 is Normalized Productivity Delivery Rate. y is the estimated effort in terms of manpower in man-month. α_0, α_1, α_2, and α_3 are the coefficients to be estimated. ε is the error term parameter with a normal distribution.

Since, in the previous section we have transformed the data set with natural logarithm in order to normalize the distribution of the data. The effort production function became as follows:

$$y^{\frac{1}{\beta}} = \alpha_0 + \alpha_1 \log x_1 + \alpha_2 \log x_2 + \alpha_3 \log x_3 + \varepsilon \qquad (4)$$

to expressed it more formally became following production function

$$y = (\alpha_0 + \sum_{i=1}^{k} \alpha_i \log x_i + \varepsilon)^{\beta} \qquad (5)$$

where k=3, β = 2,3,…., 10.

the equation can be also be expressed to easy use as follows:

$$y = (\alpha_0 + \log(\prod_{i=1}^{k} x_i^{\alpha_i}) + \varepsilon)^{\beta} \qquad (6)$$

We applied multiple linear regressions to the dataset. The regression model of MRE and adjusted R^2 were developed by applying stepwise regression. The resulting regression models, and connected significance values, from this process are listed in the Table 2. β is the power value for the effort estimation function (4). Coefficients are the estimated values for equation (4). MRE is the magnitude of relative error for each estimation function. In Table 2 it can be seen that the rate of MRE is decreasing when power is increasing. At power 5 we reach the required significant level of error rate. The adjusted R^2 is increasing very rapidly. In Figure 2 the change of adjusted R^2 can be followed. The accuracy of proposed model is very high because the Pred(0.25) values are very high and inversely the MRE values are very low.

Table 2. Regression Analysis results for different powered estimation functions.

β	**Coefficients** $\alpha_{0,} \alpha_1, \alpha_2, \alpha_3$	**MRE**	**R^2–adjusted**	**PRED(0,25)**
2	-240, -4.80, 45.3, 42.2	539,57	85,1%	71,4%
3	-30.5, -0.427, 6.40, 6.14	5,2589	92,8%	93,7%
4	-8.97, -0.109, 2.25, 2.19	0,3833	95,8%	96,4%
5	-3.57, -0.0433, 1.15, 1.13	0,0658	97,3%	99,1%
6	-1.56, -0.0219, 0.711, 0.700	0,0179	98,1%	>=99,1%
7	-0.616, -0.0128, 0.494, 0.487	0,0064	98,6%	>=99,1%
8	-0.107, –0.00829, 0.369, 0.365	0,0027	98,9%	>=99,1%
9	0.198, –0.00573, 0.290, 0.288	0,0013	99,1%	>=99,1%
10	0.393,–0.00416, 0.237, 0.235	0,0007	99,3%	>=99,1%

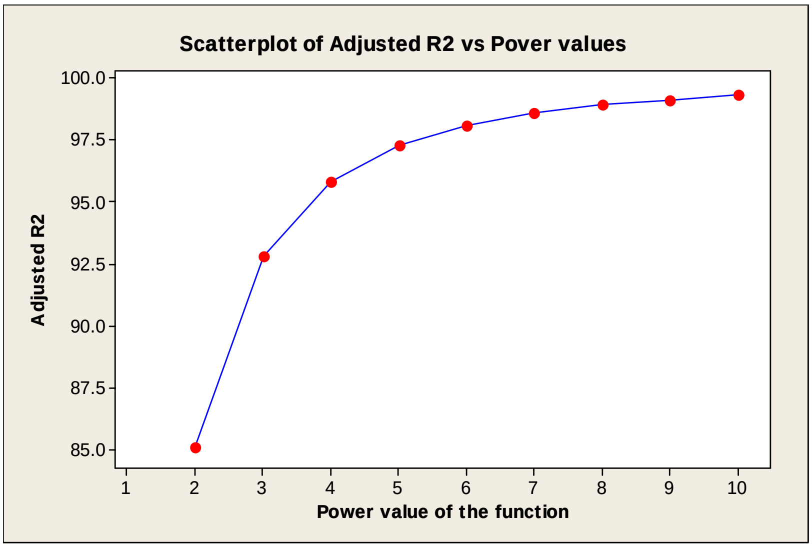

Figure 2. Change of adjusted R2 values according to power degree of the equation

Table 2 indicates the level of robustness of the variables included in the model. The variables included in our MRE-model are included in all best subset-models with number of variables greater than three, that is, the variable inclusion seems to be robust. The adjusted R^2 is satisfactory (85,1%), even for the model with the equation (4) who has power value of 2. This means that the majority of the variance in estimation accuracy is described by our model. A high adjusted R^2 entails that the model we found validates in explaining the relationship between the mean estimation accuracy and the included variables.

6 Comparison and Analysis of Results

We have compared our study with the study results of Liu [7], Stensrud [8] and Hu [9]. Hu uses Kamerer [9] data set for their analysis. He tries 5 different models and obtains best results for their calculations as adjusted R^2 89% for Minimum Software Cost Model (MSCM), adjusted R^2 56% for generalized Douglas production with time factor (GCDT) and adjusted R^2 54% for generalized Cobb-Douglas production (GCD). Liu uses the ISBSG data set as we have used and obtains an adjusted R^2 value as 61% in her study. Stensrud uses Laturi database as data set. He obtains MRE as %34 minimum value and PRED(0.25) value as %69. In our study we obtain better results than above stated results. In Table 2 we obtain a value for MRE as 0, 07461% which is very low for this kind of work and the obtained adjusted R^2 value is very high 99,3%. As a result we believe that we have obtained better results in regression based cost estimation area. Although the estimation function (5) is mathematically well known but it is not used in software cost estimation scope.

7 Discussion and Conclusion

In order to validate the model we adopt an original data set, made of 20 samples from the ISBSG data set which is not used in the training phase of the estimation equation. We obtain results given in Table 2. If we compare the obtained results with the literature they seem to be very impressive. The gap between the results from the study and from the literature is very big. We believe that by using data set from ISBSG and regression models to estimate software project efforts one can obtain results not better than from this study.

We believe that, the use of our estimation model can lead to better software cost estimation and fewer cost overruns in the software industry. In order to increase the realism of the study we have tried to use the real life data from ISBSG data set to make our estimation more realistic for use in real life projects. The next step in our research will be on understanding the relationship between project characteristics (data set properties) and estimation methods.

Acknowledgements

We would like to thank the International Software Benchmarking Standards Group (ISBSG) to approve for accessing the ISBSG data set for our research.

References

1. Grimstad S., Jorgensen, M. and Østvold K.M., “Software Effort Estimation Terminology: The tower of Babel”, Information and Software Technology 48, 2006, pp-302-310, Elsevier.
2. Boraso M., Montangero C. and Sedehi H., Software Cost Estimation: an experimental study of model performances”, Technical Report: TR-96-22, University of Pisa, Italy.
3. Wieczorek I. and Ruhe M., “How Valuable is company-specific Data Compared to multi-company Data for Software Cost Estimation?”, Proceedings of the Eighth IEEE Symposium on Software Metrics (METRICS.02).
4. Jones C., “Applied Software Measurement: Assuring Productivity and Quality”, McGraw-Hill, 1991
5. Bontempi G. And Kruijtzer K., “The use of intelligent data analysis techniques for system-level design: a software estimation example” Soft Computing 8 (2004) 477–490 _ Springer-Verlag 2003
6. Jorgensen, M. and Shepperd, M., “A Systematic Review of Software Development Cost Estimation Studies”, IEEE Transactions On Software Engineering, Vol. 33, No. 1, January 2007.
7. LIU Q. and Mintram R.C., Preliminary Data Analysis Methods in Software Estimation”, Software Quality Journal, 13, pp: 91-115, 2005.
8. Stensrud E. and Myrtveit I., “Human Performance Estimating with Analogy and Regression Models: An Empirical Validation”, Fifth International Symposium on Software Metrics (METRICS'98).
9. HU Q., Plant R.T. and Hertz D.B., “Software Cost Estimation Using Economic Production Models”, Journal of Management Information System, Vol. 15, No: 1, pp-143-163, 1998.
10. Dolado, J.J.,“On the problem of the software cost function”, Information and Software Technology, 43, pp: 61–72, 2001.
11. ISBSG: International Software Benchmarking Standards Group. http://www.isbsg.org.
12. Finnie G.R., Wittig G.E. and Desharnais J. M., “A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case-Based Reasoning and Regression Models”, Journal of Systems Software 1997, v 39, pp 281-289, Elsevier Science Inc.
13. Delany S.J., Cunningham P. and Wilke N., “The limits of CBR in Software Project Estimation”, German Workshop on Case-Based Reasoning, 1998.
14. Shepperd M. and Schofield C., “Estimating Software Project Effort Using Anologies”, IEEE Transactions on Software Engineering, Vol. 23, No.12, November 1997.