Empirical Evaluation of Missing Data Techniques for Effort Estimation

Koichi Tamura[1], Takeshi Kakimoto[2], Koji Toda[1], Masateru Tsunoda[1],
Akito Monden[1], Ken-ichi Matsumoto[1]

[1] *Graduate School of Information Science, Nara Institute of Science and Technology*
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
{koichi-t, koji-to, masate-t, akito-m, matumoto}@is.naist.jp

[2] *Graduate School of Information Science and Technology, Osaka University*
1-3 Machikaneyama, Toyonaka, Osaka, 560-8531, Japan
kakimoto@ist.osaka-u.ac.jp

Abstract

Multivariate regression models have been commonly used to estimate the software development effort to assist project planning and/or management. These models require a complete data set that has no missing values for model construction. The complete data set is usually built either by using imputation methods or by deleting projects and/or metrics that have missing values (we call this RC deletion). However, it is unclear which method is the most suitable for the effort estimation. In this paper, using the ISBSG data set of 706 projects (containing 47% missing values) collected from several companies, we applied four imputation methods (mean imputation, pairwise deletion, k-NN method and CF method) and RC deletion to build regression models. Then, using a data set of 143 projects (with no missing values), we evaluated the estimation performance of models after applying each imputation or the RC deletion. The result showed that the similarity-based imputation method (k-NN method and CF method) showed better performance than other methods in terms of MdMAE, MdMRE, MdMER and Pred(25).

1. Introduction

In a software development project, software cost estimation is necessary for management of schedule and resources. So far, various quantitative estimation methods using a historical project data set have been proposed and used [3] [12] [14]. Among these methods, the regression model has been most widely used for its convenience [1] [9] [15].

One of the practical problems in using estimation methods is that the historical project data usually contain substantial numbers of missing values [1] [11]. One reason is that different divisions in an organization might have different policies on data collection, i.e. one project collects a particular metric while other projects do not. Even if the organization has a unified policy, not all metrics are collected in each project due to the pressing development schedule. However, not only regression models but also many other estimation models need a data set with no missing values to build models.

One approach to solve this problem is to delete the metrics and projects with missing values from the data set (we call this *RC deletion*). This method is easy to use; however, deletion might remove useful information for effort estimation. In addition, there are unclear trade-offs between deletion of metrics and deletion of projects. Another commonly used approach is using imputation methods [6] [8] [11]. Imputation does not reduce the information; however, it might introduce noise to the data set. Yet another approach is pairwise deletion, which is applicable to regression models [15]. While all these methods are considered useful, it is unclear which is the best. It is important for an engineer to know which is better: (a) to avoid losing information despite introducing noise or (b) to avoid noise despite losing information.

Our primary goal is to clarify which missing data technique shows the best performance for building the regression model. In this paper, using a data set of 706 projects (containing 47% missing values) collected by the International Software Benchmarking Standards Group (ISBSG) [5], we applied four imputation methods (mean imputation, pairwise deletion, *k*-NN method and CF method) and RC deletion to build estimation models using stepwise multiple regression analysis [2]. Then, using a data set of 143 projects (with no missing values), also from the ISBSG data set, we evaluated the estimation

performance of models after applying each imputation or the RC deletion. The CF method is based on *collaborative filtering* and applying it to missing data imputation is a first-time approach.

2. Imputation and deletion methods

In this paper, the following methods were applied to the data set before building estimation models.

2.1. Mean imputation

This method fills each missing value with the mean of observed values [8] [11] [15].

2.2. Pairwise deletion

This method is particularly associated with the regression analysis [8] [15]. While calculating a correlation matrix to build a regression model, a correlation between each pair of variables is calculated from all cases (projects) that have non-missing value on those two variables. This method is widely used in statistical analysis tools such as SPSS.

2.3. Similarity-based imputation (*k*-NN method)

This method uses existing metrics values of *k* most similar projects to fill missing values of the target project. The degree of similarity between two projects is computed by *Euclidean distance* [1] [6].

2.4. Similarity-based imputation (CF method)

This method is an alternative similarity-based imputation based on *collaborative filtering* [12]. Although this method is used for cost estimation, we newly apply it to missing data imputation. The method uses *cosine similarity* to compute the similarity instead of Euclidean distance. The three-step procedure of the CF method is described below.

Step 1 (normalization of metrics): Since each metric has a different value range, this first step normalizes values of metrics so that the value range becomes [0, 1]. Here, we denote that p_i is *i*-th project, m_j is *j*-th metric, and $v_{i,j}$ is the value of metric m_j observed in project p_i. The normalized value $v'_{i,j}$ of $v_{i,j}$ (of project p_i) is calculated by the following equation:

$$v'_{i,j} = \frac{v_{i,j} - \min(P_j)}{\max(P_j) - \min(P_j)} \quad (1)$$

where P_j denotes a set of projects in which the value of metric m_j was observed (collected), $max(P_j)$ and $min(P_j)$ denote the maximum and minimum value in $\{v_{x,j} \mid p_x \in P_j\}$ respectively.

Step 2 (computation of similarity between projects): In this step, similarity $sim(p_a, p_i)$ between the target project p_a and other projects p_i is computed. Formally, we can define the $sim(p_a, p_i)$ between the target project p_a and other projects p_i as:

$$sim(p_a, p_i) = \frac{\sum_{j \in M_a \cap M_i} (v'_{a,j} - md(m'_j)) \times (v'_{i,j} - md(m'_j))}{\sqrt{\sum_{j \in M_a \cap M_i} (v'_{a,j} - md(m'_j))^2} \sqrt{\sum_{j \in M_a \cap M_i} (v'_{i,j} - md(m'_j))^2}} \quad (2)$$

where M_a and M_i denote a set of metrics observed in project p_a and p_i respectively, m'_j denotes the normalized value of m_j, and $md(m'_j)$ denotes the median of m'_j.

The metrics that are higher than $md(m'_j)$ show positive values and the metrics that are lower than $md(m'_j)$ show negative values by subtracting $md(m'_j)$. The value range of $sim(p_a, p_i)$ is [-1, 1] for this computation. Note that $sim(p_a, p_i)$ shows low or negative value (i.e., the computed similarity shows low value) if the difference of metrics between p_a and p_i is great.

Step 3 (computation of estimation): This step calculates an estimated value $\hat{v}_{a,b}$ of the metric m_b on the target project p_a using $sim(p_a, p_i)$ calculated in the previous step. The estimated value is computed as the sum of the metrics' values given by the other projects similar to p_a. Each value is weighted by the corresponding $amplifier(p_a, p_i)$ and $sim(p_a, p_i)$ between p_a and p_i. Formally, we can define the estimated value as:

$$\hat{v}_{a,b} = \frac{\sum_{i \in k-nearestPrjects} (v_{i,b} \times amplifier(p_a, p_i) \times sim(p_a, p_i))}{\sum_{i \in k-nearestPrjects} sim(p_a, p_i)} \quad (3)$$

where *k-nearestProjects* denotes a set of *k* projects (called *neighborhoods*) that have the highest similarity with p_a. The neighborhoods must have m_j as an observed metric. Generally, the neighborhood size *k* affects the estimation accuracy (This point applies to the *k*-NN method as well).

To improve accuracy of the estimation, the $amplifier(p_a, p_i)$ calculates an approximate value of the $v_{i,b}$ with comparing the sizes of projects p_a and p_i, i.e. the amplifier indicates what times p_a's value is p_i's value. The *amplifier* derived from the fact that the p_a's value is several times larger (or smaller) than the p_i's value when p_i is a similar to p_a. It's because the similarity is computed by vector operation but not *Euclidean distance*. $sim(p_a, p_i)$ is computed by comparing tendencies of the values, whereas *Euclidean distance* is computed by comparing

absolute values. Formally, we can define the *amplifier*(p_a, p_i) as:

$$amplifier(p_a, p_i) = \frac{f_a}{f_i} \quad (4)$$

where f_a denotes the function points of project p_a.

2.5. Row-column deletion method (RC deletion)

This method deletes the projects and/or metrics with missing values from a data set to build a complete data set. Listwise deletion [8] [15] is a subset of this method, which only deletes projects with missing values. The RC deletion allows deleting metrics to reduce the projects to be deleted. There exists a trade-off between deletion of metrics and deletion of projects.

3. Experiment

3.1. Overview

Using the ISBSG data set, we experimentally compare missing data techniques (imputation methods and the RC deletion) by evaluating the prediction performance of effort estimation models after applying the techniques.

We cannot evaluate the accuracy of imputed (filled) values by comparing true values with imputed values since we use the data set that originally contained missing values. Therefore, using a test data set with no missing values, we indirectly evaluate the performance of imputation methods by evaluating the performance of effort estimation for the test data.

3.2. Dataset

In the experiment, we used the ISBSG data set, collected from 20 nations' software development companies [5] and with many missing values (missing value ratio was 58%). Furthermore, the ISBSG data set has been widely used for past empirical experimentations that evaluated various missing data techniques and estimation methods [9] [10] [13].

We assumed that to estimate the effort is the end of the design phase. The extracted data set of 849 projects (missing value ratio was 39%) whose summary work effort (estimation target) was recorded (i.e. not missing), the count approach of FP was <IFPUG>, the development type was <new development> and the data quality rating was <A> or <B> [9].

Table 1 presents metrics contained in the ISBSG data set. We chose the four metrics to estimate the objective variable. Although <Project elapsed time> is an actually measured value, we included it as a predictor variable since (1) it probably affects productivity and cost, (2) the planned value of the project elapsed time is not collected, (3) the project duration is usually fixed in the initial stage of a project and <Project elapsed time> is usually not widely different from its planned value. Mean imputation and similarity-based imputation focus on filling the numerical variable, and then we removed categorical variables such as development platform, language type and business area type. Furthermore, the number of projects to build regression models falls if we include too many metrics in predictor variables. For these reasons, we chose the four metrics as predictor variables.

Table 1. Metrics used in the experiment

	Name	Missing Value Ratio
Predictor variables	Function points	0%
	Project elapsed time [month]	8.8%
	Effort plan [person-hours]	77.0%
	Effort specify [person-hours]	71.2%
Objective variable	Summary work effort [person-hours]	0%

We divided the data set according to missing values into the *fit data set* of 706 projects (containing 47% missing values) and the *test data set* of 143 projects (with no missing value). The fit data set is used for building estimation models and the test data set is for evaluation of the estimation performance of built models.

The fit data set has at least one missing value in all projects, so RC deletion first deletes the metrics with many missing values and then deletes the projects with missing values in the fit data set. According to the metrics to be deleted, we built three data sets: by deleting <Effort plan> (72 projects), by deleting <Effort specify> (28 projects) and by deleting both <Effort plan> and <Effort specify> (631 projects). We used k = 3 in the k-NN method and k = 8 in the CF method whose residual mean square showed the minimum when we built regression models.

3.3. Evaluation criteria

We used four evaluation criteria: *magnitude of absolute error* (MAE), *magnitude of relative error* (MRE), *magnitude of error relative* (MER) [3], and Pred(25) [14]. MRE and MER are defined as (5) and (6) respectively as follows (where X = actual effort, $\hat{X}$ = predicted effort):

$$MRE = \frac{\left|\hat{X} - X\right|}{X} \quad (5)$$

$$MER = \frac{\left|\hat{X} - X\right|}{\hat{X}} \ . \quad (6)$$

Table 2. Estimation performance when each method was used

	MdMAE	MdMRE	MdMER	Pred(25)
Mean imputation	2648	0.818	1.112	20%
Pairwise deletion	1036	0.461	0.609	28%
Similarity-based method (*k*-NN method)	760	0.304	0.268	43%
Similarity-based method (CF method)	829	0.274	0.295	46%
RC deletion (P deletion)	1050	0.458	0.416	28%
RC deletion (S deletion)	1463	0.479	0.526	27%
RC deletion (P, S deletion)	1875	0.555	0.563	18%

P is Effort plan and S is Effort specify

MRE and MER are criteria to evaluate overestimation and underestimation respectively [3]. Using either of them is insufficient because even if the MRE of a model is small, the model might overestimate if MER is much greater than MRE. Pred(25) is the percentage of predictions that fall within 25 percent of the MRE.

3.4. Experimental procedure

The experimental procedure is as follows.

Step 1. We apply each missing data technique to the fit data set.

Step 2. Using the fit data set, we apply a stepwise regression analysis to build an estimation model whose objective variable is <Summary work effort>.

Step 3. Considering that the summary work effort of the test data set is unknown, we estimate the summary work effort of the test data set with built models and calculate the value of each evaluation criterion.

4. Results and discussion

4.1. Overall results

Table 2 shows the median of MAE, MRE and MER (MdMAE, MdMRE and MdMER), and Pred(25) of estimations when we applied each missing data technique. Fig. 1 and Fig. 2 show box plots of the MRE and MER values when each technique is applied respectively. In Table 2, Fig. 1 and Fig. 2, "P deletion" stands for a data set built by deleting Effort plan, and "S deletion" for Effort specify. The result showed that the similarity-based imputation methods (*k*-NN method and CF method) showed better performance than other methods in all criteria.

We ascertain whether each evaluation criterion is statistically significant. The value of MAE, MRE, MER is not normally distributed, so we use the non-parametric *Wilcoxon matched-pairs signed-ranks test* to assess their statistical differences. And we use a *chi-square test* for Pred(25). The level of significance is 0.05 in each test. The result of each test is described in the following section.

4.2. Comparison of similarity-based imputation and other imputation

Among three imputation methods, similarity-based imputation methods (*k*-NN method and CF method) were much better than the mean imputation (see Table 2). This result follows the past research using artificial missing values [15]. Similarity-based methods were also much better than the pairwise deletion. As shown in Fig. 1 and Fig. 2, inter-quartile range (IQR) of similarity-based imputation methods was narrower than mean imputation and pairwise deletion, i.e. the variability of the errors is lower. Furthermore, each test of the result of similarity-based imputation and the other imputation showed that each evaluation criterion was statistically significant.

4.3. Comparison of similarity-based imputation and RC deletion

As shown in Table 2, similarity-based imputation methods were also better than all RC deletions (P deletion, S deletion and P, S deletion). In the RC deletions, although there was no significant difference between P deletion and S deletion in MdMRE and Pred(25), P deletion was better than S deletion in MdMAE and MdMER. Although 631 projects remained in the P, S deletion data set, it seems deleting both P (Effort plan) and S (Effort specify) removed useful information for the estimation. On the other hand, it can be considered that the P deletion data set (72 projects) and the S deletion data set (28 projects) contained too few projects to build an accurate model, i.e. the confidence interval might become wide.

As shown in Fig. 1 and Fig. 2, IQR of the similarity-based imputation method (*k*-NN method) was narrower than three RC deletions. IQR of the CF method was narrower than three RC deletions in MRE. Although IQR of the CF method was wider than P deletion, the position of the boxplot of the CF method was lower than three RC deletions. Therefore, the CF method was better than three RC deletions. Furthermore, each test of the result of the similarity-based imputation and RC deletion showed that

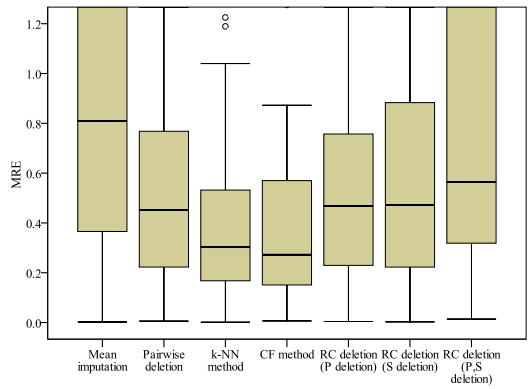

Fig. 1. MRE values when each method was used

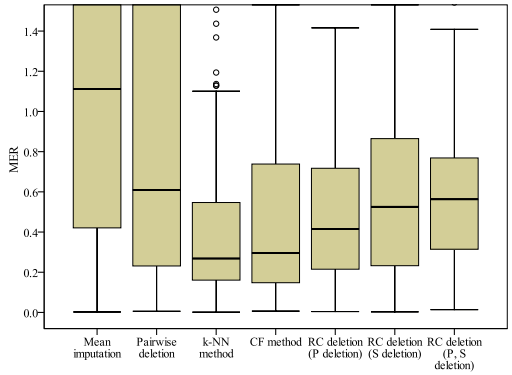

Fig. 2. MER values when each method was used

each evaluation criterion was statistically significant except for MER between the CF method and RC deletion (P deletion).

This indicates that the large project data set with a lot of missing values is worthier than the small project data set with no missing value. We believe engineers should not be afraid of usage of "incomplete" project data since it is useful enough for the effort estimation. In addition, interestingly, all RC deletions still showed better performance than the mean imputation. This indicates that the incomplete data set is useful only if missing values are properly filled in.

4.4. Comparison of similarity-based imputation methods

Comparing two similarity-based imputation methods, the *k*-NN method was better than the CF method in terms of MdMAE and MdMER, while this became the opposite in terms of MdMRE and Pred(25) (see Table 2). As shown in Fig. 1 and Fig. 2, IQR of the *k*-NN method was narrower than the CF method in MRE and MER. Furthermore, each test of the result of the CF method and *k*-NN method showed that MAE and MER were statistically significant.

From these results, there is not a big difference between the *k*-NN method and CF method in their prediction performance. However, we could say the *k*-NN method tends to build an overestimate model, while the CF method tends to build an underestimate model. Therefore, it is preferable for an estimator to use both methods to confirm that there is no big difference in their estimates.

4.5. Threats to validity

We evaluated using only one data set; however, there are many conditions (i.e., various amounts and distributions of the missing values) in the data set collected by the actual industrial organizations. Other experimental simulation would have to be performed on a different data set for improving reliability of the study.

Our experimentation focused on stepwise multiple regression analysis as the modeling technique. This is one of the most popular techniques, such as ordinary least squares regression analysis, for building cost estimation models. However, it is possible that other modeling techniques would require different missing data techniques.

5. Related work

Jonsson et al. [6] focused on filling in missing data accurately. They *artificially* deleted some recorded values at random and compared filled in values with originally recorded values. The result showed that the *k*-NN method using incomplete case strategy and *k*-value of the square root of complete cases performed well.

Sentas et al. [13] and Strike et al. [15] compared several imputation methods and listwise deletion, but they used a data set whose values were artificially deleted at random. In [13], multinomial logistic regression showed better performance than listwise deletion, mean imputation, expectation maximization and regression imputation for estimating categorical missing values. In [15], they have compared listwise deletion, mean imputation and eight different types of hot-deck imputation. The results showed that the *k*-NN method (Euclidean distance) and a z-score standardization showed the best performance.

Although the data set whose values were deleted at random was used in the experiment ([6] [13] [15]), the missing data pattern is realistically not random but burst. There are several reasons of the burst missing. The burst missing is generated if several data sets collected by different business organizations having different policies on data collection are merged. Even with the same organization, the burst missing is generated if its policy is different from each period.

Cartwright et al. [1] and Myrtveit et al. [11] compared imputation methods by evaluating the goodness of fit of effort estimation models built after imputations were applied to a fit data set containing missing values. Therefore, they did not evaluate the *prediction performance* of the models. But it remains possible that the built models too much fit (i.e., overfitting [4]) the fit data set, and then the models might not perform well when they evaluate the estimation accuracy to another data set.

In this paper, we have done imputation or RC deletion to a data set containing missing values *naturally* and prepared a test data set with no missing values to do this evaluation. Furthermore, none of the past research compared imputation methods with RC deletion, which is often used in industry.

6. Conclusion

In this paper, we experimentally compared missing data techniques (mean imputation, pairwise deletion, *k*-NN method, CF method and RC deletion) by evaluating the prediction performance of effort estimation models after applying the techniques. Our findings include:

- Similarity-based imputation methods (*k*-NN method and CF method) showed better performance than all RC deletions. This indicates that a large project data set with a lot of missing values is worthier than a small data set with no missing value.
- The mean imputation was worse than all three RC deletions. This indicates that an incomplete data set becomes useful only if its missing values are properly filled in.
- The pairwise deletion was worse than similarity-based imputation.
- The *k*-NN method tends to build an overestimate model, while the CF method tends to build an underestimate model. We recommend an estimator to use both methods to confirm there is no big difference in their estimates.

Our future work will be to use other data sets to increase the validity of the results. Furthermore, we will develop similarity-based imputation method to improve imputation performance and make comparison of the cost estimation performances with other methods given missing data (e.g. optimized set reduction).

7. Acknowledgments

This work is being conducted as a part of the StagE Project, the Development of Next Generation IT Infrastructure, supported by Ministry of Education, Culture, Sports, Science and Technology.

8. References

[1] Cartwright, M., Shepperd, M.J. and Song, Q., "Dealing with Missing Software Project Data," *Proc. 9th IEEE International Software Metrics Symposium (Metrics'03)*, Sydney, Australia, pp.154-165, 2003.

[2] Draper, N. R., Smith, H., *Applied Regression Analysis*, 3rd ed, John Wiley and Sons, New York, 1998.

[3] Foss, T., Stensrud, E., Kitchenham, B. and Myrtveit, I., "A Simulation Study of the Model Evaluation Criterion MMRE," *IEEE Transactions on Software Engineering*, Vol.29, No.11, pp.985-995, 2003.

[4] Hawkins, D.M., "The Problem of Overfitting," *Journal of Chemical Information and Modeling*, Vol.44, No.1, pp.1-12, 2004.

[5] ISBSG Estimating, Benchmarking and Research Suite Release 9, International Software Benchmarking Standards Group, http://www.isbsg.org/

[6] Jonsson, P. and Wohlin, C., "An evaluation of *k*-nearest neighbour imputation using likert data," *Proc. 10th IEEE International Software Metrics Symposium (Metrics'04)*, Chicago, Illinois, pp.108-118, 2004.

[7] Kromrey, J. and Hines, C., "Nonrandomly missing data in multiple regression: An empirical comparison of common missing-data treatments," *Educational and Psychological Measurement*, Vol.54, No.3, pp.573-593, 1994.

[8] Little, R.J.A. and Rubin, D.B., *Statistical Analysis with Missing Data*, 2nd ed., John Wiley and Sons, New York, 2002.

[9] Mendes, E., Lokan, C., Harrison, R. and Triggs, C. "A Replicated Comparison of Cross-company and Within-company Effort Estimation Models using the ISBSG Database," *Proc. 11th IEEE International Software Metrics Symposium (Metrics'05)*, Como, Italy, pp.36, 2005.

[10] Moses, J. and Farrow, M., "Assessing Variation in Development Effort Consistency Using a Data Source with Missing Data," *Software Quality Journal*, Vol.13, No.1, pp.71-89, 2005.

[11] Myrtveit, I., Stensrud, E. and Olsson, U. H., "Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods," *IEEE Transactions on Software Engineering*, Vol.27, No.11, pp.999-1013, 2001.

[12] Ohsugi, N., Tsunoda, M., Monden, A., and Matsumoto, K., "Effort Estimation Based on Collaborative Filtering," *5th International Conference on Product Focused Software Process Improvement (Profes2004)*, Kyoto, Japan, Vol.3009, pp.274-286, 2004.

[13] Sentas, P. and Angelis, L., "Categorical missing data imputation for software cost estimation by multinomial logistic regression," *The Journal of Systems and Software*, Vol.79, No.3, pp.404-414, 2006.

[14] Shepperd, M. and Schofield, C., "Estimating software project effort using analogies," *IEEE Transactions on Software Engineering*, Vol.23, No.12, pp.736-743, 1997.

[15] Strike, K., El Eman, K. and Madhavji, N., "Software cost estimation with incomplete data," *IEEE Transactions on Software Engineering*, Vol.27, No.10, pp.890-908, 2001.