PAPER *Special Section on Empirical Software Engineering*

Influence of Outliers on Estimation Accuracy of Software Development Effort[*]

Kenichi ONO[†], ***Nonmember***, **Masateru TSUNODA**[††a)], **Akito MONDEN**[†††], ***and*** **Kenichi MATSUMOTO**[†], ***Members***

SUMMARY When applying estimation methods, the issue of outliers is inevitable. The extent of their influence has not been clarified, though several studies have evaluated outlier elimination methods. It is unclear whether we should always be sensitive to outliers, whether outliers should always be removed before estimation, and what amount of precaution is required for collecting project data. Therefore, the goal of this study is to illustrate a guideline that suggests how sensitively we should handle outliers. In the analysis, we experimentally add outliers to three datasets, to analyze their influence. We modified the percentage of outliers, their extent (e.g., we varied the actual effort from 100 to 200 person-hours when the extent was 100%), the variables including outliers (e.g., adding outliers to function points or effort), and the locations of outliers in a dataset. Next, the effort was estimated using these datasets. We used multiple linear regression analysis and analogy based estimation to estimate the development effort. The experimental results indicate that the influence of outliers on the estimation accuracy is non-trivial when the extent or percentage of outliers is considerable (i.e., 100% and 20%, respectively). In contrast, their influence is negligible when the extent and percentage are small (i.e., 50% and 10%, respectively). Moreover, in some cases, the linear regression analysis was less affected by outliers than analogy based estimation.

key words: *case-based reasoning, multiple linear regression, effort prediction, outlier*

1. Introduction

Recently, a larger number of functions have become necessary in various software, thereby increasing the size of the software according to its requirements. To develop a large software, considerable effort (cost) is required. Without managing cost and schedule, it is difficult to prevent failures of the projects developing such software. In other words, for the development of large software, management is indispensable and based on effort estimation. To succeed in a software development project, it is important to accurately estimate development effort; thus, numerous quantitative estimation methods have been proposed and refined [4], [21], [28], [41]. Such methods include decision tree-based estimation [4]; search-based approaches [11]; ensemble effort estimation, which combines various estimation methods [21]; and COCOMO II, which is based on regression [28]. Regression-based estimation (e.g., multiple linear regression analysis) is widely used to mathematically estimate effort [15]. Furthermore, analogy based estimation [38] has recently attracted attention, with many proposals and case studies reported in the literature [3], [17], [18], [25], [45], [46]. One advantage of analogy based estimation is its similarity to human problem-solving behaviors [12], and such techniques can confirm the neighboring projects to use for estimation.

When implementing estimation methods, the issue of outliers is inevitable. In previous project datasets, several data points describing effort and software size [i.e., function points (FPs)] have often been observed to differ from those of other data points. These are referred to as outliers [6]. For instance, outliers occur when extensive reworking is performed on a project, causing it to require more effort than others. Furthermore, when the effort is inaccurately measured or recorded, the reported effort differs from the actual effort. An FP is often measured during the early phases of a project. If numerous functions are added after this measurement, the recorded FP differs from the actual FP. Thus, the outliers may affect the estimation accuracy.

There are two methods of suppressing the influence of outliers on the estimation accuracy. The first is to collect the data very carefully; this involves the manual recording of major metrics concerning effort estimation, such as FPs and effort. This suppresses outliers in the dataset. The other method is to apply mathematical outlier elimination methods to the dataset. Outliers are detected in the output of an estimation model. Outlier elimination methods identify a project as outlying when its impact on the outcomes is large. For instance, Cook's distance is widely used as an outlier elimination method when applying linear regression analysis. Further outlier elimination methods for effort estimation [35] have been proposed.

Although the extent of the influence of outliers has not been clarified, several studies have evaluated outlier elimination methods [34], [35]. It is unclear whether we should always treat outliers sensitively, whether outliers should always be removed before estimation, and what amount of precaution is required to collect project data. Thus, our study focuses on proposing a guideline that suggests how

Manuscript received February 28, 2020.
Manuscript revised June 26, 2020.
Manuscript publicized October 2, 2020.

[†]The authors are with Nara Institute of Science and Technology, Ikoma-shi, 630–0192 Japan.

[††]The author is with Kindai University, Higashiosaka-shi, 577–8502 Japan.

[†††]The author is with Okayama University, Okayama-shi, 700–8530 Japan.

[*]This work is an extended study of K. Ono et al., "Influence of Outliers on Analogy Based Software Development," In Proc. of International Conference on Computer and Information Science (ICIS 2016), pp.849–854, 2016.

a) E-mail: tsunoda@info.kindai.ac.jp

DOI: 10.1587/transinf.2020MPP0005

sensitively we should handle outliers.

The basic concept of our study was inspired by Seo et al. [34]. They evaluated outlier elimination methods using two effort estimation methods and found no statistical difference in accuracy between estimations with and without elimination methods. This suggests that outliers do not significantly affect estimation accuracies. Based on this suggestion, we raised the following question: "To what extent does the influence of outliers on estimation accuracy vary when the extent of outliers included in the dataset varies?"

In our experiment, we consider the influence of outliers on the effort estimation accuracy. The relationships between the number and extent of outliers and estimation accuracy are unclear. To analyze these relationships, we experimentally inserted outliers into the dataset. More specifically, we altered the values of the dependent variable "effort" and the most significant independent variable "FP." Furthermore, we set the following parameters, to analyze their influence on the estimation accuracy:

- Percentage of outliers
- Extent of outliers

The percentage of outliers is such that, for instance, when it is set at 10%, we specify 10 out of 100 data points as outliers. Furthermore, when the extent of outliers is set at 100%, a data point denoting 100 person–hours of effort instead denotes 200 person–hours. Moreover, we changed the variables that include outliers: they could be dependent or independent variables. Thus, we added outliers to effort and FP.

Additionally, we considered the locations of outliers in a dataset. We assumed that when effort is estimated, outliers are included in the following pattern:

- Past project data: includes outliers; estimation target data: no outliers
- Past project data: includes outliers; estimation target data: includes outliers
- Past project data: no outliers; estimation target data: includes outliers

Outliers could be included in the target data estimation (test dataset) alongside past project data (learning dataset). Generally, eliminating outliers in target data estimation is challenging, and therefore their influence was not evaluated. However, when an estimation model is used in practice, the influence of outliers in a test dataset should be considered because evaluating estimation accuracy without considering them may constitute overvaluing. Furthermore, our analysis considers the estimation accuracy of models in practice by demonstrating the influence of outliers in the test dataset.

The remainder of this paper is structured as follows: Sect. 2 explains software development effort estimation, Sect. 3 explains aspects of outliers, Sect. 4 describes our experimental set-up, Sect. 5 reports the results of the experiment, Sect. 6 contains a discusses thereof, Sect. 7 reviews the related work, and Sect. 8 summarizes and concludes the paper.

2. Software Development Effort Estimation

In this study, we investigated the influence of outliers on multiple linear regression analysis and analogy based estimation when estimating software development effort. Multiple linear regression analysis is widely used for constructing estimation models (e.g., [28]) and is used as the benchmark for effort estimation models. Analogy based estimation is another popular estimation method that has been widely studied [3], [17], [18], [25], [45], [46]. The estimation methods we employ are those found in Seo et al. [34]; both are explained below.

Huang et al. [13] categorized effort estimation methods into expert judgment, parametric models, and machine leaning methods. In a systematic review of effort estimation studies [15], linear regression models were found to be the most popular parametric model, and analogy based estimation was found to be the most popular machine learning approach. Although this study [15] was published in 2007, both methods have been frequently used in more recent studies (e.g., [2], [8], [34]). Therefore, we used multiple linear regression and analogy based estimation as representatives of parametric model and machine leaning approaches, respectively.

2.1 Multiple Linear Regression Analysis

Multiple linear regression analysis is widely used to mathematically estimate development effort. During regression analysis, an estimation model is constructed from the datasets of past projects using the least squares method. When the development effort (dependent variable) is denoted as y, and independent variables (e.g., the functional size) are denoted as x_1, x_2, $\ldots$, and x_k; then, the linear regression-based effort estimation model is expressed as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon. \tag{1}$$

Here, β_0 is an intercept; β_1, β_2, $\ldots$, β_k are partial regression coefficients; and ε is an error term. A logarithmic transformation is often used to construct effort estimation models. As general principle, when building a proper model using linear regression analysis, the number of data points must be five to ten times larger than the number of independent variables [44].

To build a regression model that predicts software development efforts, a log-transformation is sometimes applied to ratio scale variables, to enhance the model accuracy [19]. This is because some variables follow a lognormal distribution, and multiple linear regression analysis performed with log-transformations outperforms the same analysis without it [9]. Notably, log-transformation models should not be recommended to practitioners—they obscure the major pitfalls of the underlying data and can often lead to unsuitable decisions.

Table 1 Dataset used for analogy based effort estimation

	Variable 1	*Variable* 2	…	*Variable j*	…	*Variable l*
p_1	m_{11}	m_{12}	…	m_{1j}	…	m_{1l}
p_2	m_{21}	m_{22}	…	m_{2j}	…	m_{2l}
…	…	…		…		…
p_i	m_{i1}	m_{i2}	…	m_{ij}	…	m_{il}
…	…	…		…		…
p_n	m_{n1}	m_{n2}	…	m_{nj}	…	m_{nl}

2.2 Analogy Based Estimation

Shepperd et al. [38] proposed an analogy based estimation method built on case based reasoning (CBR), which is studied in the field of artificial intelligence. In CBR, a case resembling the current issue is selected from the accumulated set of past cases, and the solution of that case is applied to the one in question. This is because CBR assumes similar issues can be solved by similar solutions. Likewise, analogy based estimation assumes that when software development projects are similar, their development efforts are also similar. Similarity is identified using attributes such as software size, business sector, and programming language employed.

Analogy based estimation uses an $n \times l$ matrix (as shown in Table 1) in which p_i represents the *i-th* project and m_{ij} represents the value of the *j-th* variable. That is, each row denotes a data point (i.e., a project) and each column denotes a metric. We assume that p_a is an estimation target project and $\hat{m}_{ab}$ is the estimated effort of m_{ab}. The procedure of analogy-based estimation consists of three steps as follows:

Step 1: Because each variable has a different value range, a range of $[0, 1]$ is specified in this step. The value m'_{ij}, which is the normalized value of m_{ij}, is calculated as

$$m'_{ij} = \frac{m_{ij} - min(m_j)}{max(m_j) - min(m_j)}, \tag{2}$$

where m_j denotes the *j-th* variable, and $max(m_j)$ and $min(m_j)$ denote the maximum and minimum values of m_j, respectively. No variable is used when $max(m_j)$ and $\min(m_j)$ are equivalent because all values in the variable are identical in such cases, and the value becomes unusable for effort estimation. The above equation is one of the most popular methods of normalizing a value range [42].

Step 2: To find projects similar to the estimated project p_a (i.e., to identify neighboring projects), the similarity between p_a and another project p_i is calculated. Variables of p_a and p_i are used as elements of vectors, and the cosines of these vectors are regarded as similarities. Thus, the similarity $sim(p_a, p_i)$ between p_a and p_i is calculated as

$$sim(p_a, p_i) = \frac{\sum\limits_{j \in M_a \cap M_i} (m'_{aj} - avg(m'_j))(m'_{ij} - avg(m'_j))}{\sqrt{\sum\limits_{j \in M_a \cap M_i} (m'_{aj} - avg(m'_j))^2} \sqrt{\sum\limits_{j \in M_a \cap M_i} (m'_{ij} - avg(m'_j))^2}}, \tag{3}$$

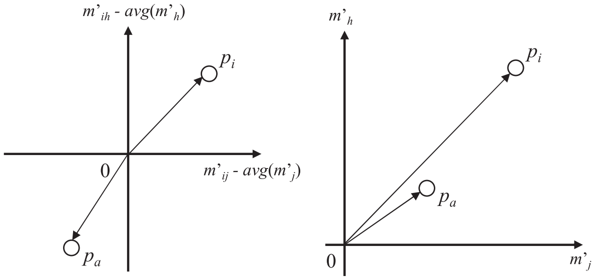


Fig. 1 Difference of similarity computation.

where M_a and M_i denote the sets of variables measured in project p_a and p_i, respectively; $avg(m'_j)$ is the average of the *j-th* variable. In the above equation, $j \in M_a \cap M_i$ indicates that the *j-th* variable exists in both M_a and M_i; thus, missing values (i.e., where m_{ij} is not recorded) are excluded from the calculation.

The essential idea of Eq. (3) was proposed in a previous study [43]; there, the median was used instead of the average. By subtracting $avg(m'_j)$ from m'_{ij} in the equation, each value becomes positive when it exceeds the average and negative when below the average. Figure 1 shows the difference of similarity computation between typical cosine similarity and the computation proposed in study [43]. The left-hand figure illustrates the relationship between p_a and p_i after subtracting $avg(m'_j)$, and the right-hand figure denotes the relationship without the subtraction (i.e., the typical cosine similarity). From the equation, it can be seen that the range of $sim(p_a, p_i)$ is $[-1, 1]$. When $sim(p_a, p_i)$ is large, p_a and p_i are regarded as similar and used in Step 3.

Step 3: The estimated effort of project p_a is calculated using the actual effort of k neighboring projects. Although the average effort of the neighboring projects is generally used, we adopted the size-adjustment method, which has produced highly accurate estimations in several studies [18], [25], [46]. The estimated effort $\hat{m}_{ab}$ of project p_a is calculated as

$$\hat{m}_{ab} = \frac{\sum\limits_{i \in k-nearestProjects} (m_{ib} \times amp(p_a, p_i) \times sim(p_a, p_i))}{\sum\limits_{i \in k-nearestProjects} sim(p_a, p_i)}, \tag{4}$$

$$amp(p_a, p_i) = \frac{fp_a}{fp_i}, \tag{5}$$

where m_{ib} is actual effort of project p_i, and fp_a and fp_i are the software sizes of projects p_a and p_i, respectively. The size-adjustment method assumes that the effort is s times larger (s is a real, positive number) when the software size is s times larger. The method adjusts the effort of p_i based on the ratio of the target project's size fp_a and a neighboring project's size fp_i; it assumes that productivity (i.e., the ratio of fp_a to fp_i) is almost identical between similar projects, though the range of productivity in datasets is often large.

3. Outlier

3.1 Outlier Elimination Method

Manual outlier elimination requires considerable resources to judge whether data points are actually outliers. In addition, if normal data points are eliminated erroneously through outlier elimination, the number of data points used to estimate effort will decrease, degrading the estimation accuracy. The identification, elimination, and analysis of outliers' impacts are very important [1]; therefore, outliers should be carefully eliminated, and several mathematical outlier elimination methods have been proposed.

An outlier is referred to as a "productivity extreme," to distinguish it from a statistical outlier. Typically, the outliers are assessed with respect to combined values: for instance, productivity is a combined value expressing the effort divided by the FP (i.e., the productivity ratio). To detect outliers, we can use Cook's distance-, least trimmed squares- [35], *k*-means- [35], or Mantel's correlation-based [17] elimination methods. However, the objective of these methods should not be confined to the elimination of outliers: further detailed studies should be conducted to try to identify the cause of the behavior. For instance, a Cook's distance-based method may detect a perfectly valid data point within the normal distribution of input values; however, this point may contain an additional factor not included in the model, which may significantly affect the estimation. This is not a valid reason to eliminate such a data point: data-point elimination must be supported by knowledge and acceptable concepts.

We did not apply outlier elimination methods to estimate effort because our goal was not to evaluate the performance of elimination methods but to evaluate the influence of outliers on estimation accuracy. It is important to consider the existence of outliers before eliminating them, even if the application of mathematical outlier deletion methods without these considerations does not degrade estimation accuracy.

Irrespective of whether mathematical or manual outlier elimination is used, it is difficult to eliminate all outliers in a test dataset. For example, assume that you estimate the effort of a project (i.e., a project in a test dataset), and that numerous additional functions are performed after the effort estimation. Then, the recorded FP (used to estimate effort) differs from the actual FP, and the estimated and actual efforts will also differ. In this case, it is difficult to classify them as outliers before finishing the project (i.e., when estimating effort).

3.2 Aspects of Outliers

We assume that the influences of outliers on effort estimation differ according to their conditions; for instance, the number of outliers (e.g., 10 out of 100 data points in the dataset are outliers). Thus, the relationship between estimation accuracy and outlier characteristics should be analyzed; this helps to clarify whether we should always be sensitive to outliers, whether outliers should always be removed before estimation, and the degree of caution required when collecting project data.

To analyze the relationship, we assumed that the conditions of outliers are affected by the four following aspects of outlier occurrence: (1) the percentage of outliers, (2) the extent of outliers, (3) the variables containing outliers, and (4) the locations of outliers.

(1) Percentage of outliers: This is the percentage of outliers in a dataset. For example, when a dataset contains 100 data points, and 10 data points are converted into outliers, this percentage is 10%. This parameter considers the frequency of outlier occurrence. For example, when the FP measurements are incorrect across many projects, the outlier percentage increases.

(2) Extent of outliers: This denotes the difference between the recorded and actual values of a variable. For example, the actual effort of a project is recorded as 200 person–hours, and it is altered to 400 person–hours in experiments; therefore, the extent is 100% ($|400 - 200|/200$). This parameter considers the measurement accuracy: when the effort measurement is highly inaccurate, the extent of outliers increases.

(3) Variables including outliers: outliers can be included in both dependent and independent variables. Thus, in the experiment, we added outliers to the effort and FP (considered to have the largest influence on effort). Effort and FPs are measured manually; however, they are difficult to measured accurately. Therefore, they may include outliers.

(4) Locations of outliers: outlier elimination methods [35] that eliminate outliers based on outcomes in the output of an estimation model (see Sect. 1) assume that outliers are included in learning datasets (past projects); thus, they remove them from the dataset. However, it is probable that test datasets also contain outliers (i.e., an estimation target project itself is an outlier). Outlier locations are classified into the following types:

- Type 1—Learning dataset: includes outliers; test dataset: no outliers.
- Type 2—Learning dataset: includes outliers; test dataset: includes outliers.
- Type 3—Learning dataset: no outliers; test dataset: includes outliers.

We assume that Type 1 is a rare case because outliers can also feature in test datasets. We set Type 2 by assuming that an outlier elimination method is not applied to the learning dataset; thus, outliers are retained in the test dataset because they are difficult to eliminate. Similarly, we set Type 3 by assuming that outliers are eliminated by some elimination method; however, outliers still remain in the test dataset because they are difficult to eliminate.

The influence of outliers is considered to vary when one of the four aspects varies. In the experiment, we changed certain aspects and analyzed the estimation accuracy.

3.3 Importance of Aspects of Outliers

After considering the treatment of outliers in outlier elimination methods, we conclude that the four aspects explained in Sect. 3.1 are the most significant, even if other characteristics may affect the estimation accuracy.

Grubbs's test can consider extremely large values as indicative of outliers, instead of measuring aspect (2). However, large values alone do not always affect the accuracy. For example, if both the software size and effort are extremely large, but the productivity is normal, accuracy is not notably affected by outliers. In contrast, we changed aspect (2) for software size and effort in our analysis. This alters productivity and can affect the accuracy.

Cook's distance-based elimination removes data points as outliers when the relationships between independent variables and a dependent variable are changed. Instead of aspect (3), outliers across a range of independent variables may affect the relationship. However, the most important independent variable is software size, and other independent variables are not always included in estimation models. For example, basic COCOMO considers only software size as an independent variable [5]. We altered aspect (3) for software size and effort, anticipating that this would affect the relationship.

Aspects (1) and (4) are very simple; therefore, we did not consider it necessary to substitute them with other outlier characteristics.

4. Experiment

4.1 Datasets

We used three datasets to evaluate the effort estimation accuracy. Effort was measured in hours for all datasets. The first dataset was provided by the International Software Benchmark Standard Group (ISBSG), and it is referred to as the ISBSG dataset; it includes project data collected from software development companies across 20 countries [14]. The dataset (Release 9) includes 3026 projects, with over half these conducted during the period 1998–2004. We recorded 99 variables. The ISBSG dataset contains low-quality project data (data quality ratings are also present in the dataset). Therefore, we extracted projects using the method of the previous study [23] (e.g., the data quality was rated as either A or B). In addition, we excluded projects with missing values. Thus, 611 projects were selected. Independent variables were selected according to the previous study [23] (unadjusted FP, duration, development type, programming language, and development platform). The categorical variables of development type, programming language, and development platform were transformed into dummy variables.

The second dataset was collected from a Canadian software house (i.e., a single company) by Desharnais during the 1980s; it includes three different development environments and is referred to as the Desharnais dataset [10]. The development domain is unknown; however, considering its functional size and age, it is most likely that of a business application. This dataset has been widely used in effort estimation studies [17], [38], and it has a relatively large number of data points and independent variables compared to other open-access datasets. A total of 77 data points were found to remain after those featuring missing values were removed. Although the dataset is not very recent, it has been used in recent studies [26], [33], [39]. We removed the development year, adjusted FP, number of transactions, and number of entities from the dataset; then, we used the unadjusted FP, duration, team experience (years), manager's experience (years), adjusted factor, and programming language as independent variables; programming language was converted into a dummy variable.

Lastly, the Kitchenham dataset describes 145 projects of a software development company; it was released by Kitchenham et al. in [20]. These projects were started between 1994 and 1998; hence, the dataset is fairly old. We selected 135 projects for which no values were missing. Three variables (duration, adjusted FP, development type) were chosen as independent variables, and any variables unsuitable for effort estimation (e.g., the project manager's effort estimate) were eliminated; development type was converted into a dummy variable.

Table 2 shows the basic statistics of the variables in the datasets that were transformed into outliers (see the following subsection).

4.2 Experimental Procedure

We experimentally added outliers using a procedure based on a hold-out method, as follows:

1. Based on the percentage of outliers, data points are selected from the dataset. For example, when the number of data points is 50 and the outlier percentage is 10%, five data points are selected. We set the outlier percentages as 10% and 20%.

Table 2 Basic statistics of variables transformed into outliers

(a) ISBSG dataset

	Min.	Average	Median	Max.	Standard deviation
Effort	62	5083.6	2408	78472	8130.5
FP	10	530.7	250	13580	967.5

(b) Desharnais dataset

	Min.	Average	Median	Max.	Standard deviation
Effort	546	4833.9	3542	23940	4188.2
FP	73	298.0	258	1127	182.3

(c) Kitchenham dataset

	Min.	Average	Median	Max.	Standard deviation
Effort	219	3169.1	1557	113930	9933.6
FP	18.9	527.8	258.24	18137.48	1572.9

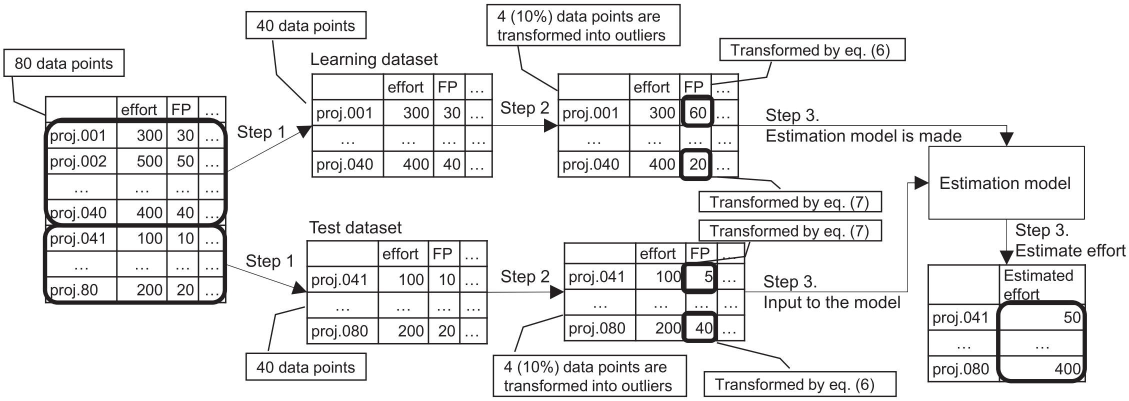


Fig. 2 Overview of the experimental procedure.

2. Based on the extent of outliers, the values of a metric (i.e., effort or FP) are changed for the selected data points. For example, say the extent of outliers is set at 100%; if FP = 100, then FP changes to 200 or 50; that is, half of the selected data points are set as overestimates (e.g., FP set at 200), and the remainder are set as underestimates (e.g., FP set at 50). The extent of outliers was set at 50% and 100%.

In Step 2, we used the following equations:

$$ov = av(1 + eo/100), \tag{6}$$

$$ov = av/(1 + eo/100). \tag{7}$$

Here, ov is an outlier value, av is an actual value, and eo represents the extent of outliers. Equations (6) and (7) are used to form overestimates and underestimates, respectively. As shown in the Appendix, the definition of eo is the same as that of the balanced relative error (*BRE*; see Sect. 4.3).

The maximum outlier percentage and extent were set at 20% and 100%, respectively. As explained in Sect. 5, when the outlier percentage is 20% or the outlier extent is 100%, the influence of outliers becomes non-negligible. That is, these settings suffice to determine whether the influence is non-negligible. By applying the aforementioned procedure, we analyzed the influence of outliers on the estimation methods, as follows:

1. The dataset was randomly divided into two equal sets. One was treated as a learning dataset, and the other was treated as a test dataset. The learning dataset was used to compute the estimated effort (past projects), and the test dataset was used as the estimation target (current projects).
2. Outliers were experimentally added to the datasets, according to the aforementioned procedure.
3. The effort was estimated using the dataset including outliers. The evaluation criteria of the estimation accuracy were then calculated from the results.
4. We repeated Steps 1–3 ten times (i.e., we divided the dataset equally into two sets ten times and added outliers after the division, to set the outlier percentage as 10% and 20% and the outlier extent as 50% and 100%); then, we calculated the average and median of the evaluation criteria. Furthermore, we calculated the estimation criteria when no additional outliers were present. Next, we compared both criteria, to analyze the influence of outliers.

Figure 2 illustrates Steps 1–3. In the figure, the outlier percentage was 10%, outliers were included in both the learning and test datasets, and linear regression analysis was used for the estimation. We did not remove any outliers from the learning dataset before adding artificial outliers. This was because the specific outliers eliminated vary under different elimination methods, and Seo et al. [34] have shown that the estimation accuracy does not differ statistically when computed with and without elimination for various datasets, including the IBSBG R9 and Desharnais datasets.

As described above, we implemented the hold-out method instead of n-fold cross-validation (i.e., 10-fold cross-validation). The hold-out method is sometimes applied to evaluate the performance of prediction models in software engineering studies (e.g., [8], [40]). We apply this method partially as a result of the limitations of the experiment [40]. In our case, when we set the outlier percentage of the test dataset to 10%, the dataset should contain a minimum of ten data points. However, if we apply 10-fold cross-validation to the Desharnais dataset, the test dataset contains 7–8 data points. Furthermore, even if a test dataset contains more than ten but less than 20 (e.g., 15) data points, replacing one data point with an outlier reduces the outlier percentage to below 10% (e.g. 6.7%). Therefore, to increase the number of data points in test datasets, we used the hold-out method. To reduce the splitting bias, we applied this method ten times, as explained in Step 4 above.

We applied a log-transformation to construct an estimation model from multiple regression analysis. We applied it to effort, software size, and duration; this was because it is often applied to effort and software size (e.g., study [9]), and applying it to duration has been found suitable for effort estimation [30]. In the datasets, the variable distributions were highly skewed (i.e., the absolute values of skewness exceeded 1.0 [31]), as shown in Table 3.

The number of neighboring projects used for analogy based estimation was set based on the estimation accuracy of each dataset without additional outliers; it was set as 12, 5, and 5 for the ISBSG, Desharnais, and Kitchenham datasets, respectively.

4.3 Evaluation Criteria

To evaluate the accuracy of effort estimation, we used the average and median of absolute error (*AE*), the magnitude of relative error (*MRE*) [7], and the balanced relative error (*BRE*) [27]. Each criterion is calculated using the following equations, where x and $\hat{x}$ denote the actual effort and estimated effort, respectively:

$$AE = |x - \hat{x}|, \tag{8}$$

$$MRE = \frac{|x - \hat{x}|}{x}, \tag{9}$$

$$BRE = \begin{cases} \dfrac{|x - \hat{x}|}{\hat{x}}, & x - \hat{x} \geq 0 \\ \dfrac{|x - \hat{x}|}{x}, & x - \hat{x} < 0 \end{cases}. \tag{10}$$

A lower value for each criterion indicates a higher estimation accuracy. The average *AE* is also referred to as the *MAR* (mean absolute residual) [37]. Intuitively, *MRE* implies the ratio of relative error to actual effort. However, the *MRE* exhibits a bias toward underestimation [22]: the maximum possible *MRE* is 1, even for an extreme underestimation (e.g., when the actual effort is 1000 person–hours and the estimated effort is 0 person–hours, *MRE* is still 1). Therefore, in addition to the *MRE*, we adopted the *BRE* because (a) its evaluation is unbiased [29], and (b) it is commonly used in other studies. However, it is not a very reliable criterion and is mainly used for reference.

We chose the learning and test datasets without additional outliers as the baseline and calculated their evaluation criteria. Next, we calculated the differences between the baseline and the cases using datasets with outliers. When the difference was negative, the estimation accuracy was considered to be degraded by outliers. Furthermore, when the difference was large, the influence of outliers was also large. Using this difference, the influence of outliers can be explicitly shown. Notably, even if the model without outliers exhibited a lower estimation accuracy than the model with outliers, the former is definitively the best model. This is because the model without outliers is correct and represents correct information from the dataset, whereas the model with outliers provides "false information."

When the difference between the *MRE* and *BRE* exceeded 5%, we considered it to be non-negligible. We chose a 5% threshold by considering the profits of software development companies; that is, we assumed that profit is the difference between price and cost, and when the error in the cost prediction (i.e., the estimated effort) degrades by more than 5%, the error cannot be neglected.

Table 3 Skewness of effort, FP, and duration in each dataset

	ISBSG	Desharnais	Kitchenham
Effort	4.4	2.0	10.5
FP	6.7	1.8	10.6
Duration	2.5	1.5	2.2

To statistically analyze the differences between the baseline (no additional outliers) and other cases, we applied the Wilcoxon signed-rank test to the average *MRE* and *BRE*; this has often been applied to analyze differences in the evaluation criteria of past studies (e.g., [13]). We set the significance level at 0.05. We did not apply this test to other criteria because we wished to focus on discussing the experimental results obtained therewith. We gave more precedence to the differences in *MRE* and *BRE* arising between the baseline and other cases than the p-values derived through the statistical test. This was because the p-value was below 0.05 (i.e., the difference was statistically significant), even when the difference was very small. For example, although the difference in average *MRE* was 0.08%, its p-value was 0.02, as shown in the first row of Table 8.

5. Results

5.1 Overview

Tables 4–19 show the differences in the evaluation criteria between the baseline and other cases. Furthermore, the tables show the variables featuring outliers and the percentage and extent of outliers. The evaluation criteria include an average taken over ten estimations (see Sect. 4.2). The negative values indicate that the criterion was improved. Furthermore, the tables indicate the datasets used for evaluation. The "Y" in the columns "LO" (learning outliers) and "TO" (test outliers) indicates that the learning or test datasets included outliers; the columns "DS" denote the dataset used; and "I," "D," and "K" refer to the ISBSG, Desharnais, and

Table 4 Estimation accuracy of multiple linear regression analysis (effort: 10%, 50%)

LO	TO	DS	Avg. *AE*	Med. *AE*	Avg. *MRE*	Med. *MRE*	Avg. *BRE*	Med. *BRE*	p-val. *MRE*	p-val. *BRE*
Y	N	I	-12.8	2.5	0.1%	-0.2%	0.7%	-2.2%	0.82	0.83
		D	-25.0	3.1	-0.3%	-0.3%	-0.2%	-0.1%	0.52	0.51
		K	4.6	-2.7	-0.4%	0.4%	-0.1%	-0.3%	**0.02**	0.72
N	Y	I	83.4	25.3	1.9%	0.5%	3.5%	2.4%	**0.02**	**0.01**
		D	81.2	72.9	1.0%	1.5%	1.9%	1.3%	0.38	0.16
		K	36.9	36.2	2.5%	2.2%	3.5%	3.9%	**0.00**	**0.00**
Y	Y	I	70.9	13.1	2.1%	0.2%	4.2%	0.3%	**0.03**	**0.01**
		D	56.2	66.7	0.9%	1.2%	1.8%	1.8%	0.63	0.70
		K	41.1	45.9	2.0%	2.2%	3.3%	3.0%	**0.00**	**0.01**

Table 5 Estimation accuracy of multiple linear regression analysis (effort: 10%, 100%)

LO	TO	DS	Avg. *AE*	Med. *AE*	Avg. *MRE*	Med. *MRE*	Avg. *BRE*	Med. *BRE*	p-val. *MRE*	p-val. *BRE*
Y	N	I	-37.0	7.7	-0.6%	-0.5%	0.6%	-1.4%	0.63	0.92
		D	71.7	35.0	2.7%	3.1%	4.6%	1.7%	0.11	0.16
		K	12.6	9.5	-0.2%	-0.3%	0.8%	-0.7%	0.45	0.12
N	Y	I	166.6	2.9	5.0%	0.6%	**9.7%**	0.1%	**0.00**	**0.00**
		D	228.3	119.4	**6.4%**	3.2%	**10.0%**	4.5%	**0.00**	**0.00**
		K	111.0	56.3	**5.3%**	1.6%	**8.8%**	3.7%	**0.01**	**0.00**
Y	Y	I	131.1	4.7	4.4%	0.1%	**10.2%**	-0.9%	**0.00**	**0.00**
		D	276.7	159.2	**8.2%**	**5.1%**	**13.0%**	**8.4%**	0.08	0.08
		K	124.0	67.4	**5.0%**	1.3%	**9.6%**	3.0%	**0.01**	**0.01**

Table 6 Estimation accuracy of multiple linear regression analysis (effort: 20%, 50%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	0.3	1.9	-0.2%	0.0%	-0.3%	-1.2%	0.63	0.65
		D	48.7	-40.5	-1.2%	2.3%	-0.4%	1.5%	0.83	0.43
		K	0.0	11.5	-0.3%	0.4%	-0.1%	-0.9%	0.43	0.93
N	Y	I	106.1	-0.7	4.0%	0.7%	**5.6%**	1.6%	**0.00**	**0.00**
		D	160.2	211.8	**5.0%**	**5.3%**	**6.9%**	**5.0%**	**0.00**	**0.00**
		K	37.5	33.2	4.5%	1.4%	**6.4%**	2.7%	**0.02**	**0.01**
Y	Y	I	106.7	-8.1	3.7%	0.8%	**5.1%**	0.4%	**0.00**	**0.00**
		D	191.8	184.0	3.1%	4.8%	**5.7%**	**8.2%**	0.08	0.08
		K	37.0	42.2	4.2%	1.7%	**6.3%**	2.2%	**0.03**	**0.01**

Table 7 Estimation accuracy of multiple linear regression analysis (effort: 20%, 100%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	33.0	-8.5	-0.7%	0.2%	-0.5%	-0.5%	1.00	1.00
		D	58.3	22.1	3.2%	2.1%	4.4%	2.8%	0.38	0.28
		K	-1.9	3.3	0.1%	0.6%	0.6%	-1.4%	0.64	0.49
N	Y	I	309.4	53.1	**12.3%**	2.5%	**20.2%**	**6.5%**	**0.00**	**0.00**
		D	317.9	217.4	**9.9%**	**5.0%**	**14.1%**	**5.7%**	**0.00**	**0.00**
		K	161.9	89.4	**6.9%**	5.4%	**12.0%**	**8.2%**	**0.00**	**0.00**
Y	Y	I	347.6	36.9	**11.4%**	2.9%	**19.3%**	**7.0%**	**0.00**	**0.00**
		D	401.0	166.0	**11.9%**	**6.9%**	**16.8%**	**9.1%**	**0.00**	**0.00**
		K	160.6	95.4	**6.9%**	4.3%	**12.7%**	**9.5%**	**0.00**	**0.00**

Table 8 Estimation accuracy on multiple linear regression analysis (FP, 10%, 50%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	2.6	9.9	0.8%	0.0%	1.4%	-0.1%	**0.02**	**0.03**
		D	-18.4	28.3	-1.8%	0.9%	-1.7%	1.2%	0.85	1.00
		K	3.8	5.2	0.3%	0.0%	0.1%	0.3%	**0.04**	0.82
N	Y	I	-4.2	28.2	0.9%	0.2%	1.6%	-0.2%	0.05	**0.01**
		D	31.5	69.3	0.8%	1.4%	1.2%	1.4%	0.20	0.30
		K	5.7	31.4	0.4%	0.8%	0.9%	0.7%	0.49	0.24
Y	Y	I	-1.5	29.1	1.6%	0.2%	3.0%	-0.2%	**0.01**	**0.01**
		D	-7.0	65.4	-1.2%	1.9%	-1.1%	1.8%	0.85	1.00
		K	8.8	32.0	0.6%	1.1%	0.9%	1.1%	0.29	0.23

Table 9 Estimation accuracy on multiple linear regression analysis (FP, 10%, 100%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	16.4	12.9	0.5%	0.0%	0.5%	0.8%	0.28	0.56
		D	105.2	140.3	4.7%	**5.5%**	**5.1%**	**5.5%**	0.28	0.38
		K	10.1	5.9	0.5%	-0.1%	0.3%	0.6%	0.09	0.50
N	Y	I	26.7	7.2	2.0%	0.3%	3.9%	0.8%	0.06	**0.01**
		D	189.2	128.4	**5.8%**	3.2%	**9.5%**	3.9%	**0.00**	**0.00**
		K	-2.4	10.4	0.8%	0.6%	1.7%	0.4%	**0.04**	**0.01**
Y	Y	I	41.7	24.3	2.3%	0.7%	4.1%	1.9%	0.07	**0.02**
		D	187.4	192.8	**6.7%**	**7.1%**	**8.3%**	**7.2%**	0.11	0.13
		K	8.3	9.3	1.3%	0.8%	1.9%	0.8%	0.06	**0.01**

Table 10 Estimation Accuracy on multiple linear regression analysis (FP, 20%, 50%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	5.1	14.4	0.4%	0.1%	0.3%	0.0%	0.06	0.19
		D	43.0	68.0	1.2%	3.6%	1.4%	2.6%	1.00	0.92
		K	1.5	-1.1	0.2%	0.6%	-0.2%	0.1%	0.56	0.66
N	Y	I	-0.3	13.3	1.5%	0.4%	1.7%	0.1%	0.07	0.22
		D	80.8	84.7	1.9%	2.4%	3.9%	2.8%	**0.05**	**0.01**
		K	32.3	42.4	1.8%	2.3%	3.0%	4.9%	0.08	**0.02**
Y	Y	I	4.4	19.1	1.8%	0.4%	1.9%	0.4%	**0.02**	0.08
		D	87.7	92.0	1.6%	4.8%	2.7%	2.7%	0.77	0.49
		K	31.9	37.8	1.9%	2.6%	2.6%	4.6%	0.06	**0.03**

Table 11 Estimation accuracy on multiple linear regression analysis (FP, 20%, 100%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	32.9	0.2	2.5%	0.7%	3.3%	1.4%	0.01	0.01
		D	71.3	94.2	-0.1%	3.2%	1.4%	1.0%	0.94	0.74
		K	8.8	10.2	0.9%	1.0%	0.4%	0.9%	0.20	0.59
N	Y	I	101.4	75.4	**5.8%**	0.9%	**9.8%**	2.2%	**0.00**	**0.00**
		D	304.4	187.2	**6.8%**	**5.1%**	**13.7%**	**7.0%**	**0.03**	**0.01**
		K	58.6	19.1	**5.1%**	-0.1%	**7.9%**	0.9%	**0.01**	**0.00**
Y	Y	I	104.4	54.4	**7.7%**	1.3%	**11.7%**	3.6%	**0.00**	**0.00**
		D	175.6	161.6	2.3%	**5.8%**	**6.2%**	**6.2%**	0.23	0.20
		K	57.4	16.8	**5.2%**	1.1%	**7.0%**	1.3%	**0.00**	**0.00**

Table 12 Estimation accuracy of analogy based estimation (effort: 10%, 50%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	-38.2	23.7	0.1%	0.5%	-0.3%	1.3%	0.43	0.72
		D	-4.6	3.9	0.1%	0.6%	-0.3%	-1.1%	0.87	0.77
		K	13.5	-5.7	1.0%	1.3%	0.9%	0.0%	0.91	0.98
N	Y	I	50.4	31.5	3.3%	0.7%	4.2%	3.4%	**0.05**	**0.01**
		D	69.8	78.8	1.3%	1.1%	1.9%	1.9%	0.20	0.14
		K	27.7	15.8	2.2%	1.9%	2.9%	1.7%	**0.03**	**0.02**
Y	Y	I	11.4	43.6	3.5%	1.2%	4.0%	3.4%	0.28	0.16
		D	62.2	60.1	1.4%	1.8%	1.7%	0.6%	0.25	0.32
		K	40.9	13.8	3.4%	2.5%	3.9%	0.5%	0.15	0.19

Table 13 Estimation accuracy of analogy based estimation (effort: 10%, 100%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	203.0	77.7	**11.7%**	0.9%	**10.9%**	2.0%	**0.00**	**0.00**
		D	93.6	33.8	3.2%	1.8%	3.3%	2.2%	**0.02**	**0.04**
		K	60.7	40.2	1.4%	1.3%	2.3%	0.8%	0.34	0.19
N	Y	I	162.9	53.1	**10.5%**	0.8%	**12.7%**	1.5%	**0.00**	**0.00**
		D	205.8	134.9	**5.5%**	3.5%	**7.9%**	4.5%	**0.01**	**0.01**
		K	101.4	44.6	**5.8%**	2.7%	**7.8%**	3.2%	**0.01**	**0.00**
Y	Y	I	366.6	103.4	**23.3%**	2.1%	**24.6%**	3.8%	**0.00**	**0.00**
		D	299.5	247.6	**9.2%**	**5.4%**	**11.7%**	**7.9%**	**0.01**	**0.01**
		K	161.1	67.0	**7.2%**	3.7%	**10.4%**	4.3%	**0.02**	**0.01**

Kitchenham datasets, respectively. A difference of over 5% between the *MRE* and *BRE* is denoted in bold. Note that the information presented in Tables 4–19 is NOT based on log-transformations. Before calculation, inverse log transformations were applied to the values.

In Tables 4–19, the columns "p-val. *MRE*" denote the p-value of the difference between the baseline and other cases for the average *MRE*, derived using the Wilcoxon signed-rank test. Similarly, the columns "p-val. *BRE*" contain the p-values for the average *BRE*. P-values smaller than 0.05 are denoted in bold. As shown in the tables, when the difference in the *MRE* and *BRE* between the baseline and other cases exceeds 5%, the p-values of the average *MRE* and *BRE* (derived using the Wilcoxon signed-rank test) are below 0.05 in many cases, except that in which the learning dataset included outliers and the test dataset did not (Table 17). Thus, we only considered the differences in *MRE* and *BRE* for the following analyses.

5.2 Influence on Multiple Linear Regression Analysis

Before explaining the results, we illustrate the influence of

Table 14 Estimation accuracy of analogy based estimation (effort: 20%, 50%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	69.4	32.0	2.9%	1.5%	2.6%	2.6%	0.70	0.70
		D	42.5	90.6	1.4%	-0.1%	1.7%	1.7%	0.11	0.14
		K	25.2	7.8	2.3%	0.9%	3.1%	-1.5%	0.06	**0.03**
N	Y	I	106.2	73.0	**7.6%**	0.6%	**8.6%**	2.4%	**0.01**	**0.00**
		D	123.4	134.9	4.8%	2.1%	**6.1%**	2.9%	**0.00**	**0.00**
		K	41.5	25.0	**5.7%**	2.1%	**7.2%**	3.6%	**0.01**	**0.00**
Y	Y	I	176.9	95.3	**10.9%**	2.6%	**11.6%**	3.2%	**0.05**	**0.05**
		D	164.1	140.7	**6.2%**	1.4%	**8.0%**	4.0%	**0.00**	**0.00**
		K	57.0	27.8	**7.8%**	1.7%	**10.0%**	0.8%	**0.01**	**0.00**

Table 15 Estimation accuracy of analogy based estimation (effort: 20%, 100%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	124.2	64.3	**13.7%**	2.1%	**11.7%**	3.8%	0.13	0.13
		D	101.1	173.6	4.0%	-0.7%	3.8%	1.1%	0.08	0.24
		K	111.0	86.2	**15.5%**	2.5%	**15.6%**	2.6%	**0.00**	**0.00**
N	Y	I	277.8	103.9	**20.3%**	2.9%	**24.9%**	**6.9%**	**0.00**	**0.00**
		D	262.9	157.8	**10.6%**	4.4%	**15.1%**	**5.3%**	**0.00**	**0.00**
		K	141.5	56.7	**10.0%**	**6.1%**	**13.6%**	**6.9%**	**0.00**	**0.00**
Y	Y	I	391.6	123.9	**34.9%**	4.6%	**37.2%**	**11.6%**	**0.00**	**0.00**
		D	372.8	253.6	**15.1%**	2.1%	**19.2%**	4.7%	**0.01**	**0.01**
		K	233.5	147.9	**24.0%**	**5.9%**	**27.2%**	**6.0%**	**0.00**	**0.00**

Table 16 Estimation accuracy of analogy based estimation (FP: 10%, 50%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	33.3	12.0	1.9%	0.5%	1.5%	2.4%	0.08	0.08
		D	9.4	12.2	0.6%	0.0%	0.5%	0.2%	0.29	0.52
		K	-7.2	4.4	-0.1%	0.5%	-0.5%	-1.5%	0.92	0.52
N	Y	I	50.1	50.0	1.8%	0.4%	2.2%	0.5%	0.06	**0.02**
		D	9.2	0.6	0.7%	0.6%	0.5%	1.0%	0.13	0.36
		K	5.5	6.2	0.3%	0.3%	0.6%	0.3%	0.71	0.56
Y	Y	I	81.7	25.4	3.7%	0.8%	3.8%	2.6%	**0.01**	**0.00**
		D	27.6	33.8	1.4%	0.3%	1.2%	1.0%	**0.01**	**0.04**
		K	-5.6	8.0	-0.1%	-0.5%	-0.2%	-0.4%	0.91	0.92

Table 17 Estimation accuracy of analogy based estimation (FP: 10%, 100%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	182.5	107.8	**10.2%**	2.0%	**9.9%**	2.5%	0.49	0.49
		D	26.1	79.3	1.2%	0.5%	1.5%	-0.4%	**0.04**	**0.04**
		K	18.6	-25.1	2.2%	2.2%	2.3%	1.6%	0.14	0.16
N	Y	I	58.0	20.9	1.7%	1.0%	2.8%	0.6%	0.23	0.13
		D	55.8	46.8	2.1%	0.6%	2.4%	0.4%	**0.02**	**0.02**
		K	1.7	-4.6	1.0%	0.1%	1.5%	0.4%	0.36	0.23
Y	Y	I	223.5	96.9	**10.9%**	2.3%	**11.6%**	3.4%	0.32	0.23
		D	76.3	129.6	3.0%	0.4%	3.7%	-0.4%	**0.01**	**0.01**
		K	23.2	-25.0	2.8%	1.4%	3.4%	1.8%	0.13	0.13

outliers in the dependent variable. We assume that the learning data include outliers in the dependent variable and that the effort of project A is 1,000 person–hours (100 is the correct value). When the effort of Project B (which is similar to Project A) is estimated by analogy based estimation, the estimated value erroneously indicates 1,000 person–hours. Moreover, we assumed that it is probable that the effort of the estimation target project is an outlier. Without considering this, the evaluations of outlier influence are optimistic. Therefore, we evaluated the case in which the test data include outliers in the dependent variable.

Table 18 Estimation accuracy of analogy based estimation (FP: 20%, 50%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	214.1	55.5	**18.0%**	1.1%	**17.5%**	2.6%	**0.00**	**0.00**
		D	-33.4	59.3	-0.2%	-0.5%	-0.3%	2.2%	0.26	0.38
		K	4.9	-35.7	-0.4%	1.2%	-0.9%	-1.6%	1.00	0.92
N	Y	I	-4.0	29.0	0.3%	0.1%	0.7%	1.4%	0.68	0.54
		D	21.0	27.4	1.1%	0.9%	1.6%	0.3%	**0.02**	0.08
		K	37.3	9.9	3.1%	0.8%	4.2%	1.1%	**0.01**	**0.02**
Y	Y	I	209.9	81.8	**18.2%**	1.5%	**18.0%**	4.5%	**0.01**	**0.01**
		D	-6.2	79.2	0.7%	0.3%	1.2%	1.5%	0.63	0.32
		K	41.4	-35.0	1.8%	1.2%	2.3%	-0.5%	0.38	0.32

Table 19 Estimation accuracy of analogy based estimation (FP: 20%, 100%)

LO	TO	DS	Avg.	Med.	Avg.	Med.	Avg.	Med.	p-val.	p-val.
			AE	*AE*	*MRE*	*MRE*	*BRE*	*BRE*	*MRE*	*BRE*
Y	N	I	54.3	132.1	**12.1%**	1.3%	**11.6%**	3.1%	0.08	0.08
		D	78.7	88.0	1.5%	2.8%	2.2%	4.4%	0.05	**0.02**
		K	51.0	8.2	**7.9%**	0.4%	**6.8%**	-0.6%	0.06	0.13
N	Y	I	187.5	41.5	**6.7%**	0.9%	**9.1%**	2.4%	**0.01**	**0.00**
		D	84.1	115.9	1.9%	1.5%	2.8%	2.6%	**0.01**	**0.01**
		K	46.4	19.5	4.6%	2.8%	**6.1%**	0.7%	**0.05**	**0.03**
Y	Y	I	252.5	151.3	**19.3%**	1.6%	**20.9%**	**6.8%**	**0.01**	**0.01**
		D	150.1	216.2	3.9%	4.0%	**5.2%**	**5.1%**	**0.01**	**0.00**
		K	121.3	13.3	**15.5%**	2.1%	**15.3%**	1.4%	**0.05**	0.08

Adding outliers to dependent variables: First, we consider the results when the outlier percentage was 10%. When the extent of outliers was 50% (see Table 4), the degradation of the *MRE* and *BRE* was less than 5% for all datasets, regardless of the location of outliers. In particular, the degradation was very small when the learning dataset included outliers and the test dataset did not include (additional) outliers. In contrast, when the extent of outliers was 100% (see Table 5), the degradation of the average *BRE* was approximately 10%, except when the learning dataset included outliers and the test dataset did not.

Next, we consider the results when the outlier percentage was 20%. When the extent of outliers was 50% (see Table 6), the degradation of the average *BRE* was approximately 5%, except when the learning dataset included outliers and the test dataset did not. When the extent of outliers was 100% (see Table 7), the degradation of the average *BRE* was approximately 15–20%, except when the learning dataset included outliers and the test dataset did not.

The results suggest that the influence of outliers is non-negligible when the extent of outliers is 100% or the outlier percentage is 20%. In contrast, when the extent of outliers is 50% and the outlier percentage is 10%, the influence of outliers is small.

Adding outliers to independent variables: Overall, the degradation of the *MRE* and *BRE* was small compared to the case in which outliers were added to the effort. First, we consider the results for when the outlier percentage was 10%. When the extent of outliers was 50% (see Table 8), the degradation of the *MRE* and *BRE* was less than 5%, regardless of the location of outliers. When the extent of outliers was 100% (see Table 9), the degradation was larger. In particular, the degradation in the average *BRE* exceeded 5% on the Desharnais dataset. These results indicate that when the extent of outliers is 50% and the outlier percentage is 10%, the influence of outliers is small.

Next, we consider the results for when the outlier percentage was 20%. When the extent of outliers was 50% (see Table 10), the degradation of the *MRE* and *BRE* was less than 5%, regardless of the location of outliers. When the extent of outliers was 100% (see Table 11), the degradation of the average *BRE* was approximately 5%, except when the learning dataset included outliers and the test dataset did not.

Therefore, we suggest that when FP is suspected to include outliers, they should be focused upon. However, if the extent and number of outliers are not large (i.e., if the percentage is below 20% and the extent is below 50%), the influence of outliers can be neglected.

5.3 Influence on Analogy Based Estimation

Adding outliers to dependent variables: First, we consider the results when the outlier percentage was 10%. When the extent of outliers was 50% (see Table 12), the degradation of the *MRE* and *BRE* was less than 5%, regardless of the location of outliers. However, when the extent of outliers was 100% (see Table 13), the estimation accuracy degraded. In particular, the average *MRE* and *BRE* degraded by more than 10% on the ISBSG dataset.

Next, we consider the results when the outlier percentage of outliers was 20%. When the extent of outliers was 50% (see Table 14), the average *BRE* degraded by approximately 10%, except when the learning dataset included outliers and the test dataset did not. When the extent of outliers was 100% (see Table 15), the degradation of the average *BRE* exceeded 10%, except for the Desharnais dataset when the learning dataset included outliers and the test dataset did not.

Therefore, the influence of outliers is non-negligible, especially when the extent of outliers is large (e.g., 100%) or the outlier percentage is large (e.g., 20%) for dependent variables.

Adding outliers to independent variables: First, we consider the results when the outlier percentage was 10%. When the extent of outliers was 50% (see Table 16), the degradation of the *MRE* and *BRE* was very small; that is, less than 5%. When the extent of outliers was 100% (see Table 17), the estimation accuracy degraded, especially for the ISBSG dataset when the learning dataset contained outliers. The degradation was approximately 10%; however, it was not found statistically significant.

Next, we consider the results when the percentage of outliers was 20%. When the extent of outliers was 50% (see Table 18), the degradation of the average *MRE* and average *BRE* exceeded 10% for the ISBSG dataset, except when the learning dataset contained outliers. When the extent of outliers was 100% (see Table 19), the average *BRE* degraded by approximately 5–20% for all datasets except for the Desharnais dataset.

Overall, the degradation of the *MRE* and *BRE* was small compared to the case in which outliers were added to the effort. The results suggest that when analogy based estimation is applied to effort estimation, outliers should be considered for software size and estimation effort when the outlier percentage is large (e.g., 20%). When the extent of outliers is large (e.g., 100%) in the dependent variable, outliers should also be considered.

5.4 Comparison of Linear Regression Analysis and Analogy Based Estimation

Overall, when both learning and test datasets contained (additional) outliers, the estimation accuracy was lowest, regardless of which variable included outliers and the estimation method employed. In contrast, when the learning dataset contained outliers and the test dataset did not, the degradation of estimation accuracy was relatively small. Moreover, when outliers were included in the dependent variable (i.e., effort), the degradation in estimation accuracy was larger than when they were present in the independent variable (i.e., FP), regardless of the estimation method used. However, when the extent and percentage of outliers were not large (i.e., 50% and 10%, respectively), the outliers could be neglected, regardless of which variable included outliers and the estimation method used.

When outliers were added to the independent variable and the percentage was 10%, the accuracy degradation of the analogy based estimation did not differ considerably from that of the linear regression analysis (e.g., the degradation of the average *MRE* and *BRE* exceeded 5% for one out of three datasets, as shown in Tables 9 and 17; furthermore, the degradation was not statistically significant in Table 17). However, when the outlier percentage was 20%, this degradation exceeded that of the linear regression analysis (e.g., the degradation of the average *BRE* was larger than that of the linear regression analysis for the ISBSG dataset, as shown in Tables 10 and 18).

When outliers were added to the dependent variable, when the outlier percentage was 20%, or when the extent of outliers was 100%, the degradation of the analogy based estimation exceeded that of the linear regression analysis. For example, considering the ISBSG dataset in Tables 5 and 13 (i.e., where the outlier percentage is 10% and the extent of outliers is 100%), the degradation of the average *BRE* exceeded 20% when both the learning and test datasets included outliers, whereas this degradation was approximately 10% for linear regression analysis. Similarly, in Tables 6 and 14 (i.e., where the percentage is 20% and the extent is 50%), the degradation of the average *BRE* was approximately 10%; however, it was approximately 5% for linear regression analysis, except when the learning dataset included outliers and the test dataset did not.

These results suggest that when analogy based estimation is applied to effort estimation and the extent or percentage of outliers is considered large, the outliers should be considered.

6. Discussion

6.1 Experimental Results

In the experimental results, when the extent of outliers was 100%, the estimation accuracy was more affected when outliers were present only in test dataset than when they were present only in the learning dataset (except for the conditions given in Table 17). This is because outliers in the test dataset directly impact the estimation accuracy. For example, if the outlier percentage is 100% and the extent of outliers is 100% in the test dataset, the *BRE* increases by approximately 100%. In contrast, outliers in the learning dataset do not significantly affect the regression model and are not always used (i.e., not always included in the k-nearest neighborhoods) in analogy based estimation. Thus, the influence of outliers is larger in the test dataset than in the learning dataset.

When outliers are included in the independent variable (i.e., FP), the degradation in estimation accuracy is smaller than when they are included in the dependent variable, especially when the extent and percentage of outliers are 100% and 20%, respectively. This may be because effort is not estimated solely based on FP, but it considers other variables that suppress the influence of outliers. More specifically, regression analysis estimates the effort using a linear combination of independent variables, and analogy based estimation uses independent variables in similarity computations and a size-adjustment method [see Eqs. (3)–(5)].

In some cases, linear regression analysis is less affected by outliers than analogy-based estimation. When performing linear regression analysis, a logarithmic transformation is applied to reduce the extent of outliers. In contrast, the size-adjustment method (see Eqs. (4) and (5)) directly affects the extent of outliers. This may be the reason why analogy based estimation is sometimes more affected by outliers.

6.2 Utilizing the Results

Here, we discuss how to utilize these results in practical software development. When the effort of software development is estimated for practical use, there are two ways to avoid the influence of outliers:

- Measure and collect data precisely to suppress the inclusion of outliers.
- Remove data points suspected as outliers.

The removal of outliers may be inexpensive if a mathematical outlier elimination method is applied. On the contrary, precise data measurements can be costly, discouraging people from collecting data. Our experimental results suggest that the influence of outliers is small in certain cases (i.e., the extent of outliers is small); hence, it is not necessary to remove them. In addition, the balance between measurement precision and estimation accuracy should be considered; for example, say a task (e.g., the troubleshooting of past projects) interrupts a main task (e.g., development); to measure effort precisely, the time it requires should not be included with that of the main task. Furthermore, the time required for short meetings and breaks should be measured. However, excessive measurement rules leave a mental burden on practitioners. This suggests that we should not be extensively concerned about outliers. In fact, in the experimental dataset, the estimation errors (e.g., the average *BRE*) did not notably degrade, even when FP and effort included outliers with 10% and 50% percentage and extent, respectively.

The elimination of statistical outliers is still important. Even if outlier elimination reduces the estimation accuracy (i.e., a higher *BRE*), elimination is still the correct procedure if the outliers have undue statistical influence. Additionally, when the extent of outliers exceeds 100% or the outlier percentage exceeds 20%, outlier elimination should be considered. Our suggestion is as follows: in practical software development, collecting and analyzing data is important but outliers need not be excessively guarded against.

Practitioners (e.g., project managers) consider the influence of outliers on the estimation accuracy before constructing an estimation model, by considering the outlier parameters (e.g., the number of outliers). Notably, not all data points need to be checked to determine the parameters. The parameters can be roughly speculated to collect some data points (i.e., sampling) and then checked. In addition, it is difficult to eliminate data points when the extent of outliers is 50%, because these outliers do not notably differ from other data points. To eliminate such data points, precise data measurements are needed; however, this requires a non-negligible effort. In contrast, our results suggest that if the number of outliers is not large, they need not be eliminated to enhance the estimation accuracy.

6.3 Outliers in Test Dataset

This subsection discusses reasons for considering outliers in a test dataset. In essence, we cannot remove outliers from test datasets because they are estimation target projects; therefore, eliminating the projects means we do not estimate their effort. As explained in Sect. 1, measurement errors are a cause of outliers. In Kemerer's experiment (evaluated interrater reliability of FP) [16], the maximum measurement error (difference in FP between raters) exceeded 100%. Thus, it is probable that the FP includes 50% measurement errors in some projects. We assume that effort is estimated using the following equation:

$$\text{effort} = \exp(3.3 + 0.7 \ln(\text{function points})). \quad (11)$$

In the above equation, $\exp(n)$ represents the exponential function, and $\ln(n)$ represents the natural logarithm function. If FP is accurately measured as 100, the effort estimated using the equation is 681; however, if it is inaccurately measured as 150, the estimated effort becomes 905. Even if the estimation error of the equation is small for accurate FP, the error increases with inaccurate FP.

Outliers can also occur when an exceptional amount of rework occurs on a project and consumes more effort, as explained in Sect. 1. We assume that the actual effort of an ideal project should be 500 without such rework; however, the effort increases to 750 in the case of an exceptional amount of rework. In such cases, even if a model can accurately estimate the project effort as 500, the estimation error (i.e., the difference between the estimated and actual effort) of the model is enlarged.

Expressed otherwise, the influence of outliers in a test dataset is inevitable, and our experiment shows that the degradation of estimation accuracy is non-negligible when the test dataset contains outliers of a certain extent. We assume the case in which a software development company sets contingency reserves (i.e., a reserve budget for unexpected costs [32]) by considering the estimation accuracy of a model; this accuracy is derived by assuming that the test dataset does not contain outliers. In this case, some projects in the company would face a shortage of contingency reserves, due to an optimistic allocation thereof. Additionally, the outliers in test datasets may affect the results of comparative effort prediction studies such as that conducted in [9].

6.4 Threats to Validity

Internal validity: When the percentage and extent of outliers were 10% and 50%, respectively, the influence of outliers on the estimation accuracy was small. It is possible that the selected data points, transformed into outliers, did not affect the accuracy by accident. If we use 10- or 5-fold cross-validation, the likelihood of this is fairly large because the number of data points in the test dataset is small; hence, the number of outliers in the dataset is also small. In contrast, when we applied a hold-out method, the number of outliers in the test dataset was not small. For example, approximately 30 out of 300 (i.e., 10%) data points were selected when using the ISBSG dataset. We performed this experiment ten times. It is possible that none of the 30 projects selected had any effect on the estimation accuracy in any experiment; however, this probability is considered to be very low. Thus, the result has internal validity.

External validity: In the experiment, the three datasets we used had varying specifications. For example, the ages of the datasets were different: in the ISBSG dataset, over half of the projects were conducted in the period 1998–2004, whereas the Desharnais dataset was collected during the 1980s. Furthermore, the number of data points differed: 611 we used for the ISBSG dataset and 135 for the Kitchenham dataset. Moreover, the ISBSG dataset was collected from various companies, whereas the Desharnais dataset was collected from a single company. Therefore, our suggestions, based on the analysis results, are considered applicable to other datasets.

7. Related Work

Several studies have attempted to define guidelines for constructing an effort estimation model. For example, Mendes et al. [24] evaluated an effort estimation model by using a cross-company dataset (i.e., data collected from various companies), to verify its effectiveness when building an effort estimation model. Strike et al. [42] compared the performances of missing-value imputation methods in effort estimation models, to confirm which imputation methods were effective at enhancing the estimation accuracy. Furthermore, Azzeh [3] compared various size-adjustment methods using analogy-based estimation (see Sect. 2.2), to clarify which methods were most effective in enhancing the estimation accuracy.

A small number of studies evaluated the performance of outlier elimination methods, to help select methods. Seo et al. [35] proposed the application of least trimmed squares- and *k*-means-based elimination; to evaluate the method performances, they used a linear regression model, neural network, and Bayesian network to estimate effort after outlier elimination. Seo et al. [34] also evaluated five outlier elimination methods using two effort estimation methods. They demonstrated that the estimation accuracy was not statistically different between estimations made with and without elimination methods. We also focus on the outliers in this study; however, our viewpoint differs from those of previous studies [34], [35], which have predominately focused on outlier elimination methods. In contrast, we focus on the influence of outliers. That is, our study relates to data collection and the importance (NOT selection) of outlier elimination methods.

Shepperd et al. [36] compared the estimation accuracies of various methods (e.g., analogy based estimation and multiple linear regression) by considering characteristics of the dataset. To prepare datasets with different characteristics, they fabricated datasets experimentally, to generate multicollinearity, outliers, and other features. Although they added outliers in the independent variables, they did not evaluate the influence of outliers by considering their various aspects, as performed here (see Sect. 3.2). For instance, they did not alter the outlier locations.

8. Conclusions

To clarify the influence of outliers on effort estimation, we experimentally added outliers to datasets and evaluated the estimation accuracy. In the analysis, we considered (1) the percentage of outliers, (2) the extent of outliers, (3) the variables containing outliers, and (4) the locations of outliers. We used multiple linear regression analysis and analogy based estimation as the effort estimation methods. The analysis results are as follows:

- When the learning dataset contained outliers but the test dataset did not, the degradation in estimation accuracy was relatively small, regardless of the variables containing outliers and the estimation method.
- When the outliers were included in the dependent variable (i.e., effort), the degradation of estimation

accuracy was larger than that achieved when they were included in the independent variable (i.e., FP), regardless of the estimation method.

- When the extent and percentage of outliers were not large (i.e., 50% and 10%, respectively), outliers could be neglected, regardless of the variables containing outliers and the estimation method.
- When analogy based estimation was applied to effort estimation and the extent or percentage of outliers was considered large (i.e., 100% and 20%, respectively), the influence of outliers was large.

Intuitively, the results show that the influence of outliers in software size and effort is negligible, unless the percentage or extent of outliers is extremely large.

We do not maintain that outlier elimination is unnecessary. Particularly, when the extent of outliers exceeds 100% or the outlier percentage exceeds 20%, outlier elimination should be considered. Our suggestion is that the degradation of estimation accuracy is small if no variables include extreme outliers. In short, it would be beneficial to focus on collecting data in practical software development. Notably, the experimental scope only covered 10–20% of the data points, whereas the other 80–90% were still considered to be "correctly measured." Thus, practitioners should not neglect the influence of outliers.

Our solutions in practical software development are as follows:

- Collecting and analyzing data is important but outliers need not be excessively guarded against.
- The balance between measurement precision and estimation accuracy should be considered.
  - ➢ Excessive measurement rules leave a mental burden on practitioners.
- The elimination of statistical outliers is still important,
  - ➢ especially when the extent of outliers exceeds 100% or the outlier percentage exceeds 20%.
  - ➢ Practitioners consider the influence of outliers before constructing an estimation model, by considering the outlier parameters.
  - ➢ The parameters can be roughly speculated by sampling, and then checked.
- If the collected data is of low quality, it is worthwhile to attempt to mathematically estimate effort.

The contributions of our paper are to show how characteristics of outliers affect the estimation accuracy, and to explain how to address the outliers in practical software development, as explained above. Although the solutions are not very clear, it is difficult to provide clear solutions for the outliers. For example, although Seo et al. [34] suggested that it is necessary to consider the outlier elimination and to conduct a detailed analysis of the effort estimation results to improve the estimation accuracy in software organizations, they did not provide clear solutions. Providing such solutions is one of our future works.

In future, we will alter the parameters of the outliers (e.g., the percentage and extent) for other datasets, and we hope to clarify the relationship between these parameters and the estimation accuracy.

Acknowledgments

This research was partially supported by the Japan Society for the Promotion of Science (JSPS) [Grants-in-Aid for Scientific Research (A) (No.17H00731)]

References

[1] A. Abran, I. Ndiaye, and P. Bourque, "Evaluation of a Black-Box Estimation Tool: A Case Study," Software Process: Improvement and Practice, Wiley, vol.12, no.2, pp.199–218, 2007.

[2] S. Amasaki and C. Lokan, "A Replication of Comparative Study of Moving Windows on Linear Regression and Estimation by Analogy," Proc. International Conference on Predictive Models and Data Analytics in Software Engineering (PROMISE), Article 6, pp.1–10, 2015.

[3] M. Azzeh, "A replicated assessment and comparison of adaptation techniques for analogy-based effort estimation," Empirical Software Engineering, vol.17, no.1-2, pp.90–127, 2012.

[4] M. Basgalupp, R. Barros, and D. Ruiz, "Predicting software maintenance effort through evolutionary-based decision trees," Proc. Annual ACM Symposium on Applied Computing (SAC), pp.1209–1214, 2012.

[5] B. Boehm, Software Engineering Economics, Prentice Hall, 1981.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Outlier detection: a survey," ACM Computing Surveys, vol.41, no.3, Article No.15, July 2009.

[7] S. Conte, H. Dunsmore, and V. Shen, Software Engineering, Metrics and Models, Benjamin/Cummings, 1986.

[8] A. Corazza, S. Di Martino, F. Ferrucci, C. Gravino, and E. Mendes, "Investigating the use of Support Vector Regression for web effort estimation," Empirical Software Engineering, vol.16, pp.211–243, 2011.

[9] K. Dejaeger, W. Verbeke, D. Martens, and, B. Baesens, "Data Mining Techniques for Software Effort Estimation: A Comparative Study," IEEE Trans. Softw. Eng., vol.38, no.2, pp.375–397, 2012.

[10] J. Desharnais, Analyse Statistique de la Productivitie des Projets Informatique a Partie de la Technique des Point des Function, Master Thesis, University of Montreal, 1989.

[11] M. Harman, S. Mansouri, and Y. Zhang, "Search-based software engineering: Trends, techniques and applications," ACM Computing Surveys, vol.45, no.1, article11, 2012.

[12] F. Heemstra, "Software cost estimation," Information and Software Technology, vol.34, no.10, pp.627–639, 1992.

[13] J. Huang, Y. Li, and M. Xie, "An empirical analysis of data preprocessing for machine learning-based software cost estimation," Information and Software Technology, vol.67, pp.108–127, 2015.

[14] International Software Benchmarking Standards Group (ISBSG), ISBSG Estimating: Benchmarking and research suite, ISBSG, 2004.

[15] M. Jørgensen and M. Shepperd, "A Systematic Review of Software Development Cost Estimation Studies," IEEE Trans. Softw. Eng., vol.33, no.1, pp.33–53, 2007.

[16] C. Kemerer, "Reliability of function points measurement: a field experiment," Commununications of ACM, vol.36, no.2, pp.85–97, 1993.

[17] J. Keung, B. Kitchenham, and R. Jeffery, "Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation," IEEE Trans. Softw. Eng., vol.34, no.4, pp.471–484, 2008.

[18] C. Kirsopp, E. Mendes, R. Premraj, and M. Shepperd, "An Empirical Analysis of Linear Adaptation Techniques for Case-Based

Prediction," Proc. International Conference on Case-Based Reasoning, pp.231–245, 2003.

[19] B. Kitchenham and E. Mendes, "Why comparative effort prediction studies may be invalid," Proc. International Conference on Predictor Models in Software Engineering (PROMISE), art.4, p.5, 2009.

[20] B. Kitchenham, S. Pfleeger, B. McColl, and S. Eagan, "An Empirical Study of Maintenance and Development Estimation Accuracy," Journal of Systems and Software, vol.64, no.1, pp.57–77, 2002.

[21] E. Kocaguneli, T. Menzies, and J. Keung, "On the Value of Ensemble Effort Estimation," IEEE Trans. Softw. Eng., vol.38, no.6, pp.1403–1416, 2012.

[22] C. Lokan, "What Should You Optimize When Building an Estimation Model?," Proc. International Software Metrics Symposium (METRICS), p.34, 2005.

[23] C. Lokan and E. Mendes, "Cross-company and single-company effort models using the ISBSG Database: a further replicated study," Proc. International Symposium on Empirical Software Engineering (ISESE), pp.75–84, 2006.

[24] E. Mendes, S. Martino, F. Ferrucci, and C. Gravino, "Cross-company vs. single-company web effort models using the Tukutuku database: An extended study," The Journal of Systems and Software, vol.81, no.5, pp.673–690, 2008.

[25] E. Mendes, N. Mosley, and S. Counsell, "A Replicated Assessment of the Use of Adaptation Rules to Improve Web Cost Estimation," Proc. International Symposium on Empirical Software Engineering (ISESE), pp.100–109, 2003.

[26] T. Menzies, E. Kocaguneli, B. Turhan, L. Minku, and F. Peters, "Sharing Data and Models in Software Engineering," p.406, Morgan Kaufmann Publishers Inc., 2014.

[27] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki, "Robust Regression for Developing Software Estimation Models," Journal of Systems and Software, vol.27, no.1, pp.3–16, 1994.

[28] R. Moazeni, D. Link, and B. Boehm, "COCOMO II parameters and IDPD: bilateral relevances," Proc. International Conference on Software and System Process (ICSSP), pp.20–24, 2014.

[29] K. Mølokken-Østvold and M. Jørgensen, "A Comparison of Software Project Overruns-Flexible versus Sequential Development Models," IEEE Trans. Softw. Eng., vol.31, no.9, pp.754–766, 2005.

[30] A. Monden and K. Kobayashi, "The Effect of Log Transformation in Multivariate Liner Regression Models for Software Effort Prediction," Computer Software, vol.27, no.4, pp.4_234–4_239, 2010 (in Japanese).

[31] H. Origasa, "How to comfirm normality," Japanese Pharmacology Therapeitics, vol.45, no.12, pp.1993–1995, 2017 (in Japanese).

[32] Project Management Institute, A Guide to the Project Management Body of Knowledge: PMBOK Guide Fourth Edtion, Project Management Institute, 2008.

[33] F. Sarro, A. Petrozziello, and M. Harman, "Multi-objective software effort estimation," Proc. International Conference on Software Engineering (ICSE), pp.619–630, 2016.

[34] Y. Seo and D. Bae, "On the value of outlier elimination on software effort estimation research," Empirical Software Engineering, vol.18, no.4, pp.659–698, 2013.

[35] Y. Seo, K. Yoon, and D. Bae, "An Empirical Analysis of Software Effort Estimation with Outlier Elimination," Proc. international workshop on Predictor models in software engineering (PROMISE), pp.25–32, 2008.

[36] M. Shepperd and G. Kadoda, "Comparing Software Prediction Techniques Using Simulation," IEEE Trans. Softw. Eng., vol.27, no.11, pp.1014–1022, 2001.

[37] M. Shepperd and S. MacDonell, "Evaluating prediction systems in software project estimation," Information and Software Technology, vol.54, no.8, pp.820–827, 2012.

[38] M. Shepperd and C. Schofield, "Estimating software project effort using analogies," IEEE Trans. Softw. Eng., vol.23, no.12, pp.736–743, 1997.

[39] B. Sigweni, M. Shepperd, and T. Turchi, "Realistic assessment of software effort estimation models," Proc. International Conference on Evaluation and Assessment in Software Engineering (EASE), no.41, p.6, 2016.

[40] R. Silhavy, P. Silhavy, and Z. Prokopova, "Evaluating subset selection methods for use case points estimation," Information and Software Technology, vol.97, pp.1–9, 2018.

[41] L. Song, L. Minku, and X. Yao, "The potential benefit of relevance vector machine to software effort estimation," Proc. International Conference on Predictive Models in Software Engineering (PROMISE), pp.52–61, 2014.

[42] K. Strike, K. Eman, and N. Madhavji, "Software Cost Estimation with Incomplete Data," IEEE Trans. Softw. Eng., vol.27, no.10, pp.890–908, 2001.

[43] K. Tamura, T. Kakimoto, K. Toda, M. Tsunoda, A. Monden, K. Matsumoto, and N. Ohsugi, "Empirical Evaluation of Similarity-Based Missing Data Imputation for Effort Estimation," Computer Software, vol.26, no.3, pp.3_44-3_55, 2009 (in Japanese).

[44] H. Tan, Y. Zhao, and H. Zhang, "Conceptual data model-based software size estimation for information systems," ACM Transactions on Software Engineering and Methodology (TOSEM), vol.19, no.2, Article 4, p.37, 2009.

[45] A. Tosun, B. Turhan, and A. Bener, "Feature weighting heuristics for analogy-based effort estimation models," Expert Systems with Applications, vol.36, no.7, pp.10325–10333, 2009.

[46] F. Walkerden and R. Jeffery, "An Empirical Study of Analogy-based Software Effort Estimation," Empirical Software Engineering, vol.4, no.2, pp.135–158, 1999.

Appendix:

We can transform Eq. (6) (for overestimation) as follows:

$$
\begin{aligned}
&ov = av(1 + eo/100)\\
&1 + eo/100 = ov/av\\
&eo/100 = ov/av - 1\\
&eo/100 = (ov - av)/av. \qquad \text{(A· 1)}
\end{aligned}
$$

If we denote $eo/100$ as the *BRE*, ov as $\hat{x}$, and av as x, then Eq. (A· 1) is equivalent to Eq. (10) when $x - \hat{x} < 0$.

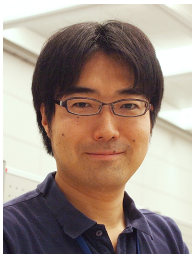
Similarly, we can transform Eq. (7) (for underestimation) as follows:

$$
\begin{aligned}
&ov = av/(1 + eo/100)\\
&ov(1 + eo/100) = av\\
&1 + eo/100 = av/ov\\
&eo/100 = av/ov - 1\\
&eo/100 = (av - ov)/ov. \qquad \text{(A· 2)}
\end{aligned}
$$

If we denote $eo/100$ as the *BRE*, ov as $\hat{x}$, and av as x, then Eq. (A· 2) is equivalent to Eq. (10) when $x - \hat{x} \geq 0$.

Kenichi Ono received the M.E. degree (2017) in information science from Nara Institute of Science and Technology. His research interests include software development effort estimation and mining software repository.

Masateru Tsunoda is an associate professor in the Department of Informatics at Kindai University, Japan. His research interests include software measurement and human factors in software development. Tsunoda received a Doctor of Engineering in information science from the Nara Institute of Science and Technology. He is a member of IEEE, the Institute of Electronics, Information, and Communication Engineers, the Information Processing Society of Japan, JSSST, and the Japan Society for Information and Systems in Education.



Akito Monden is a professor in the Graduate School of Natural Science and Technology at Okayama University, Japan. His research interests include software measurement and analytics, and software security and protection. Monden received a Doctor of Engineering in information science from Nara Institute of Science and Technology. He is a member of IEEE, the Institute of Electronics, Information, and Communication Engineers, the Information Processing Society of Japan, and JSSST.

Kenichi Matsumoto is a professor in the Graduate School of Information Science at Nara Institute of Science and Technology, Japan. His research interests include software measurement and software process. Matsumoto received a PhD in information and computer sciences from Osaka University, Japan. He is a fellow of the Institute of Electronics, Information, and Communication Engineers and the Information Processing Society of Japan, a senior member of IEEE, and a member of ACM and JSSST.