Using Industry Based Data Sets in Software Engineering Research

Qin Liu
University of Tongji
Shanghai, China

Robert Mintram
University of Bournemouth
Bournemouth, UK

ABSTRACT

This paper describes the use of software project development data obtained from industry based projects. It argues the importance of carrying out a preliminary data analysis procedure for software development cost estimation. The paper also presents the limitations of using these industrial data (ISBSG R9 and Bank63 Data) based on the above research. Current state of the research and further work is discussed.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Theory

Keywords

ISBSG Data Set

Introduction

One of the basic goals of software engineering is the establishment of useful models and equations to predict the cost of any given programming project [2]. The research [20, 21, 22, 23] mentioned in this paper provides a formal basis for building software project effort prediction models based on historical data sets. It defines a formal procedure to identify what project attributes and how much project data from an industrial historical data set should be used to build project effort prediction models. Some limitations of using industrial data in software engineering research are identified and discussed in the light of constructing software cost estimation models.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SSEE'06 May 20, 2006 Shanghai, China.
Copyright 2006 ACM 1-59593-085-X/06/0005 ...$5.00.

Data Based Research in Software Cost Estimation

Many software cost prediction techniques, models and tools have been proposed, for example, SLIM [31], COCOMO [5], Case Based Reasoning [35, 5], Bayesian belief networks [11] and artificial neural networks [17, 33]. In order to build such models, a historical data set containing a reasonable number of observations is required [5, 3, 4]. The variety of the data resources (e.g. government, industry, research institution, simulated data resources) and their characteristics (outliers, collinearity, number of features, number of cases etc) in software estimation inevitably caused inconsistent evaluation of cost models. Lack of data on completed software projects has also been an impediment for many years [14]. In addition, preliminary data analysis has rarely been emphasized [24], which means the data set used to train the prediction models and consequently to estimate effort could consist of outliers and interrelated factors. These factors often affect development effort and productivity, and mislead the understanding of relationships. Furthermore, at the early stage of the prediction, the model is often loosely defined as mapping all of the available predictor variables (project attributes) in a historical data set to the project effort, since, usually, there is little precognition of which factors are necessary for prediction. Consequently, constructing a specific model based on such loose definitions makes the prediction neither reliable nor efficient [18]. Therefore, there is a clear demand that a formal procedure is established to manage problematic data and select appropriate (significant) predictor variables from historical data sets. In effect, this means removing projects which appear as outliers and project attributes which appear as non-dominant or inter-related factors.

Constraints Related to Software Engineering Data

In the literature, we observed that software effort estimation data sets have some common features:

- The predicted variable, i.e. effort, is often positive, with a right skewed distribution [7, 8].
- They contain both continuous, for example project size or duration, and discrete, for example application types, variables [7];
- They contain significantly correlated or statistically not significant predictor variables [10, 25, 26].

Furthermore software engineering data sets tend to suffer from a number of problems:

- They are difficult and expense to collect. Given the nature of software development, large amounts of data cannot be collected in an experimental setting.
- They are often of limited sample size [14, 32]. The timescale of a typical software project means that relatively few data points may be collected from each organization per year.
- Outliers can and sometimes do have a great influence over estimation [7, 12, 11].
- The interdependencies of predictor variables can affect the understandability of models [10, 25, 26], but are not always harmful to their accuracy [7]. For example regression models [13].
- Academics have restricted access to industrial project data, due in part to the reluctance of companies to release such data into the public domain [5, 19].
- Missing values are a widespread problem as a result of lack of motivation on the part of some software engineers, some variables being more difficult to collect than others [30, 9, 34, 29].

In addition, when many variables are measured some practical problems arise. For example, with as few as 10 variables there are 45 correlations that must be considered; with 20 variables there are 190 correlations; with 40 variables there are 780; and the number of correlation coefficients rises rapidly as the number of variables increases [13]. Obviously, with large numbers of variables the number of relationships is so large as to be beyond computational tractability and data reduction techniques that can systematically summarize large correlation matrices are needed.

The Preliminary Data Analysis Framework

The principle motivation behind Liu's research [20, 21, 23] is to define a preliminary data analysis framework that provides a formal basis for building software project effort prediction models based on historical data sets. The proposed framework [20, 21] defines and evaluates a formal procedure to identify what project attributes and how much project data from a historical data set should be used to build project effort prediction models. The predictor variables are selected by their statistical power of explaining predicted variable's variation. The more variation of the predicted variable that is explained by a predictor variable, the more significant this predictor variable is. Building upon past work on preliminary data analysis which assisted project effort prediction[8, 26], the proposed framework is based on a set of statistical analysis methods such as correlation analysis, stepwise ANOVA, univariate analysis, etc. The principal contributions made by this research are:

- Formally establish a statistically based preliminary data analysis framework for analyzing software project data.
- Provides a basis for the erection of cost prediction models at early stage of the project development by defining a preliminary analysis framework of historical data sets.
- Provides empirical evidence to support the accuracy and applicability of the above framework.

The proposed framework is empirically evaluated against commonly used prediction methods, namely Ordinary Least-Square Regression (OLS), Robust Regression (RR), Classification and Regression Trees (CART), K-Nearest Neighbour (KNN), and is also applied to both heterogeneous and homogeneous data sets. Formal statistical significance testing was performed for the comparisons.

The results from the comparative evaluation suggest that the proposed preliminary data analysis framework is capable to construct more accurate prediction models for all selected prediction techniques. The framework processed predictor variables are statistic significant, at 95% confidence level for both parametric techniques (OLS and RR) and one non-parametric technique (CART). Both the heterogeneous data set (ISBSG R9 Data) and homogenous data set (Bank63 Data) benefit from the application of the proposed framework for improving project effort prediction accuracy. The homogeneous data set is more effective after being processed by the framework. Overall, the evaluation results demonstrate that the proposed framework has an excellent applicability.

Contribution and Limitations

This research initiative is the first to undertake a wide-scale and consistent comparison of cost models on relatively large software engineering data sets. Although it overcomes several of the weaknesses of other empirical studies, there are some limitations of this work related to construct, internal, and external validity [16] of the data set variables.

Construct Validity Limitations

Construct Validity refers to the confounding levels of construct [16]. In other words, assuming there is a causal relationship in this study, can we claim that the program reflected well our construct of the program and that our measure reflected well our idea of the construct of the measure? Within the context of software effort estimation, it refers to the functional relationship of the models, as discussed by Liu and Mintram [22].

Without expert opinion and additional insights into the projects' details, this study was limited concerning the application of pure data-driven estimation. Several studies [37] emphasize the importance of expert input. Moreover, without the assistance of "inside knowledge" suitable weights could not be assigned to variables and individual specific projects could not be identified as outliers or "exceptional" cases. A possible objective method to assign weights would be to incorporate them according to the relative significance of the variables, such as the percentage of the explained variation identified by the proposed framework.

The current study's construct validity is the extent to which the prediction accuracy, is operationalized, measured by magnitude of relative error (MRE). Construct validity is threatened since there are drawbacks associated with the MRE measure [27]. There is an apparent inconsistency in the way the effort estimation methods optimize the MRE.

However, the MRE is a widely used measure for evaluating estimation methods and thus was used in this study. As a cross-check, we applied alternative evaluation measures and found no significant changes in the general trends of the results.

Internal Validity Limitations

Internal validity refers to confidence that the findings from a given study or experiment are attributable to the treatment alone [16]. The internal validity limitations of Liu's research are discussed as the following.

Project Size Measurement

It is widely accepted that among the factors that affect the cost of software projects software size is one of the key cost drivers [6]. The size of a software system was measured in Function Points in the both data sets in this study.

The Function Point (FP) count is a weighted sum of different function types and can further be adjusted for technical processing complexities. However, the reason supporting the adjustment is not clear and it is also of concern that using the complexity weights does not necessarily explain more variation in effort compared than non-weighted function points [15]. To avoid this problem, in this study, we used the unadjusted function point count from ISBSG R9 data set. However, for Bank63 data set we only have the so called Experience Function Points. Care must be taken when using a model constructed from data from one environment to make predictions about data from another environment.

Currency of the Data

Another limitation of models constructed using historical data is that attributes used to predict software development effort can change over time and/or differ between software development environments [36]. Mohanty [28] makes this point in comparisons between the predictions of a wide variety of models on a single hypothetical software project. In this study we used the latest version of the ISBSG data set with 70% of the projects being less than six years old. However, the Bank63 data is not recent, being collected in 1993.

Data Resource Limitations

Data are collected for this research under the guidance of formal procedures. The data are verified and maintained by highly reputable institutions (ISBSG) and experts (K. Maxwell). Despite these assurances, it should be noted that the data collection process was beyond the control of the current study. The reliability of the measurements for all variables, the sample size, and causes for missing data could not be specifically determined and are understood to be potential problem areas. It should also be emphasized that the presented results are explicitly based upon the data collected.

External Validity Limitations

External validity refers to the generalizability of the findings to other populations,settings, occasions. Within the ISBSG R9 data set, the project IDs are totally random. There is no company-specific data available for evaluating cost models' accuracy based on heterogeneous and homogeneous data. If one or more companies within the ISBSG R9 data set were examined, this could have revealed different results. Therefore, the comparison of results between heterogeneous data sets and homogeneous data sets should be considered in the context of both the entire ISBSG R9 data and Bank63 data set. Thus, the external validity of the research is threatened at the company-specific data level.

Conclusions

Within the context of the limitations discussed above this study concludes that the effectiveness of historical data sets in the generation of cost estimation/prediction models depends critically on the nature of the data within the data set. In particular, to be effective the data sets should be subjected to a formal process of "data cleaning". It is observed that there may be further problems hidden within such data sets related to the issue of heterogeneous versus homogeneous data origins. i.e., similarity of projects and the conducting institutions.

Further Research

Further research could focus on two main purposes: First, improve the applicability by integrating missing data techniques such as listwise deletion (LD) [30], mean imputation (MI) [30], similar response pattern imputation (SRPI) and full information maximum likelihood (FIML), sparse data method (SDM) [34] etc., for handling missing values in historical data sets. Second, apply benchmarking to enable comparisons, i.e. allowing companies to compare themselves with respect to their productivity or quality.

In addition, as argued by Stensrud *et al.* [37] and Baik *et al.* [1], resource estimation methods are typically not used in isolation but as a support tool for human estimators. It is therefore important that research in cost estimation takes into account this human factor component. Methods that combine expert knowledge with data driven techniques may allow individual organizations to develop specific resource estimation models while alleviating the stringent requirements for project data, e.g., Bayesian Analysis method. Such methods make use of all information available in organizations and therefore make practical and economic sense.

1. REFERENCES

[1] J. Baik, B. Boehm, and B. M. Steece. Disaggregating and calibrating the case tool variable in COCOMO II. *IEEE Transactions on Software Engineering*, 28(11):1009–1022, 2002.

[2] J. Bailey and V. Basili. A meta-model for software development resource experiments. In *Proceedings of the 5th International Software Engineering*, pages 107–116, Los Alamitos, California, 1981. IEEE CS Press.

[3] V. R. Basili. Quantitative evaluation of software methodology. In *Proceedings of the First Pan Pacific Computer Conference*, 1985.

[4] V. R. Basili and H. D. Rombach. The TAME project: Towards improvement-oriented software environments. *IEEE Transactions on Software Engineering*, 14(6):758–773, August 1988.

[5] B. W. Boehm. *Software Engineering Economics*. Prentice Hall, Englewood Cliffs, NJ, 1981.

[6] B. W. Boehm, C. Abts, B. A. Insor, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. J. Reifer, and B. Steece. *Software Cost Estimation with COCOMO II*. Prentice Hall PTR, Upper Saddle River, NJ, 2000.

[7] L. C. Briand, V. R. Basili, and W. Thomas. A pattern recognition approach for software engineering data analysis. *IEEE Transactions on Software Engineering*, 18(11):931–942, 1992.

[8] L. C. Briand and J. Wüst. The impact of design properties on development cost in Object-Oriented systems. In *Proceedings of the 7th IEEE International Software Metrics Symposium (METRICS 2001)*, pages 260–271, London, England, 4-6 April 2001. IEEE Computer Society.

[9] M. Cartwright, M. J. Shepperd, and Q. Song. Dealing with missing software project data. In *Proceedings of the 9th IEEE International Metrics Symposium*, Sydney, Australia, 2003. IEEE Computer Society.

[10] S. Chulani. Incorporating Bayesian analysis to improve the accuracy of COCOMO II and its quality model extension. Qualifying exam report, University of Southern California, February 1998.

[11] S. Chulani, B. W. Boehm, and B. Steece. Bayesian analysis of empirical software engineering cost models. *IEEE Transactions on Software Engineering*, 25(4):573–583, 1999.

[12] R. Cordero, M. Constamagna, and E. Paschetta. A genetic algorithm approach for the calibration of COCOMO-like models. In *Proceedings of the 12th COCOMO Forum*, 1997.

[13] W. R. Dillon and M. Goldstein. *Multivariate Analysis: Methods and Applications*. John Wiley & Sons, New York, 1984.

[14] F. Heestra. Software cost estimation. *Information and Software Technology*, 34(10):627–639, 1992.

[15] R. Jeffery and J. Stathis. Function point sizing: Structure, validity and applicability. *Empirical Software Engineering*, 1(1):11–30, 1996.

[16] C. M. Judd, E. R. Smith, and L. H. Kidder. *Research Methods and Social Relations*. Harcourt Brace Jovanovich College Publishers, USA, 6th edition, 1991.

[17] N. Karunanithi, D. Whitley, and Y. K.Malaiya. Using neural networks in reliability prediction. *IEEE Software*, 9(4):53–59, 1992.

[18] B. A. Kitchenham and N. R. Taylor. Software cost models. *ICL Technical Journal*, 4(1):73–102, May 1984.

[19] B. A. Kitchenham and N. R. Taylor. Software projects development cost estimation. *Journal of Systems and Software*, 5(4):267–278, 1985.

[20] Q. Liu and R. C. Mintram. Preliminary data analysis methods in software estimation. In *Proceedings of the BCS 2004 SQM and INSPIRE Conferences*, pages 273–298, Canterbury, UK, ISBN 1-902505-56-5, April 5-7 2004.

[21] Q. Liu and R. C. Mintram. Preliminary data analysis methods in software estimation. *Software Quality Journal*, 13(1):91–115, 2005.

[22] Q. Liu and R. C. Mintram. The problem of estimating the duration of software development projects. In *Proceedings of the BCS 2005 SQM and INSPIRE Conferences*, pages 293–299, Cheltenham, UK, ISBN:1-902505-67-0, March 21-23 2005.

[23] Q. Liu, R. C. Mintram, and J. Vincent. Evaluation of cost estimation models. In *Proceedings of the International Conference on Computer Science and Information Systems*, Athens, Greece, 2005. Athens Institute For Education And Research.

[24] Q. Liu and W. H. R. C. Mintram. Recent advances in software estimation techniques. In *Proceedings of the BCS 2006 SQM and INSPIRE Conferences*, Southampton UK, 2006. British Computing Society.

[25] K. Maxwell, L. V. Wassenhove, and S. Dutta. A software development productivity of european space, military and industrial applications. *IEEE Transactions on Software Engineering*, 22(10):704–718, 1996.

[26] K. D. Maxwell. *Applied Statistics for Software Managers*. Pearsson Education Inc., UpperSaddle River, New Jersey, 2002.

[27] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki. Robust regression for developing software estimation models. *Journal of Systems and Software*, 27(1):3–16, 1994.

[28] S. Mohanty. Software cost estimation: Present and future. *Software Practice and Experience*, 11(2):103–121, 1981.

[29] J. Moses and M. Farrow. Assessing variation in development effort consistency using a data source with missing data. *Software Quality Journal*, 13(1):71–89, 2005.

[30] I. Myrtveit, E. Stensrud, and U. H. Olsson. Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods. *IEEE Transactions on Software Engineering*, 27(11):999–1013, 2001.

[31] L. H. Putnam. A general empirical solution to the macro software sizing and estimating problem. *IEEE Transaction Software Engineering*, 4(4):345–361, 1978.

[32] H. Rubin. Software process maturity: Measuring its impact on productivity and quality. *Proceedings of the 15th International Conference on Software Engineering*, pages 468–476, 1993.

[33] B. Samson, D. Ellison, and P. Dugard. Software cost estimation using an albus perceptron (cmac). *Information and Software Technology*, 39(1):55–60, 1997.

[34] M. Shepperd and M. Cartwright. Predicting with sparse data. *IEEE Transactions on Software Engineering*, 27(11):987–998, 2001.

[35] M. Shepperd and C. Schofield. Estimating software project effort using analogies. *IEEE Transactions on Software Engineering*, 23(12):736–743, 1997.

[36] K. Srinivasan and D. Fisher. Machine learning approaches to estimating software development effort. *IEEE Transactions on Software Engineering*, 21(2):126–137, 1995.

[37] E. Stensrud and I. Myrtveit. Human performance estimation with analogy and regression models. In *Proceedings of the IEEE International Symposium on Software Metrics (METRICS 1998)*, pages 205–213. IEEE Computer Society Press, 1998.