Data Smoothing for Software Effort Estimation

Kento Korenaga
Graduate School of National Science and Technology
Okayama University
Okayama, Japan
pll97u88@s.okayama-u.ac.jp

Akito Monden
Graduate School of National Science and Technology
Okayama University
Okayama, Japan
monden@okayama-u.ac.jp

Zeynep Yücel
Graduate School of National Science and Technology
Okayama University
Okayama, Japan
zeynep@okayama-u.ac.jp

***Abstract*—The goal of this paper is to improve the estimation performance of software development effort by mitigating the problem caused by outliers in a historical software project data set, which is used to construct an effort estimation model. To date, outlier removal methods have been proposed to solve this problem; however, they are not always effective because removing outliers reduces the number of data points (= software projects in our case) in a data set, and a model built from a small data set often suffers from lack of generality. In such a case, estimation performance can become even worse. In this paper we propose a method called data smoothing to mitigate the problem of outliers without reducing the number of data points. We consider that data points are outliers if they do not meet the assumption of Analogy-Based Estimation (ABE) such that "projects with similar features require similar development efforts." The proposed method changes the effort values (person-months or person-hours) in a data set so as to satisfy this assumption; and by this way, all outliers become non-outliers without decreasing the data points. As a result of experimental evaluation using 8 software development data sets, we found that the proposed data smoothing showed the same or higher effort estimation accuracy than the non-smoothing case, while conventional outlier removal method showed worse accuracy in some data set.**

***Index Terms*—Software project planning, outlier removal, data preprocessing**

I. Introduction

In order to make a software development project successful, it is important to estimate the development effort with high accuracy in an early stage of development. For that purpose, methods for constructing an effort estimation model from a historical project data set have been studied [3] [5] [10]. One of a major issues in constructing an effort estimation model is that it is difficult to construct a high-performance estimation model due to the large number of noises and/or outliers included in the data set.

To date, various outlier removal methods have been proposed and evaluated [11] [13] [17] [21]. However, due to the reduction in the number of data points by outlier removal, it was an issue that some data sets were not necessarily expected to improve estimation performance [17] [21]. In this study, we propose a new method called "data smoothing" to mitigate the problem of outliers without reducing the number of data points. Data smoothing is a method of changing (i.e. smoothing) the development effort value of each project in the data set so that the data set satisfies the assumption of Analogy Based Estimation (ABE) [1] such that "projects with similar features require similar development efforts." After data smoothing, all outliers become non-outliers without decreasing the data points.

The organization of the paper is as follows. Section II describes the basic idea and the procedure of the proposed data smoothing method. Section III describes experimental settings including data sets, performance measures and procedure of experiment. Section IV describes the result of experiment followed by discussion. Section V introduces related works. Finally, Section VI summarizes the paper.

II. Proposed Method

A. Basic idea

Data smoothing is a method to smooth the software development effort of all projects in a software project data set so as to satisfy the assumption of ABE. At first, we calculate the similarity for all project pairs in the data set; then, the effort value of each project are changed by the weighted average of similar projects' efforts. Figure 1 explain the idea of data smoothing. Each bar shows a project. The length of bar shows the value of effort and distance between projects shows similarity, i.e. similar projects are close each other in the figure. In this case, project A is considered an outlier because effort value is much larger than that of other similar projects and thus it does not satisfy the ABE assumption. Therefore, we calculate new effort value for project A based on effort values of k-nearest neighbor projects to satisfy the assumption. After calculating new effort value of project A, we apply this process to all other projects as well.

This process is repeated several times to deepen the data smoothing. However, too much smoothing is obviously harmful because efforts of all projects become the same when we maximize the data smoothing. Therefore, we need to experimentally determine the appropriate level of smoothing.

978-1-7281-1651-8/19/$31.00 ©2019 IEEE
SNPD 2019, July 8-11, 2019, Toyama, Japan

After applying data smoothing to the data set, we construct an effort estimation model. In this paper we employ multivariate regression model, which is commonly used in effort estimation studies [10].

B. Procedure of data smoothing

The effort estimation using data smoothing is composed of seven steps as follows:

Step 1. For all pairs of projects included in the data set, the similarity between projects is calculated using explanatory variables of effort such as function point and project duration. In this study, we use Euclidean distance as a similarity measure.

Step 2. A project to be smoothed is selected and the top k-nearest neighbour (k-nn) projects is selected based on the Euclidean distance.

Step 3. Using the effort values of (k + 1) projects including the target project to be smoothed and its k-nn projects, new effort value for the target project is calculated by the weighted average of the effort values. In this paper, k = 3 is used and the weight = 7 is assigned to the project to be smoothed and weight = 1 to each of k-nn projects.

Step 4. Steps 2 and 3 are repeated until all projects in the data set are selected.

Step 5 Replace the effort value of all project to the new effort values calculated in step 3; and now we call 1 round of smoothing is finished.

Step 6 Repeat N rounds of smoothing, that is, go through Step 1 to 5 n times. Here, the first round of smoothing uses the original development efforts recorded in the data set, and for the second and subsequent rounds, we will use the development effort value updated in the previous round. Therefore, it is not necessary to consider the order of projects in which smoothing is applied.

Step 7. Finally, we construct an effort estimation model using the updated efforts.

III. Experimental Evaluation

To evaluate the effectiveness of data smoothing, this paper presents an experiment using 8 industry data sets. In the experiment, we compare data smoothing with a conventional outlier removal method using Cook's distance.

A. Data sets

In this experiment, eight data sets are used: Albrecht [2], China [15], Coc81-dem [4], Desharnais [6], Kemerer [8], Kitchenham [9], Maxwell [14] , and Miyazaki94 [16]. The overview of these data sets are in Table I. All data sets are obtained from an open access repository called Zenodo[1]. We removed some of variables such as project ID and development year because they are unnecessary in effort estimation.

[1]https://zenodo.org/

B. Performance measures

In this experiment we used the following six measures to assess the effort estimation performance:

- **MAE** : Mean Absolute Error
- **MdAE** : Median of Absolute Error
- **MaxAE** : Maximum Absolute Error
- **MMRE** : Mean Magnitude of Relative Error
- **MdMRE** : Median Magnitude of Relative Error
- **MaxMRE** : Maximum Magnitude of Relative Error

C. Procedure of experiment

We applied logarithmic transformation to the data set because it is a common technique to improve effort estimation performance when using linear regression models [10]. The idea behind it is that the logarithmic transformation makes the variable distribution closer to a normal distribution, and this mitigates the problems of data skewness, heteroscedasticity, outliers and heterogeneity [10]. Since data smoothing also can benefit from logarithmic transformation, we applied data smoothing on the log-transformed data set.

Regarding the number of rounds of data smoothing, we tried 0 round (i.e. no smoothing) through 10 rounds to explore the best number of rounds.

We used the leave-one-out cross-validation [12] to evaluate the estimation performance of software development effort after applying (or not applying) data smoothing or outlier removal. All performance measures (such as MAE and MMRE) are calculated after inverse logarithmic transformation of effort values.

IV. Result and Discussion

A. Result of experiment

Figure 2 and 3 show the effort estimation results in terms of MMRE and MdMRE respectively when applying data smoothing for 8 data sets. In these figures, x-axis shows the number of rounds of data smoothing. The number of rounds of 0 indicates that data smoothing is not applied. Y-axis shows the mean and median of the relative error respectively.

In Figure 2, comparing with no-smoothing (i.e. round 0), MMRE of 6 data sets were improved when 1, 2 or 3-round data smoothing were applied. However, when more than 3-round smoothing was applied, MMRE became greater in most data sets.

In Figure 3, MdMRE of 5 data sets were improved when 1-round data smoothing were applied. For Kemerer and Desharnais, MdMRE was greatly improved by 2-round smoothing. When more than 2-round smoothing was applied, MdMRE became greater in most data sets. From these results, we recommend to use 1 or 2-round smoothing. The details of results of 2-round smoothing were shown in Table II, which compares non-smoothing and 2-round smoothing.

Comparison of 2-round data smoothing with the conventional outlier removal method (based on Cook's distance) is shown in Figure 4 and 5 in terms of relative error. In these figures, "original" indicates no data smoothing nor outlier

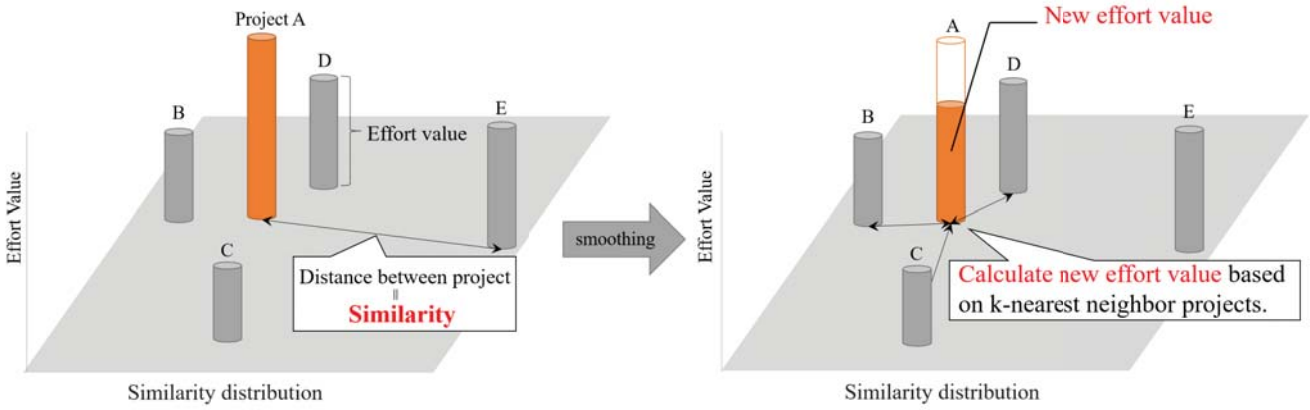

Fig. 1. The overview of data smoothing

TABLE I
OVERVIEW OF PROJECT DATA SETS

Data set	projects	variables	Effort				
			Unit	Min	Med	Max	Avg
Albrecht	24	8	month	0.5	11.45	105.2	21.86
China	499	19	hour	26	1829	54620	3921.05
Coc81-dem	63	23	month	5.9	98	11400	683.32
Desharnais	81	13	hour	546	3647	23940	5046.31
Kemerer	15	8	month	23.2	130.3	1107.31	219.25
Kitchenham	145	10	hour	219	1557	113930	3113.12
Maxwell	62	27	hour	583	5189.5	63694	8223.21
Miyazaki94	48	9	month	5.6	38.1	1586	87.48

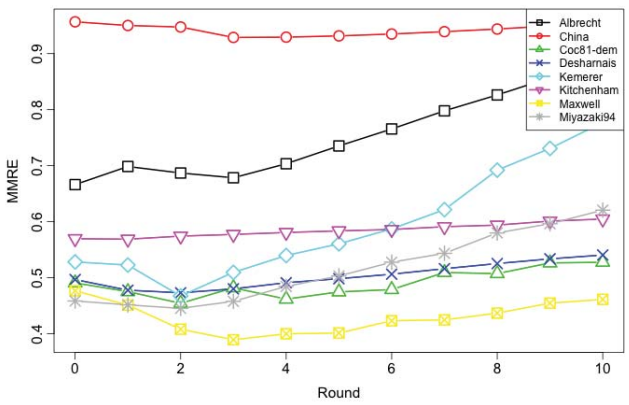

Fig. 2. Result of n-round data smoothing (MMRE)

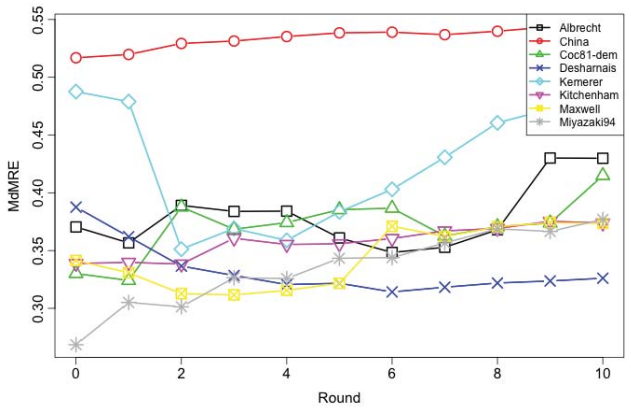

Fig. 3. Result of n-round data smoothing (MdMRE)

removal was applied. From Figure 4 and 5, relative error is obviously improved by data smoothing in 3 data sets Desharnais, Kemerer and Maxwell. On the other hand, there is no obvious improvement by the data removal (Cook).

In this study, Wilcoxon's signed rank test with no pre-processing (original) is performed for each of the proposed method (smoothing) and the comparison method (cook) on the relative error (Table III.) In this study, the significance level is set to 5%, i.e. $p < 0.05$. As shown in Table III, for Desharnais data set, relative error was significantly improved by data smoothing. Other data set did not show the significant difference in data smoothing. On the other hand, relative error became significant worse by data removal (Cook) in Coc81-dem data set. Other data set did not show the significant difference in data removal.

TABLE II
COMPARISON OF NON-SMOOTHING AND 2-ROUND SMOOTHING

Data set	Evaluation index					
	MAE	**MdAE**	**MaxAE**	**MMRE**	**MdMRE**	**MaxMRE**
Albrecht	11.414	4.279	90.984	0.666	0.370	4.389
Albrecht(smoothing)	9.597	4.524	49.370	0.687	0.389	5.153
China	2289.020	885.083	39175.530	0.957	0.517	27.581
China(smoothing)	2332.958	898.476	44690.810	0.948	0.529	25.250
Coc81-dem	346.343	30.486	5917.068	0.491	0.330	2.999
Coc81-dem(smoothing)	267.213	33.171	6319.621	0.454	0.388	2.742
Desharnais	2015.998	1109.638	9645.567	0.497	0.388	2.250
Desharnais(smoothing)	1835.563	1115.685	9093.817	0.473	0.337	2.027
Kemerer	113.907	47.144	850.057	0.528	0.488	1.333
Kemerer(smoothing)	113.831	66.402	828.132	0.468	0.351	1.145
Kitchenham	1466.981	495.635	70678.120	0.570	0.339	9.042
Kitchenham(smoothing)	1530.884	518.202	75361.000	0.574	0.338	7.637
Maxwell	4124.008	1737.251	66150.410	0.477	0.341	1.774
Maxwell(smoothing)	3382.584	1732.081	27423.000	0.408	0.313	2.093
Miyazaki94	48.041	12.922	1247.952	0.459	0.268	2.479
Miyazaki94(smoothing)	45.472	10.872	1275.673	0.445	0.301	3.206

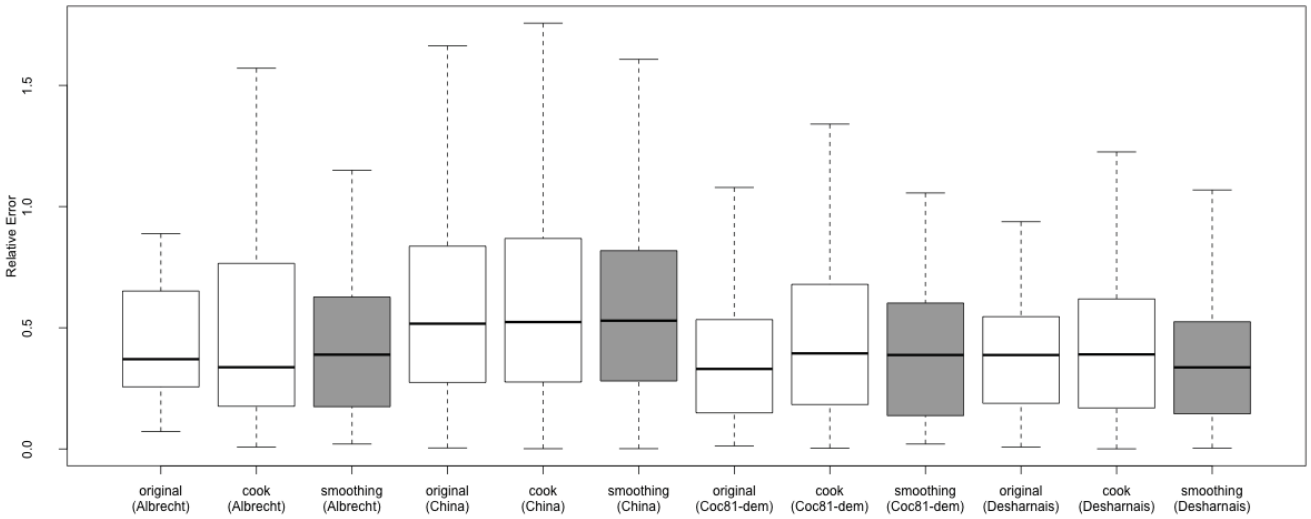

Fig. 4. Boxplots of relative error of Albrecht, China, Coc81-dem and Desharnais data sets

TABLE III
RESULT OF WILCOXON'S SIGNED-RANK TEST

Wilcoxon's signed-rank test	p-value	
	Cook	Smoothing
Albrecht	0.6431	0.8334
China	0.1223	0.0649
Coc81-dem	0.0054	0.5745
Desharnais	0.3437	0.0210
Kemerer	0.7197	0.2524
Kitchenham	0.2122	0.0580
Maxwell	0.7764	0.0744
Miyazaki94	0.8909	0.6219

B. Discussion

Although the result is not strong enough, we confirmed that data smoothing has a potential to improve effort estimation performance for some data set at low risk, while conventional data removal method has a risk to degrade the performance.

However, there are several points where we can improve our data smoothing method. Firstly, we used fixed number of neighbors k=3 with fixed weights in data smoothing, that is, 7 to the target project and 1 to its neighbors. In future, we need to test different k and weights to optimize our method. In addition, weights could be defined based on the similarity of neighbors, e.g. assign smaller weight on the less similar neighbor.

Another avenue to improvement is similarity computation. We used Euclidean distance in this paper. In future, we need to try alternative similarity measures such as cosine similarity and probabilistic-based similarity [18]. Also, data preprocessing and solution adaptation, which are important components in Analogy-Based Estimation (ABE) [19], can be applied in our data smoothing for further improvement.

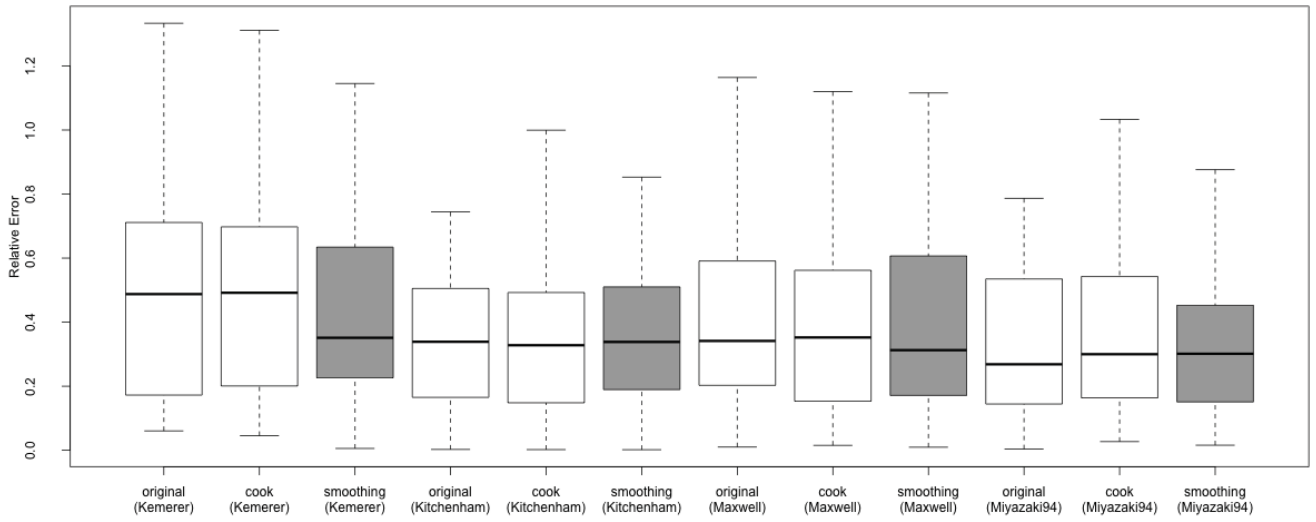

Fig. 5. Boxplots of relative error of Kemerer, Kitchenham, Maxwell and Miyazaki94 data sets

V. Related Work

To date, data smoothing techniques have been proposed to find smooth traces of time series data containing noise [22] [23]. However, these techniques are not suitable for a data set with many variables. High dimension data smoothing is also proposed [7]; however, it also targets time series data, thus it is not suitable for software project data sets. In addition, similarity-based data smoothing is also proposed [20]. However, it targets a sparse co-occurrence data matrix of recommender systems, which contains many zeros, and it is used to replace zero entries with non-zero values. Therefore, their method do not fit our purpose.

Regarding the outlier removal method, this paper employed a commonly-used method based on Cooks distance as a comparison method. On the other hand, there are several outlier removal methods that follows the same basic idea of our proposal (such as TEAK [11] and FISI [13]). These methods measure if a project case should be removed from a data set prior to building an ABE model based on it. The removal criteria are based on whether including a project case would violate the assumption of ABE. It is our important future work to compare our data smoothing method with these outlier removal methods.

VI. Conclusion

This paper proposed a method called data smoothing to mitigate the problem of outliers without reducing the number of data points in a data set. The proposed method follows the assumption of ABE such that "projects with similar features require similar development efforts," and changes the effort values in a data set so as to satisfy this assumption.

An experimental evaluation using 8 software development data sets showed that the proposed data smoothing method showed the same or higher effort estimation accuracy than the non-smoothing case in terms of relative error, while conventional outlier removal method (based on Cook's distance) showed worse accuracy in some data set.

We are aware that our experimental result is not strong enough because the significant improvement was seen in only one data set, and also we compared data smoothing only with Cook's distance-based outlier removal method, which is a very basic method. However, this is the first step to propose and implement the idea of data smoothing, and we believe that the idea itself has a great potential to improve machine learning in many domain because it can mange with outliers without reducing the number of data points. Moreover, as described in Section IV-D, there are many avenues to improve data smoothing in future. In addition, our important future work is to compare data smoothing with other data removal methods such as TEAK [11] and FISI [13].

VII. Acknowledgment

Part of this research was supported by JSPS KAKENHI Grant number 17K00102.

References

[1] M.Azzeh, Y.Elsheikh, M.Alseid, "An Optimized Analogy-Based Project Effort Estimation," International Journal of Advanced Computer Science and Applications, vol.5, no.4, pp.6-11, April 2014.

[2] A.AlbrechtJ.Gaffney, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation," IEEE Transactions on Software Engineering, Vol.9, No.6, pp.83-92, 1979.

[3] B. Baskeles, B. Turhan, and A. Bener, "Software effort estimation using machine learning methods," Proc. 22nd International Symposium on Computer and Information Sciences (ISCIS2007), pp. 126131, 2007.

[4] B. W. Boehm, "Software Engineering Economics," Prentice Hall PTR, 1981.

[5] L. Briand, T. Langley, and I. Wieczorrek, "A replicated assessment and comparison of common software cost modeling techniques," Proc. 22nd International Conference on Software Engineering (ICSE2000), pp. 377386, 2000.

[6] J.M.Desharnais, ”Analyse statistique de la productivitie des projets informatique a partie de la technique des point des fonction,” Masters Thesis, University of Montreal, 1989.
[7] D. Garcia, ”Robust smoothing of gridded data in one and higher dimensions with missing values,” Computational Statistics & Data Analysis, Vol.54, No.4, pp. 11671178, April 2010.
[8] Chris.F.Kemerer, ”An empirical validation of software cost estimation models,” Comm. ACM, vol. 30, no. 5, pp. 416-429, 1987.
[9] B.Kitchenham, ”An empirical study of maintenance and development estimation accuracy,” Journal of Systems and Software, Vol.64, No.1, pp.57-77, 2002.
[10] B. Kitchenham and E. Mendes, ”Why comparative effort prediction studies may be invalid,” Proc. 5th International Conference on Predictor Models in Software Engineering, Article No.4, May 2009.
[11] E. Kocaguneli, T. Menzies, A. Bener, and J. W. Keung, ”Exploiting the essential assumptions of analogy-based effort estimation,” IEEE Trans. Software Eng., vol. 38, no. 2, pp. 425438, 2012.
[12] E. Kocaguneli and T. Menzies, ”Software effort models should be assessed via leave-one-out validation,” Journal of Systems and Software, vol. 86, pp. 1879-1890, 2013.
[13] T. K. Le-Do, K.-A. Yoon, Y.-S. Seo, and D.-H. Bae, ”Filtering of inconsistent software project data for analogy-based effort estimation,” Proc. 34th Annual Computer Software and Applications Conf., pp. 503508, 2010.
[14] K. Maxwell, ”Applied statistics for software managers,” Prentice Hall, 2002.
[15] T. Menzies and M. Rees-Jones and R. Krishna and C. Pape, ”terapromise: one of the largest repositories of se research data,” 2015.
[16] 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.
[17] P. Phannachitta, J. Keung, K. E. Benin, A. Monden, K. Matsumoto, ”Filter-INC: Handling effort-inconsistency in software effort estimation datasets,” Proc. 23rd Asia-Pacific Software Engineering Conference (APSEC2016), pp.185-192, Dec.2016.
[18] P. Phannachitta, J. Keung, A. Monden, K. Matsumoto, ”Improving analogy-based software cost estimation through probabilistic-based similarity measures,” Proc. 20th Asia Pacific Software Engineering Conference (APSEC2013), pp.541-546, December 2013.
[19] P. Phannachitta, J. Keung, A. Monden, and K. Matsumoto, ”A stability assessment of solution adaptation techniques for analogy-based software effort estimation,” Empirical Software Engineering, vol. 22, no. 1, pp.474504, 2017.
[20] A. Popescul, D. M. Pennock, S. Lawrence, ”Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments,” Proc. 17th conference on Uncertainty in artificial intelligence (UAI’01), pp. 437-444, 2001.
[21] 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’08), pp.25-32, May 2008.
[22] P. F. Velleman, ”Definition and comparison of robust nonlinear data smoothing algorithms,” Journal of the American Statistical Association, Vol. 75, No.371, pp. 609-615, 1980.
[23] G. A.Wood and L. S. Jennings, ”On the use of spline functions for data smoothing,” Journal of Biomechanics, Vol. 12, No. 6, pp. 477-479, 1979.