Empirical Software Engineering, 10, 235–243, 2005.

A Short Note on Safest Default Missingness Mechanism Assumptions

QINBAO SONG qsong@bmth.ac.uk
Empirical Software Engineering Research Group, School of Design, Engineering and Computing, Bournemouth University, UK

MARTIN SHEPPERD mshepper@bmth.ac.uk
Empirical Software Engineering Research Group, School of Design, Engineering and Computing, Bournemouth University, UK

MICHELLE CARTWRIGHT mcartwri@bmth.ac.uk
Empirical Software Engineering Research Group, School of Design, Engineering and Computing, Bournemouth University, UK

Editor: James Miller

Abstract. A very common problem when building software engineering models is dealing with missing data. To address this there exist a range of imputation techniques. However, selecting the appropriate imputation technique can also be a difficult problem. One reason for this is that these techniques make assumptions about the underlying missingness mechanism, that is how the missing values are distributed within the data set. It is compounded by the fact that, for small data sets, it may be very difficult to determine what is the missingness mechanism. This means there is a danger of using an inappropriate imputation technique. Therefore, it is necessary to determine what is the safest default assumption about the missingness mechanism for imputation techniques when dealing with small data sets. We examine experimentally, two simple and commonly used techniques: Class Mean Imputation (CMI) and k Nearest Neighbors (k-NN) coupled with two missingness mechanisms: missing completely at random (MCAR) and missing at random (MAR). We draw two conclusions. First, that for our analysis CMI is the preferred technique since it is more accurate. Second, and more importantly, the impact of missingness mechanism on imputation accuracy is not statistically significant. This is a useful finding since it suggests that even for small data sets we can reasonably make a weaker assumption that the missingness mechanism is MAR. Thus both imputation techniques have practical application for small software engineering data sets with missing values.

Keywords: Software effort prediction, missing data, data imputation, missingness mechanism.

1. Introduction

Missing data imputation is an important issue in software effort prediction, since missing values are a commonplace problem. Several imputation methods (Strike et al., 2001; Myrtveit et al., 2001; Schafer and Olsen, 1998) have been used for this purpose. But, each method suffers from various limitations (Strike et al., 2001; Myrtveit et al., 2001), not least that these methods make assumptions about the underlying missingness mechanism, that is, how the missing values are distributed within the data set. The problem is compounded by the fact that for small data sets it may be unclear what the missingness mechanism is. So the aim of this short note is to assess the impact of different missingness mechanisms—through experimentation on a real world software

project effort data set—to help assess what imputation method can be used under what circumstances.

In general, the missingness mechanism concerns whether the missingness is related to the study variables or not. This is extremely significant as it determines how difficult it may be to impute missing values and at the same time how risky it is to ignore them. Little and Rubin (1987) divide these mechanisms into three classes: Missing Completely at Random (MCAR), Missing at Random (MAR), and Non-ignorable (NI).

Suppose $\mathbf{Z}$ is a data matrix that includes observed and missing data, let $\mathbf{Z_{obs}}$ be the set of observed values of $\mathbf{Z}$, let $\mathbf{Z_{mis}}$ be the set of missing values of $\mathbf{Z}$ and let $\mathbf{R}$ be the missing data indicator matrix, i be the ith case and j the jth feature:

$$\mathbf{R_{ij}} = \begin{cases} 1 & \text{if } \mathbf{Z_{ij}} \text{ is missing} \\ 0 & \text{if } \mathbf{Z_{ij}} \text{ is observed} \end{cases}$$

Missing Completely At Random (MCAR): indicates that the missingness is unrelated to the values of any variables, whether missing or observed, so:

$$p(\mathbf{R}|\mathbf{Z}) = p(\mathbf{R}) \text{ for all } \mathbf{Z}$$

MCAR is an extreme condition and from an analysts point of view, ideal. Generally you can test whether MCAR conditions can be met by showing there is no difference between the distribution of the observed data of the observed cases and missing cases (Little, 1988). Unfortunately this is hard when there are few cases as there can be a problem with Type I errors.

Non-Ignorable (NI): is at the opposite end of the spectrum. It means that the missingness is related to the missing values. It is non-random and is not predictable from any one variable in the data set, that is:

$$p(\mathbf{R}|\mathbf{Z}) \neq p(\mathbf{R}) \text{ for all } \mathbf{Z},\ p(\mathbf{R}|\mathbf{Z}) \text{ depends on } \mathbf{Z_{mis}}$$

NI is the worst case since, as the name implies, the problem cannot be avoided by a deletion technique nor are imputation techniques in general effective unless the analyst has some model of the cause of missingness. This is best illustrated by an example. Suppose software engineers are less likely to report high defect rates than low rates, perhaps for reasons of politics. Merely to ignore the incomplete values leads to a biased sample and an over optimistic view of defects. On the other hand imputation techniques do not work well either since they attempt to exploit known values and as we have already observed this is a biased sample. Unless one has some understanding of the process and can construct explanatory models there is little that can effectively be done with NI missingness.

Missing At Random (MAR): lies between these two extremes. It requires that the cause of the missing data is unrelated to the missing values, but may be related to the observed values of other variables, that is:

$$p(\mathbf{R}|\mathbf{Z}) = p(\mathbf{R}|\mathbf{Z_{obs}}) \text{ for all } \mathbf{Z_{mis}}$$

Most missing data methods assume MAR. It is less restrictive than MCAR because MCAR is a special case of MAR. MAR and MCAR are both said to be ignorable missing data mechanisms.

Strike, Emam, and Madhavji's (Strike et al., 2001) experiment shows that the mean imputation and Hot-deck methods tend to perform slightly worse under the NI missingness mechanism compared with MCAR and MAR. But what is the situation with small data sets? Specially, what is the safest default assumption about the missingness mechanism for some imputation methods? We try to answer this question in this short paper.

2. Method

The context for this investigation is software project data sets that contain missing values. Frequently such data sets contain very few cases—sometimes fewer than 20—which is problematic for a number of reasons. It is undesirable to sacrifice scarce data by using an ignoring technique such as case deletion, thus imputation is desirable. However, with few cases it can be difficult to obtain statistically significant results when formally analysing the data to determine the missingness mechanism. Without knowing the missingness mechanism an assumption must be made and an imputation technique might be used inappropriately.

In our analysis we wished to consider the following three aspects:

- Missingness mechanisms
  MCAR: random removal of values for the chosen feature or features.
  MAR: based on the size of the project. The bigger the project the greater the probability of missing values (Strike et al., 2001).

- Missingness patterns
  Univariate missing data in a continuous feature and bivariate missing data in continuous features.

- Missing data percentages
  For each missing data pattern of each missing data mechanism, four different percentages of missing values, 10%, 15%, 20%, and 30% were simulated.

After that, we chose two different methods, Class Mean Imputation (CMI) (Reinsdorf et al., 1996) and k Nearest Neighbors (k-NN) (Fix and Hodges, 1952), as the missing data imputation techniques because of their stability, simplicity, and frequency of use.

The data set used in this study is the International Software Benchmarking Standards Group (ISBSG) v7 database. The ISBSG has a large fraction of missing data, exceeding 40% for some features (Angelis et al., 2001; Jeffery et al., 2001), so in order to simulate various missing patterns and compare the accuracy of different imputation techniques, we cleaned the ISBSG database as follows.

First, we used an information theoretic method (Quinlan, 1996), which can be applied to data sets containing a mixture of nominal and continuous features, to determine key features and reduce the dimensions of the data set to six categorical and three continuous features.

Key feature selection is the process of identifying and removing as much irrelevant and redundant information as possible. It is a problem which occurs in many contexts, Kirsopp and Shepperd (Kirsopp et al., 2002) explored this problem for purpose of software project effort prediction. If a feature containing missing data was not a key feature for predicting software project effort, it was deleted, otherwise, delete the corresponding case. In this way we obtained a new data set that comprised of 363 complete cases named ISBSG-363.

In order to validate how the two different techniques, CMI and *k*-NN, can be used to impute missing data in small data sets, and to help ensure the results are consistent across multiple samples, we randomly extracted 100 and 50 cases from data set ISBSG-363, and named them ISBSG-100 and ISBSG-50 respectively.

Next we simulated various missing data conditions by sampling cases from the data set and removing values according to various pre-determined missingness mechanisms. This meant we were able to assess the performance of the imputation technique by comparing the imputed and true values.

This simulation approach was applied to data sets ISBSG-100 and data set ISBSG-50 respectively to provide the data sets with different missingness mechanisms, different missingness patterns and different missing data percentages. From this simulation, we generated a total of 160 data sets, (i.e. 80 data sets with 100 cases and 80 data sets with 50 cases). For each group, we had two missingness mechanisms (MAR and MCAR), two missingness patterns (one and two continuous features with missing values) and four levels of missing data values (10%, 15%, 20%, 30%). And for each specific missingness mechanism and each specific missing data pattern and each specific missing data percentage, we sampled five data sets in order to gain confidence in our results. Next, the two different techniques, CMI and *k*-NN, were applied and their performance evaluated in comparison to real or known values.

Finally, we use Mean Magnitude of Relative Error (MMRE) (Conte et al., 1986) as the accuracy measure of different imputation methods. MMRE is a widely used indicator of prediction accuracy in software engineering domain and is defined as:

$$MMRE = \frac{100}{n} \sum_{i=1}^{i=n} \frac{|x_i - \hat{x}_i|}{x_i}$$

where n is the number of predictions $\hat{x}$ of x.

3. Results

Table 1 summarises our results. It shows the absolute differences of average MMREs between MAR and MCAR for both groups of data sets with one or two continuous

Table 1. Absolute difference of MMREs between MAR and MCAR for both CMI and *k*-NN methods with ISBSG-100 and ISBSG-50 data sets.

			Proportion of missing values			
Data set	Missingness pattern	Technique	10%	15%	20%	30%
ISBSG-100	One continuous feature	CMI	1.55	0.59	0.52	0.89
	With missing values	*k*-NN	3.43	2.12	2.12	0.33
	Two continuous features	CMI	0.83	2.94	1.54	0.53
	With missing values	*k*-NN	2.54	2.12	1.35	5.25
ISBSG-50	One continuous feature	CMI	2.14	1.17	0.82	1.01
	With missing values	*k*-NN	1.18	0.12	1.84	0.23
	Two continuous features	CMI	0.54	2.38	1.15	0.97
	With missing values	*k*-NN	5.61	4.11	5.90	0.61

features with missing values for CMI and *k*-NN method. We see that the MMRE differences between MAR and MCAR missingness mechanisms are extremely small (the greatest difference is under 6%) which is indicates that we don't need to consider the effect of missingness mechanism in terms of differentiating between MCAR and MAR. This pattern seems to be hold, independently of the data set size (50 or 100 cases) or the proportion of missing values (10–30%).

Since the MMRE statistic describes the centre, and each analysis was repeated five times with different samples from ISBSG-363, we constructed confidence intervals based on the t-distribution using a sample-based standard deviation estimate. For the t-interval for individual μ's, the individual confidence is 90%. Table 2 presents the results that indicate relatively narrow confidence limits, in other words relatively little variability due to sampling.

For the purpose of more formally determining the significance of the results, we used a Mann–Whitney test to compare sample medians of the MMRE values. This is a non-parametric test that doesn't assume that the data follows a Gaussian distribution. We had two alternate hypotheses:

- class mean imputation is more accurate when the missingness mechanism is MCAR than MAR

Table 2. Confidence intervals for CMI and *k*-NN with ISBSG-100 and ISBSG-50 under MAR and MCAR.

Missingness mechanism	Technique	Bounds
MAR	CMI	$67.58 < \mu < 73.13$
	k-NN	$76.09 < \mu < 81.49$
MCAR	CMI	$67.55 < \mu < 71.30$
	k-NN	$73.94 < \mu < 78.12$
MAR+	CMI	$68.13 < \mu < 71.59$
MCAR	*k*-NN	$75.46 < \mu < 78.16$

Table 3. Mann–Whitney test of imputation accuracy differences between missingness mechanisms for *k*-NN and CMI methods.

Imputation technique	Missingness mechanism	n	Median	*p*
CMI	MCAR	80	69.6	0.73
	MAR	80	71.7	
k-NN	MCAR	80	73.9	0.06
	MAR	80	78.0	

- *k*-NN imputation is more accurate when the missingness mechanism is MCAR than MAR

The null hypotheses are that there is no difference with ($\alpha = 0.05$). From Table 3 we see that the alternate hypotheses are rejected, strongly in the case of CMI and weakly for *k*-NN.

So we can conclude that the impact of the missingness mechanism on the relative accuracy of the two imputation techniques is not statistically significant, thus we can safely assume MAR when we impute missing software project data using CMI, but possibly with the need for more caution for *k*-NN based imputation.

Table 4 contains the results of Mann–Whitney tests comparing sample medians of the MMRE values for the imputation errors of the *k*-NN and CMI methods. Here we have three alternate hypotheses:

- class mean imputation is more accurate than *k*-NN when the missingness mechanism is MAR

- class mean imputation is more accurate than *k*-NN when the missingness mechanism is MCAR

- class mean imputation is more accurate than *k*-NN

We see that all three alternate hypotheses are accepted. This suggests that overall CMI performs a little better than *k*-NN.

Table 4. Mann–Whitney test of imputation accuracy differences between *k*-NN and CMI.

Missingness mechanism	Technique	n	Median	*p*
MAR	*k*-NN	80	74.70	0.0009
	CMI	80	71.50	
MCAR	*k*-NN	80	72.56	0.0001
	CMI	80	68.33	
MCAR	*k*-NN	160	73.40	<0.0001
+MAR	CMI	160	69.64	

We also explored the impact of missing data percentages, of between 10% and 30%, on the relative accuracy of the two methods under MCAR and MAR missingness mechanisms. However, here we found no statistically significant differences.

4. Conclusions

In this study we have assessed the impact of different missingness mechanisms on the imputation accuracy of two popular imputation techniques, namely Class Mean Imputation (CMI) and *k* Nearest Neighbours. This is to help determine which imputation technique can be used under what circumstances. The analysis used the real world software project data set ISBSG. In particular we were interested in examining the effect on very small data sets, so we randomly sampled 50 and 100 cases.

Our results revealed that for both CMI and *k*-NN methods, in the context of software project effort prediction, the effect of the missingness mechanism discounting NI, was not significant for either imputation technique. We were also able to show that CMI performed better, i.e. imputed better values than *k*-NN, irrespective of the type of missingness mechanism. Therefore we conclude it may be possible to assume that the missingness mechanism is MCAR when dealing with missing data values. Also our recommendation is to prefer CMI when dealing with small software project data sets.

Acknowledgments

The authors would like to thank the International Software Benchmarking Standards Group (ISBSG) for providing the data set for this analysis. The authors also thank Dr. Colin Kirsopp and anonymous reviewers for their helpful comments. This work was partly funded by the Engineering and Physical Sciences Research Council (UK) under grant GR/S55347.

References

Angelis, L., Stamelos, I., and Morisio, M. 2001. Building a software cost estimation model based on categorical data. *Proceedings Seventh International Software Metrics Symposium (METRICS 2001)*, pp. 4–15.

Conte, S., Dunsmore, H., and Shen, V. Y. 1986. *Software Engineering Metrics and Models*. Menlo Park, CA: Benjamin Cummings.

Fix, E., and Hodges, J. L. 1952. Discriminatory analysis: Nonparametric discrimination: Small sample performance, Technical Report Project 21-49-004, Report Number 11, USAF School of Aviation Medicine, Randolf Field, Texas.

Jeffery, R., Ruhe, M., and Wieczorek, I. 2001. Using public domain metrics to estimate software development effort. *Proceedings Seventh International Software Metrics Symposium (METRICS 2001)*, pp. 16–27.

Kirsopp, C., Shepperd, M. J., and Hart, J. 2002. *Search Heuristics, Cased-Based Reasoning and Software Project Effort Prediction*. GECCO 2002: Genetic and Evolutionary Computation Conf., New York, AAAI.

Little, R. J. A. 1988. A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association* 83(404): 1198–1202.

Little, R. J. A., and Rubin, D. B. 1987. *Statistical Analysis with Missing Data*. New York: John Wiley & Sons.

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 Transactions on Software Engineering* 27(11): 999–1013.

Quinlan, J. R., 1996. Learning decision tree classifiers. *ACM Computing Surveys* 28: 71–72.

Reinsdorf, M. B., Liegey, P., and Stewart, K. J. 1996. New Ways of Handling Quality Change in the U.S. Consumer Price Index. *Bureau of Labor Statistics working paper no. 276*, USA.

Schafer, J., and Olsen, M. 1998. Multiple imputation for multivariate missing-data problems: A data analyst's perspective. *Multivariate Behavioural Research* 33: 545–571.

Strike, K., Emam, K. E., and Madhavji, N. 2001. Software cost estimation with incomplete data. *IEEE Transactions on Software Engineering* 27(10): 890–908.

Qinbao Song received a Ph.D. in computer science from the Xi'an Jiaotong University, P. R. China, in 2001. He is currently a Research Fellow with the Empirical Software Engineering Research Group at the Bournemouth University, UK. He has published 46 refereed papers in the area of data mining, machine learning, and software engineering. His research interests include machine learning, data mining & knowledge engineering, and empirical software engineering.

Martin Shepperd received a Ph.D. in computer science from the Open University, UK in 1991. He is professor of software engineering at Bournemouth University, UK. He has published more than 90 refereed papers and three books in the area of empirical software engineering, machine learning and statistics. He is editor of the journal Information & Software Technology and was Associate Editor of IEEE Transactions on Software Engineering (2000–4).

Michelle Cartwright is a senior lecturer in the Empirical Software Engineering Group, in the school of Design, Engineering and Computing, Bournemouth University. She received a BSc degree (honours) in Computer Science from the University of Wolverhampton, and a Ph.D. from Bournemouth University. Her research interests include software metrics for object-oriented systems, empirical software engineering, data imputation and software project management issues.