

Multiple Logistic Regression as Imputation Method Applied on Software Effort Prediction

Panagiotis Sentas, Lefteris Angelis, Ioannis Stamelos, G. L. Bleris
Department of Informatics,
Aristotle University of Thessaloniki
54124, GREECE
email: {psentas, lef, stamelos, bleris}@csd.auth.gr

Abstract

A common problem in software cost estimation is the manipulation of incomplete or missing data in databases used for the development of prediction models. In such cases, the most popular and simple method of handling missing data is to ignore either the projects or the attributes with missing observations. This technique causes the loss of valuable information and therefore may lead to inaccurate cost estimation models. On the other hand, there are various imputation methods used to estimate the missing values in a data set. These methods are applied mainly on numerical data and produce continuous estimates. However, it is well known that the majority of the cost data sets contain software projects with mostly categorical attributes with many missing values. It is therefore reasonable to use some estimating method producing categorical rather than continuous values. The purpose of this paper is to investigate the possibility of using such a method for estimating categorical missing values in software cost databases. Specifically, the method known as Multinomial Logistic Regression (MLR) is suggested for imputation and is applied on projects of the ISBSG multi-organizational software database. Comparisons of MLR with other missing data techniques, such as listwise deletion (LD), mean imputation (MI), expectation maximization (EM) and regression imputation (RI) show that the proposed method is efficient, especially when the percentage of missing values is high.

Keywords: Software effort prediction, cost estimation, missing data, imputation, multinomial logistic regression, ISBSG.

1 Introduction

The importance of Software Cost Estimation (SCE) as one of the most crucial phases of software development has been recognized since long. Attempts to estimate the effort and time involved in the development of a software product usually involve the construction of one or more cost estimation models (see for example [1], [2]) by applying statistical methods to historical data sets with completed software projects. Most commonly, cost models are obtained by applying regression methods ([3], [4]).

A major problem in building such a model arises from the fact that missing values are often encountered in these historical data sets ([4], [5]). The lack of data values in several important project attributes is a common phenomenon, which may cause misleading results regarding the models' accuracy and prediction ability. The fact is that many software databases suffer from missing values and that is the result of several reasons related to the demanding process of collecting adequate data. Indeed, the data collection requirements include consistence, experience, time, cost and methodology for a company. Furthermore, when the model is based on multi-organizational data, the problem of missing values is caused by the different methods the various companies use to measure and record their data.

There are various techniques dealing with missing data. The most common one, known as *listwise deletion* (LD) [6], simply ignores the cases with missing observations. The major advantage of the method is its simplicity and the ability to do statistical calculations on a common sample base of cases. The disadvantages of the method are the dramatic loss of information in data sets with high percentages of missing values and the possible bias in the data. These drawbacks are more or less always apparent especially when there is some type of pattern in the missing data, i.e. when the distribution of missing values depends on certain valid observations in the data.

According to some other techniques, the missing values are replaced by estimates obtained from statistical procedures. The complete data set resulting from such a process is then analyzed by standard statistical methods (for example regression analysis). These techniques are commonly known as *imputation methods* [6]. The problem is that most of the imputation methods produce in general continuous estimates, which are not realistic replacements of the missing values when the variables are categorical. Since often the majority of the variables in the software data sets are categorical with many missing values, it is reasonable to use an imputation

method producing categorical values in order to fill the incomplete data set and then to use it for constructing a prediction model.

In this paper we investigate the possibility of using the statistical procedure known as Multinomial Logistic Regression (MLR) as imputation method for categorical variables. The data we used for experimentation and comparisons come from the International Software Benchmarking Standards Group (ISBSG) multi-organizational software database [7]. MLR was applied on a set of carefully selected software projects that have been pre-processed by statistical analysis and was compared with four other missing data techniques: listwise deletion (LD), mean imputation (MI), expectation maximization (EM) and regression imputation (RI).

In order to compare the efficiency of the above methods in various incompleteness situations, the data set we used was originally complete (no missing values) and the missing values were created by simulating three different mechanisms: missing completely at random (MCAR), missing at random (MAR) and non-ignorable missingness (NI).

The comparisons were conducted on the basis of the predictive accuracy of a regression model fitted to the data after applying each missing data technique. The results of the study indicate that for small percentages of missing values, MLR gives as satisfactory results as all the other methods do. However, when the percentages of missing values increase, MLR typically outperforms the other methods.

The structure of the paper is the following: In Section 2 we outline related work in the area. In Section 3 we describe the different mechanisms used to create missing data and the most common techniques for handling them. In Section 4 we present the data set used in the analysis and the statistical methods in details. In Section 5 we give the results and finally in Section 6 we conclude by discussing the findings as well as some directions for future work.

2. Related work

Although the problem of handling missing data has been treated adequately in various real world data sets, there are rather few published works concerning data in the field of software engineering.

In [8], two imputation methods, class mean imputation (CMI) and k-nearest neighbors (k-NN), were considered with respect to two mechanisms of creating missing data: missing completely at random (MCAR) and missing at random (MAR).

Emam and Birk [9] have used multiple imputation in order to induce missing values in their analysis of software process data performance.

Strike et al. [4] compared LD, MI and eight different types of hot-deck imputation for dealing with missing data in the context of software cost modeling. Three missing data mechanisms were evaluated (MCAR, MAR and NI) and two patterns of missing data were simulated (univariate and monotone) in order to induce missing values on a complete large software engineering data set. The results showed that all the missing data techniques performed well and therefore the simplest technique, LD, is a reasonable choice. However, best performance was obtained by using hot-deck imputation with Euclidean distance and a z-score standardization.

Myrtveit et al. [10] evaluated LD, MI, similar response pattern imputation (SRPI) and full information maximum likelihood (FIML) on an enterprise resource planning (ERP) data set with real missing values. Their results indicated that FIML is appropriate when data are not missing completely at random. LD, MI and SRPI resulted in biased prediction models unless the data is MCAR. Also, MI and SRPI, compared to LD, were suitable only if the data set after applying LD was very small in order to construct a meaningful prediction model.

M. Cartwright et al. [11] examined sample mean imputation (SMI) and k-NN on two industrial data sets with real missing data. They found that both methods improved the model fit but k-NN gave better results than SMI.

3. Missing Data Mechanisms and Missing Data Techniques

3.1 Missing Data Mechanisms

The methods of handling missing data are directly related to the mechanisms that caused the incompleteness. Generally, these mechanisms fall into three classes [6]:

1. Missing Completely at Random (MCAR): The missing values in a variable are unrelated to the values of any other variables, whether missing or valid.
2. Non-Ignorable missingness (NI): NI can be considered as the opposite of MCAR in the sense that the probability of having missing values in a variable depends on the variable itself (for example a question regarding skills may be not answered when the skills are in fact low).
3. Missing at Random (MAR): MAR can be considered as an intermediate situation between MCAR and NI. The probability of having missing values, does not depend on the variable itself but on the values of some other variable.

Unfortunately, it is very difficult to recognize if the mechanism is MCAR, MAR or NI in a data set with real missing values. Therefore, a reasonable approach in order to investigate the problem of incomplete data sets is to generate artificial missing values from complete databases.

3.2 Missing Data Techniques (MDTs)

The techniques used in this paper for handling missing data and for comparisons with MLR are [6]:

1. Listwise Deletion (LD): It is a typical method that belongs to a broader class, namely the deletion methods. According to LD, cases with missing values for any of the variables are omitted from the analysis. The procedure is quite common in practice because of its simplicity, but when the percentage of missing values is high, it results in a small complete subset of the initial data sets and therefore in difficulties in constructing a valid cost model.
2. Mean imputation (MI): This method replaces the missing observations of a certain variable with the mean of the observed values in that variable. It is a simple method that generally performs well, especially when valid data are normally distributed.
3. Regression Imputation (RI): The missing values are estimated through the application of multiple regression where the variable with missing data is considered as the dependent one and all other variables as predictors.
4. Expectation Maximization (EM): The EM algorithm is an iterative two step procedure obtaining the maximum likelihood estimates of a model starting from an initial guess. Each iteration consists of two steps: the Expectation (E) step that finds the distribution for the missing data based on the known values for the observed variables and the current estimate of the parameters and the Maximization (M) step that replaces the missing data with the expected value.

The reason for choosing LD and MI is that both are very popular methods in software cost estimation and very simple in their implementation. RI is a method based on a regression model and so it was interesting to compare it with the model produced by MLR. EM was chosen because it is an elegant and sophisticated method that combines statistical methodology and algorithmic implementation. Moreover, it has gained much attention recently in various applications and in general is considered a very promising method.

3.3. Multinomial Logistic Regression (MLR).

The method we investigate in this paper is a generalization of the Logistic Regression (LR) method which is used to model the relationship between a dichotomous (binary) dependent variable and a set of k predictor variables $\{x_1, x_2, \dots, x_k\}$, which are either categorical (factors) or numerical (covariates). As the binary dependent variable can be always interpreted as the occurrence or not of an event E , the logistic regression model is an expression of the form

$$\log\left(\frac{\text{prob}(E)}{1 - \text{prob}(E)}\right) = b_0 + \sum_{i=1}^k b_i x_i \quad (1)$$

where the b_i 's denote the unknown logistic regression coefficients (b_0 is the intercept) while $\text{prob}(E)$ denotes the probability that event E will occur. The quantity on the left side of equation (1) is called a *logit*. So, the simple LR model can be used for predicting the probability of an event occurrence.

The model can be generalized in the case where the dependent variable is polytomous, i.e. its values are more than two categories. In such a case, if we assume that the possible categories are q , we need to model $q - 1$ logits,

$$\log\left(\frac{\text{prob}(\text{category } j)}{\text{prob}(\text{category } q)}\right) = b_0^{(j)} + \sum_{i=1}^k b_i^{(j)} x_i, \quad j = 1, \dots, q - 1. \quad (2)$$

In (2), we can see that one of the categories is used as reference and is called *baseline* category. After estimating the coefficients of the model (2) by the method of maximum likelihood, we can readily calculate the logits and therefore the probabilities of each one of the categories. The final prediction is the category with the maximum probability.

MLR can be used for imputation by considering dependent the categorical variable with the missing values and predictors all the others. Note that a similar method has already been used for prediction of productivity in [12].

4. Research methodology

The method of approach for the comparison of MLR with the other four MDTs was based on studying the impact of each one of the MDTs on the predictive accuracy of a cost estimation model. Below we describe the data set used, the preprocessing for obtaining a final complete data set, the cost model derived from the data and the accuracy measure employed for the comparisons.

4.1 Selection of the data set

The database originally used to derive our complete data set is the ISBSG (release 7) project repository, which contains 1238 multi-organizational projects [7]. The large number of missing values in almost all of the important variables and the different methods of measuring the work effort and the size of the projects resulted in the significant reduction of the data. Specifically, considering the data quality rating (following the recommendation of ISBSG7), the homogeneity of the measurements and the projects with missing data, we ended up with a complete data set with 166 projects.

4.2 Variables of the cost model

The cost model was constructed by considering as dependent variable the logarithm of work effort (*ln_{effort}*). Only two of the predictor variables are numerical, the year of implementation (*year*) and the logarithm of the size of the project in function points (*ln_{size}*). The logarithmic transformations for effort and size were applied since they improve the efficiency of the model [13].

Since the rest of the predictor variables are categorical (factors) with a large number of categories each, we conducted a preliminary study in order to merge the original categories into homogeneous groups and therefore to work with only a few categories for each factor. This was achieved by analysis of variance (ANOVA) to identify the most important factors for the effort and multiple range tests for concatenation of the homogeneous categories of the important factors. The categories of each factor were represented by ordinal numbers according to their mean impact on the effort. The categorical variables resulted from this study are given in Table 1.

Table 1
The categorical predictor variables used for the cost model

Variable	Description & Levels
<i>develp</i>	Development type. Levels: Enhancement, New Development, Re-development
<i>platfr</i>	Development platform. Levels: MainFrame, Mid Range, PC
<i>lang</i>	Language type. Levels: 3GL, 4GL, Application Generator
<i>primar_4</i>	Primary Programming Language. Levels (after merging): 1={access}, 2={easytrieve, natural, oracle, pl/i, power builder, visual basic}, 3={cobol, ideal, other apg, sql, telon}, 4={C, C++, clipper, cobol II, other 4gl}
<i>orgtype_4</i>	Organisation Type. Levels (after merging): 1={communication, computers, consultancy, energy, financial, property & business services, medical and health care, professional services, public administration, wholesale & retail trade}, 2={banking, community services, construction, defence, electronics, insurance, manufacturing, mining}, 3={electricity, gas, water}, 4 ={aerospace/automotive, consumer goods, distribution, government, occupational health and safety, transport & storage}
<i>bartype_4</i>	Business Area Type. Levels (after merging): 1={activity tracking, claims processing-product pays claim, environment, fine enforcement, generate & distribute electricity, project management & job control, research & development, sales & marketing, telecommunications}, 2={engineering}, 3={accounting, banking, financial (excluding banking), provide computer services and IT consultation}, 4={architectural, blood bank, chartered flight operation, customs, energy generation, insurance, inventory, legal, logistics, manufacturing, pension funds managements, personnel, procurement, public administration, transport/ shipping}
<i>apltype_4</i>	Application Type. Levels (after merging): 1={advertising/ mailing campaign, corporate taxation, data warehouse, decision support system, network management, reconciliation, transportation}, 2={management information system, process control}, 3={inventory control, system conversion, transaction/production system}, 4={office information system, technical information system}

4.3. Predictive accuracy of a model

In order to measure the predictive accuracy of a cost estimation model, we randomly split the data set into two subsets: a training set with 150 projects and a test set with 16 projects. A cost regression model was built on the training data set and its accuracy was evaluated on the test data set. The measure of accuracy used is the one suggested by Foss et al in [14], namely the standard deviation (SD) of the residual error:

$$SD = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n-1}},$$

where y_i and \hat{y}_i denote the actual and the predicted effort respectively in the test set of size n .

4.4. The regression model

In order to fit an effort prediction model in our training data set, we used stepwise ordinary least squares (OLS) regression. The procedure resulted in a model with the following predictor variables: *lnsize*, *platfr*, *bartype_4*, *primar_4* and *apltype_4*. The fitting of the model and the significance of each variable were evaluated by the usual statistics. From now on, we will call this model "*baseline model*" as it has been built on the original training data set.

4.5. Simulation of missing values

In order to explore the performance of MDTs, under different missing data mechanisms, we used the complete training data set in order to artificially generate missing values by mimicking the MCAR, MAR and NI mechanisms. A similar approach was followed in [4] and [8].

The MCAR mechanism was simulated by creating missing values completely at random for each variable. For the MAR mechanism, we first sorted the projects of the training data set in ascending order of their size. Then, we divided the sorted data set into five subsets of equal size. If the total percentage of missing data is p , the percentages of missing values in each one of the five subsets will be $0.4p$, $0.3p$, $0.2p$, $0.1p$ and 0 respectively, so that the missing values will be induced with bias related to size. The missing values in each subset were assigned randomly. For the NI mechanism, the same procedure as MAR was followed, except now the projects in the training data set were sorted according to the same variable where the missing values were generated.

4.6. The design of the study

The study design involves three percentages of incomplete data: 10%, 20% and 30%, which are quite realistic for any data set. Since the baseline model has four categorical predictor variables, we considered missing values on all of them, one at a time. The above percentages were combined with the three different missing data mechanisms (MCAR, MAR and NI) and the five techniques for handling missing data. So, in total 180 different models were constructed from the original data set and the accuracy of each one was measured by the SD computed from the predictions on the test data set.

5. Results

The results of the study were tabulated and examined carefully. Due to the space limitations, we give here a summary of the results:

- LD and MI are efficient methods when the percentage of missing data is small (10%). For percentages 20% and 30% these method are not suitable.
- EM is a "stable" method. It is characterized of the absence of any intensive fluctuations of the resulting models and achieves average results for all percentages and mechanisms of missing data.
- RI is a very efficient method but in our experiments MLR performed better. For 10% missing values, the method cannot be distinguished from MI, LD or EM. When the percentage of missing values increases, RI and MLR in general outperform LD and MI.
- MLR is the method, which in our experiments gave generally the best results, especially for missing value percentages 20% and 30%. When the missing data mechanism is either MCAR or NI, the MLR method provides better results for all percentages of incompleteness. When the mechanism is MAR, the predictive accuracy of MLR is directly comparable with that of RI.

Table 2 is indicative of the aforementioned results. It presents for each percentage the number of times that a MDT gave the best accuracy measured by SD.

Table 2
The number of times that each MDT gave the best accuracy

10% of missing data					
	MI	LD	EM	RI	MLR
MCAR				1	3
MAR			1	1	2
NI		1			3
20% of missing data					
	MI	LD	EM	RI	MLR
MCAR				2	2
MAR				2	2
NI		1		1	2
30% of missing data					
	MI	LD	EM	RI	MLR
MCAR				1	3
MAR				2	2
NI				1	3

6. Conclusion and future work

In this paper we investigated the use of multinomial logistic regression for estimating the missing values of categorical variables, predictors in software cost models. We experimented by considering a complete data set of software projects, by generating artificial missing values in four categorical predictor variables with three different mechanisms and finally by handling the missing data by the method under investigation along with four other well-known methods.

The results are encouraging in the sense that a purely categorical method performs better or at least equally well when compared to other popular methods applied to missing data. Future work includes the application of MLR to real missing categorical data, especially in the full ISBSG database. Also, one of our aims is to study other categorical methods either for exploring the reasons of incompleteness in databases or for estimating the missing values.

References

- [1] M.J. Shepperd, C. Schofield, Estimating Software Project Effort Using Analogies, *IEEE Trans. on Software Engineering*, 23 (1997), 736-743.
- [2] R. Jeffery, M. Ruhe and I. Wiecek, "A Comparative Study of Two Software Development Cost Modelling Techniques Using Multi-organizational and Company- Specific Data", *Information and Software Technology*, 42 (2000), 1009-1016.
- [3] L. Angelis, I. Stamelos, M. Morisio, Building a Software Cost Estimation Model Based on Categorical Data, *Proceedings of the 7th IEEE International Software Metrics Symposium*, 2001, 4-15.
- [4] K. Strike, K.E. Emam, and N. Madhavji, "Software Cost Estimation with Incomplete Data," *IEEE Trans. Software Eng.*, vol. 27, no. 10, pp. 890-908, Oct. 2001
- [5] L. Briand, V. Basili, and W. Thomas, "A Pattern Recognition Approach for Software Engineering Data Analysis," *IEEE Trans. Software Eng.*, vol. 18, no. 11, pp. 931-942, Nov. 1992.
- [6] R.J.A. Little and D.B. Rubin, *Statistical Analysis with Missing Data*, 2nd edition, New York, Wiley, 2002.
- [7] ISBSG Data Disk, Release 7, June 2001.

- [8] Q. Song, M. Shepperd, "A Short Note on Safest Default Missingness Mechanism Assumptions" ESERG Technical Report TR02-07, Bournemouth University ([url: dec.bournemouth.ac.uk/ESERG/TechnicalReports.html](http://dec.bournemouth.ac.uk/ESERG/TechnicalReports.html)), Aug. 2003.
- [9] K. E. Emam and A. Birk, "Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Process Capability", IEEE Trans. Software Eng., vol. 26, no. 6, pp. 541-566, June 2000.
- [10] I. Myrtveit, E. Stensrud, and U. Olsson, "Analyzing Data Sets with Missing Data: An Empirical Evaluation of Imputation Methods and Likelihood-Based Methods", IEEE Trans. Software Eng., vol. 27, no.11, pp. 999-1013, Nov 2001
- [11] M. H. Cartwright, M. J. Shepperd and Q. Song, "Dealing with Missing Software Project Data" Proc. METRICS, pp. 154-165, 2003
- [12] P. Sentas, L. Angelis, I. Stamelos, "Ordinal Regression Applied on Software Productivity and Effort Prediction". Information and Software Technology, to appear.
- [13] K. Maxwell, Applied Statistics for Software Managers, Prentice-Hall, 2002
- [14] T. Foss , E. Stensrud, B. Kitchenham, I. Myrtveit, "A Simulation Study of the Model Evaluation Criterion MMRE", IEEE Trans. Software Eng., vol. 29, no.11, pp. 985-995, Nov. 2003.