Ensemble of Missing Data Techniques to Improve Software Prediction Accuracy

Bhekisipho Twala
Brunel University
UB8 3PH
United Kingdom
+44 (0) 1895 266 041
bhekisipho.twala@brunel.ac.uk

Michelle Cartwright
Brunel University
UB8 3PH
United Kingdom
+44 (0) 1895 267 187
michelle.cartwright@brunel.ac.uk

Martin Shepperd
Brunel University
UB8 3PH
United Kingdom
+44 (0) 1895 267 188
martin.shepperd@brunel.ac.uk

Abstract
Software engineers are commonly faced with the problem of incomplete data. Incomplete data can reduce system performance in terms of predictive accuracy. Unfortunately, rare research has been conducted to systematically explore the impact of missing values, especially from the missing data handling point of view. This has made various missing data techniques (MDTs) less significant. This paper describes a systematic comparison of seven MDTs using eight industrial datasets. Our findings from an empirical evaluation suggest listwise deletion as the least effective technique for handling incomplete data while multiple imputation achieves the highest accuracy rates. We further propose and show how a combination of MDTs by randomizing a decision tree building algorithm leads to a significant improvement in prediction performance for missing values up to 50%.

Categories and Subject Descriptors
D.2; E.0

General Terms
Management; performance

Keywords
Machine learning; decision trees; ensemble; software prediction; incomplete data.

1. INTRODUCTION AND RELATED WORK
Accurate effort prediction is a challenge in software engineering. Industrial datasets are used to build and validate prediction systems of software development effort. A very important and common issue faced by researchers who use industrial and research datasets is incompleteness of data. Even if part of a well thought out measurement programme, industrial datasets can be incomplete, for a number of reasons. These include inaccurate or non-reporting of information (without a direct benefit, a project manager or developer might see data collection as an overhead they can ill afford, for example) or, where data from a number of different types of projects or from a number of companies are combined, certain fields may be blank because they are not collectable for all projects. Often data is collected either with no specific purpose in mind (i.e. it is collected because it might be useful in the future) or the analysis being carried out has a different goal than that for which the data was originally collected. Software engineering researchers have become increasingly aware of the problems and biases caused by missing or incomplete data. In fact, not only is any analysis not addressing incomplete data problematic but the presence of missing data may result to misleading conclusions drawn from a research study and limit generalizability of the research findings. The seriousness of these problems depends in part on how much data is missing, the pattern of missing data, and the mechanism underlying the missingness of the data. There has been a large increase in the amount of knowledge for dealing with incomplete data on fields such as marketing [12], education [16], economics [10], psychometrics [4], medicine [1] and nursing [14]. Unfortunately, there has been conspicuously little research concerning missing data problems in the software engineering literature. The literature abounds with a variety of procedures, including discarding instances with missing values from the study and imputing *ad hoc* values, that fail to deliver efficient and unbiased parameter estimates. Some of the informative papers are described below.

[13] perform a simulation study comparing two missing data imputation methods based on machine learning (ML) algorithms. Their results show that for the single imputation task, the supervised learning algorithm, C4.5 [17], which utilizes the fractional cases (FC) strategy, performed better than Autoclass [6], a strategy based on unsupervised Bayesian probability. For the multiple imputation (MI) task, both methods perform comparably.

MI is used by [8] to handle missing values in their empirical study that evaluated the predictive validity of software requirements analysis. The study was conducted on 56 projects.

[20] perform a comprehensive simulation study to evaluate three MDTs in the context of software cost modelling. These techniques are listwise deletion (LD), mean or mode single imputation (MMSI) and eight different types of hot deck single imputation (HDSI). Their results show LD as not only having a severe impact on regression estimates but yields small bias as well. However, the precision of LD worsens with increases in missing data proportions. Their results further show that better performance would be obtained from applying imputation techniques.

Another comparative study of MDTs in the context of software cost estimation is carried out by [15]. The simulation study is carried out using 176 projects. The four missing data techniques were LD, MMSI, similar response pattern imputation (SRPI) and full information maximum likelihood (FIML). Their results showed FIML performing well for missing completely at random (MCAR) data. Also, LD, MMSI and SRPI were shown to yield biased results

ICSE'06, May 20–28, 2006, Shanghai, China.
ACM 1-59593-085-X/06/0005.

for other missing data mechanisms other than MCAR. Their recommendations were to use FIML if one had enough data and to use MMSI or SRPI if one needed more data. A combination of LD with a regression model was recommended for small datasets where FIML cannot be used. However, LD should only be used for MCAR data.

The performance of *k*-nearest neighbour single imputation (*k*NNSI) and sample mean imputation (SMI) was analysed by [5] using two small industrial datasets. Their results showed both methods yielding good results with *k*NNSI providing a more robust and sensitive method for missing value estimation than SMI.

[19] evaluate *k*NNSI and class mean imputation (CMI) for different patterns and mechanisms of missing data. Their results show *k*NNSI slightly outperforming CMI with the missing data mechanisms having no impact on either of the two imputation methods.

The *k*NNSI method is evaluated by [11] using a likert dataset with 56 cases in the software engineering context. His results not only showed that imputing missing likert data using the *k*-nearest neighbour method was feasible they showed that the outcome of the imputation depends on the number of complete instances more than the proportion of missing data. Their results further showed the importance of choosing an appropriate *k* value when using such a technique.

The use of multinomial logistic regression imputation (MLRI) for handling categorical attribute values on a dataset on 166 projects of the ISBSG multi-organizational software database was proposed by [18]. Their proposed procedure is compared with LD, MMSI, expectation maximization single imputation (EMSI) and RBSI. Their results show LD and MMSI as efficient when the percentage of missing values is small while RBSI and MLRI is shown to outperform LD and MEI as the amount of missing values increases. Overall, MLRI gives the best results, especially for MCAR and informatively missing (IM) data. For missing at random (MAR) data, MLRI compares favourably with RBSI.

In [21], an ensemble MDTs approach which combines *k*NNSI and MI is proposed. The goal of this approach is to improve predictive accuracy and it is evaluated on two complete industrial datasets. Missing data are simulated for three different proportions, two patterns, and three mechanisms of missing data. Empirical results show that an ensemble generated using this new approach yield results that are superior in predictive accuracy to individual MDTs.

In summary, we think that the available prior research supports a lot of the questions we are targeting with our study. First, no MDT was found to be uniformly superior to the others. However, the performance of each technique differs with increases in the amount of missing data. Also, despite the scarcity of software data and the fact that the LD procedure involves an efficiency cost due to the elimination of a large amount of valuable data, most software engineering researchers have used it due to its simplicity and ease of use. Among imputation techniques, the results are not so clear. However, *k*NNSI achieves superior performance to other single imputation methods such as MMSI and CCSI. In addition, maximum likelihood procedures represent a superior approach to missing data. ML methods appear to achieve higher accuracy than traditional statistical approaches because of their complicated processing even though they take much more time in processing than statistical methods do. Also, multiple imputation, which overcomes limitations of single imputation seem not to have been widely adopted by researchers even though it has been shown to be flexible and software for creating multiple imputations is available. Finally, results from previous studies suggest that results achieved using simulated data are very sensitive to the MAR assumption. Hence, if there is a reason to believe that if the MAR assumption does not hold, alternative methods should be used.

This paper seeks to address the deficit of methodological and technical guidance in software engineering research for handling incomplete data by familiarizing software engineers with newer techniques, use simulation to evaluate the predictive accuracy associated with different techniques, and encouraging them to apply these techniques in their work. The paper further investigates how an ensemble of MDTs could be exploited to improve prediction accuracy generated by a randomized decision tree induction method [2, 17]. Randomization is introduced by using random samples of the training data as in bagging [3] or boosting [9] and running a conventional tree-building algorithm. We hope these findings encourage researchers to examine the use of ensemble missing data techniques for predicting software effort.

The rest of the paper is organized as follows; after investigating the impact of missing values on predictive accuracy and how an ensemble of MDTs could be utilized to improve prediction accuracy in section 3 we present preliminary empirical results of the proposed approach. Discussion of these results and direction for future research are also presented in this section.

2. SIMULATION STUDY AND PRELIMINARY RESULTS

2.1 Simulation study

One of the objectives of this paper is to investigate the robustness and accuracy of seven different methods for tolerating incomplete data using tree-based models. The results enable us to explore ways of improving prediction accuracy using an ensemble of MDTs created using randomized decision trees. A combination of small and large datasets, with a mixture of both nominal and numerical attribute variables, is used for these tasks. For the datasets with missing values, instances with missing values were removed before starting the experiments. The reduced datasets are summarized in Table 1.

The main reason for using datasets with no missing values is to have total control over the missing data in each dataset. 5-fold cross validation is used for the experiments by splitting each dataset randomly into five parts of equal size. Each part is alternatively selected as a test set and the remaining four sets from the training set for the learning algorithm. In other words, 80% of the data is used for training with the remaining 20% used for testing. Prediction accuracies on test sets gathered from all parts are then averaged to give a performance measure of the algorithm. In addition, in this paper we are dealing with a classification-type of problem that predicts values of a categorical dependent attribute from one or more continuous or categorical attributes the values of the dependent attribute were converted into intervals through a process called discretization [7].

Table 1. Datasets used for the experiments

Dataset	Instances	Attributes	
		Numerical	Categorical
Kemerer	18	4	2
Bank	18	2	7
Test equipment	16	17	4
DSI	26	5	0
Moser	32	1	1
Desharnais	76	3	6
Experience	95	1	5
ISBSG version 7	166	2	7

The simulation study concentrates on performing experimental analysis of MDTs which range from simple statistical algorithms to machine learning algorithms. The seven MDTs (LD, EMSI, kNNSI, MMSI, EMMI, FC, and SVS) are compared by artificially simulating three different missing data proportions; two patterns of missing data; and three mechanisms of missing data. Then, an ensemble of the two missing data methods which achieved the highest accuracy rates in the previous experiments is used to obtain a significant improvement in prediction accuracy. A 4-way repeated measures design is employed to analyze the data with each effect tested against its interaction with datasets at the 1% level of significance. The 1 % level is used because of the many number of effects.

2.2 Experimental Results

The results are presented in two parts. The first part compares the performance of seven methods for classifying incomplete vectors using decision trees. The second part evaluates the effectiveness of the proposed ensemble MDTs approach that utilizes randomized decision trees.

Main Effects:

All the main effects were found to be significant at the 1% level of significance (F149.7, df = 7 for missing data techniques; F=478.3, df = 1 for number of attributes with missing values; F=881.8, df=2 for missing data proportions; F=3888.2, df=2 for missing data mechanisms; $p<0.01$ for each main effect).

Figure 1 plots the overall excess error rates for eight MDTs and an ensemble of MDTs. From the results it follows that from the seven MDTs, EMMI achieves the highest accuracy rates, followed by FC, EMSI, SVS, *k*NNSI and MMSI, respectively. The worst performance is by LD. There appears to be no significant difference among the single imputation techniques at the 1% level of significance. The randomized ensemble of EMMI and FC (which we shall now call FCMI) achieves the highest accuracy rates compared to individual MDTs. The difference in error rates between FCMI, on the one hand, and EMMI and FC (individually), on the other hand, is significantly different at the 1% level of significance.

Interaction Effects:

Two interaction effects were found to be significant (F=2.719, df=14 for missing data techniques and missing data mechanism; F=14.8, df=2 for number of attributes with missing values and missing data mechanisms; $p<0.01$ for each interaction effect). Figure 2 shows all techniques achieving bigger error rates when dealing with IM data compared with either MCAR or MAR data. In addition, some techniques appear to be severely impacted by the different missing mechanisms than others.

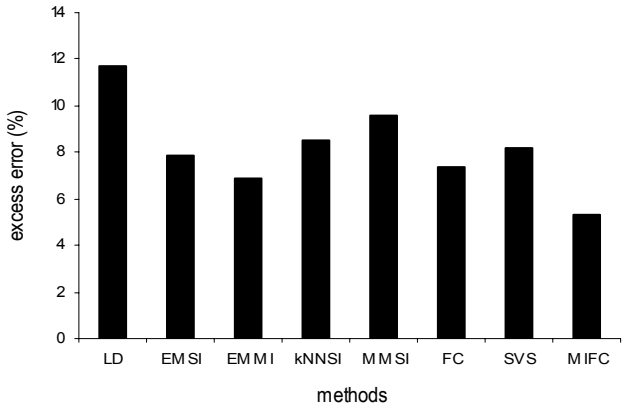

Fig. 1. Overall means for current and ensemble missing data techniques

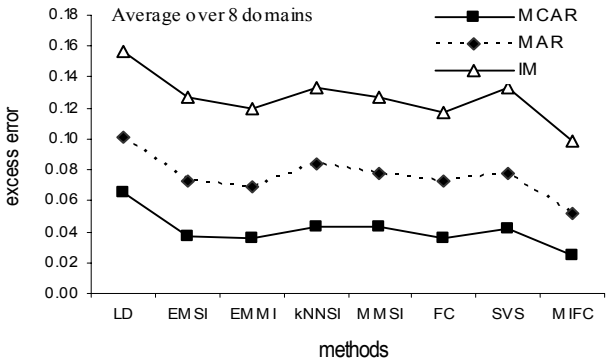

Fig. 2. Interaction between missing data techniques and missing data mechanisms

3. DISCUSSION

The analysis examined the accuracy of several commonly used methods to handle incomplete data on 8 public domain problems, and then used an ensemble strategy of two MDTs to improve software prediction accuracy. To date there has been a few studies examining the impact of MDTs for different proportions, patterns of and mechanisms of missing data on resulting DTs. We are also not aware of any existing study that has looked at an ensemble of MDTs in the context of software prediction.

The analysis demonstrated the strengths and limitations of the procedures as reported in the literature. For example, the findings from the current analyses suggest differences in performance between methods with increases in proportion of missing data. The results further suggest maximum likelihood procedures as a superior approach to missing data with EMMI having on average the best prediction accuracy. A major strength of using EMMI is that it restores the variability of missing values. This means that the same analyses must be run multiple times, with multiple datasets.

However, there are software programs available free of charge (for example, at *http://methodology.psu.edu/*) that reduces the inconvenience of having to run analyses multiple times as well as the possibility of human errors in terms of analyzing multiple datasets and re-entering parameter estimates by hand into multiple imputation inference. The poor performance of *k*NNSI was not surprising as most of our datasets are small with many attributes, a common scenario in software engineering. In fact, when using small data sets, one runs the risk of using the same donor many times, thus resulting in a loss of precision in the imputed value. LD of missing data, a technique widely used by software engineers due to its simplicity and ease of use, was the least accurate approach. However, the accessibility and ease of use of model-based techniques such as EMMI provides further argument that it is no longer justifiable to continue using older techniques like LD and MMSI. The results from our analysis also show missing values as having more impact when they are uniformly distributed among all the attributes compared with when they are on a single attribute. The impact of missing values was more severe on predictive accuracy rates for IM data compared with MCAR or MAR data.

In this paper we have also proposed an approach to the generation of MDTs ensemble where randomization is introduced in the DT induction through the use of samples. Our early experimental results using public domain datasets show that there is a promising approach, both in terms of accuracy and computational cost. However, much remains to be done. We want to improve the performance of the ensemble by exploring and considering other sampling strategies (like stratified sampling). Further, we are interested in seeing if the results would be improved by using non-tree machine learning algorithms in the ensemble. In addition, we plan to conduct studies using much more balanced types of datasets to see if the results carry over to these datasets as well, especially larger datasets. No optimization criterion was used for selecting the best combination of MDTs for the ensemble. Instead the methods selected were the ones which achieved higher accuracy rates in our first experiment. Future research that uses an optimization criterion for selecting every combination of the MDTs is recommended**.**

4. ACKNOWLEDGMENTS

This work was funded by the UK Engineering and Physical Sciences Research Council under grant GR/S55347. The comments and suggestions from Allan White improved this paper.

5. REFERENCES

[1] Berk, K. (1987). "Computing incomplete repeated measures", Biometrics, **43**, 269-291.

[2] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*, Wadsworth.

[3] Breiman, L. (1996). "Bagging predictors", *Machine Learning*, **26** (2), 123-140.

[4] Brown, C.H. (1983). "Asymptotic comparison of missing data procedures for estimating factor loadings", *Psychometrics*, **48**, 269-291.

[5] Cartwright, M., Shepperd, M.J., and Song, Q. (2003). "Dealing with Missing Software Project Data", In *Proc. of the 9th Int. Symp .on Software Metrics 2003*, 154-165.

[6] Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., and Freeman, D. (1988). "Bayesian Classification", In *AAAI*, Morgan Kaufmann Publishers: San Meteo, CA, 607-611.

[7] Dougherty, J., Kohavi, R., and Sahami, M. (1995). Supervised and Unsupervised Discretization of Continuous Features. In *Proc. of the 12th International Conference on Machine Learning*. Morgan Kauffmann, Los Altos, CA

[8] El-Emam, K. and Birk, A. (1999). "Validating the ISO/IEC 15504 Measures of Software Development Process Capability**",** *Journal of Syst. and Software*, **51** (2), 119-149.

[9] Freund, Y. and Schapire, R. (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the 13th International Conference*, 148-156.

[10] Johnson, E.G. (1989). "Considerations and techniques for the analysis of NAEP data", *Journal of Educational Statistics.*, **14**, 303-334.

[11] Jönsson, P. and Wohlin, C. (2004). An Evaluation of k-Nearest Neighbour Imputation Using Likert Data. In *Proc. of the 10th Int. Symp. on Software Metrics*, 108-118.

[12] Kaufman, C.J. (1988). "The application of logical imputation to household measurement", *Journal of the Market Research Society*, **30**, 453-466.

[13] Lakshminarayan, K., Harp, S.A., Samad, T. (1999). "Imputation of Missing Data in Industrial Databases", *Applied Intelligence*, **11**, 259-275.

[14] Musil, C.M., Warner, C.B., Yobas, P.K., and Jones, S.L. (2002). "A Comparison of Imputation Techniques for handling Missing Data", *Western Journal of Nursing Research*, **24** (5), 815-829.

[15] Myrtveit, I., Stensrud, E., and Olsson, U. (2001). "Analyzing Data Sets with Missing Data: An Empirical Evaluation of Imputation Methods and Likelihood-Based Methods", *IEEE Transaction on Software Engineering*, **27** (11), 1999-1013.

[16] Poirier, D.J. and Rudd, P.A. (1983). "Diagnostic testing in missing data models", *International. Economic Review*, **24**, 537-546.

[17] Quinlan, J.R. (1993). *C.4.5: Programs for machine learning.* Los Altos, California: Morgan Kauffman Publishers, INC.

[18] Sentas, P., Lefteris, A., and Stamelos, I. (2004). "Multiple Logistic Regression as Imputation method Applied on Software Effort prediction", In *Proc. of the 10th Int. Symp. on Software Metrics,* Chicago, 14-16 September 2004.

[19] Song, Q. and Shepperd, M. (2004). "A Short Note on Safest Default Missingness Mechanism Assumptions", In *Empirical Software Engineering*. (accepted in 2004).

[20] Strike, K., El-Emam, K.E., Madhavji, N. (2001). "Software Cost Estimation with Incomplete Data", *IEEE Transaction on Software Engineering*, **27** (10), 890-908.

[21] Twala, B. and Cartwright, M. (2005). "Ensemble Imputation Methods for Missing Software Engineering Data", In *Proc. of the 11th Int. Symp. on Software Metrics*, Como, Italy, September 19-22, 2005.