Applying Outlier Deletion to Analogy Based Cost Estimation

Masateru Tsunoda
Nara Institute of Science and Technology
Kansai Science City, 630-0192 Japan
masate-t@is.naist.jp

Akito Monden
Nara Institute of Science and Technology
Kansai Science City, 630-0192 Japan
akito-m@is.naist.jp

Mizuho Watanabe[1]
Nara Institute of Science and Technology
Kansai Science City, 630-0192 Japan
mizuho.watanabe01@is.naist.jp

Takeshi Kakimoto
Kagawa National College of Technology
355 Chokushicho, Takamatsu-shi, Kagawa 761-8058 Japan
kakimoto@t.kagawa-nct.ac.jp

Ken-ichi Matsumoto
Nara Institute of Science and Technology
Kansai Science City, 630-0192 Japan
matumoto@is.naist.jp

ABSTRACT
In this research, we apply outlier deletion methods to analogy based software effort estimation to evaluate their effects. We employed existing deletion methods (Cook's distance based deletion, and Mantel's correlation based deletion) and new deletion method proposed in this research. While existing deletion methods eliminates outliers from entire dataset before estimation, our method identifies and eliminates outliers from neighborhood projects of estimation target. Our method treats a project as an outlier when the effort of the project is extremely higher or lower than other neighborhood projects. In the experiment, our method showed highest performance among applied deletion methods, and average *BRE* (Balanced Relative Error) indicated 20.8% improvement by our method.

Categories and Subject Descriptors
D.2.9 [**Software Engineering**]: Management – *Cost estimation*, K.6.1 [**Computing Milieux**]: Project and People Management – *Staffing*

General Terms
Management, Measurement, Economics, Experimentation.

Keywords
Case based reasoning, effort prediction, abnormal value, project management, productivity.

1. INTRODUCTION
To achieve success of software development project, it is important to estimate development effort accurately, and therefore many quantitative estimation methods have been proposed [2][17][20]. Recently, analogy based estimation [19] gets attention, and many proposals and case studies have been reported [6][7][13][21][22]. Analogy based estimation selects projects (neighborhood projects) which are similar to the estimated project from past project dataset, and estimates effort based on similar projects' effort. One of the advantages of analogy based estimation is that estimation results are comprehensible for estimators such as project managers [22], because they can confirm neighborhood projects used for estimation. Although ordinary estimation models like linear regression model estimate various target projects' effort by one model, analogy based estimation does not make such a model, and estimates effort by neighborhood projects' effort. So analogy based estimation can reflect individuality of each target project in estimation.

Past project dataset sometimes includes project data which should not be used for estimation [18]. For example, projects where exceptional amount of reworks were occurred have larger effort than other same scale projects. Additionally, when effort was inaccurately collected or recorded, recorded effort is different from actual effort. These projects should be eliminated from dataset before estimation, because they lessen estimation accuracy. However, to identify these projects is not easy because inside details of each project are usually not recorded in the dataset. Even if such details can be grasped, it is difficult to settle elimination criteria (For example, to settle criterion of abnormal amount of reworks).

Thus, such projects are often eliminated by statistical outlier deletion methods. Outlier deletion methods identify projects as outliers when specific variants' values are extremely large or combination of variants' values (effort, system size, or duration) is fairly different from other projects' one, and remove them from dataset. Cook's distance is widely used as outlier deletion method when applying linear regression analysis. In addition to Cook's distance, some outlier deletion methods for effort estimation [6][18] have been proposed. However, there are very few case studies which apply outlier deletion methods to analogy based estimation, and evaluate effect of outlier deletion methods toward analogy based estimation.

In this research, we apply outlier deletion methods to analogy based estimation to evaluate their effects. Two types of outlier deletion methods (Cook's distance based deletion, and Mantel's

[1] Presently with IBM Japan, Ltd.

correlation based deletion) are applied to ISBSG dataset [5], and development effort is estimated by analogy based estimation. ISBSG dataset includes many project data which are collected from software development companies, and it is widely used in many researches.

Also, we propose new outlier deletion method considering characteristics of analogy based estimation, and compare its effect to others. In analogy based estimation, when variance of actual effort of neighborhood projects is large, estimation accuracy gets low [16]. Our method identifies a project as an outlier when the effort of the project is extremely higher or lower than other neighborhood projects, and excludes it from computation of estimated effort. Actual effort of neighborhood projects is normalized by Z-score computation [9], and when the normalized value is larger than threshold, the project is identified as an outlier. While existing deletion methods eliminates outliers from entire dataset before estimation, our method does from neighborhood projects.

In what follows, Section 2 explains analogy based estimation. Section 3 explains outlier deletion methods, and Section 4 describes experimental settings. Section 5 shows results of the experiment and discusses it, and Section 6 concludes the paper with a summary.

2. ANALOGY BASED ESTIMATION

The origin of analogy based estimation is CBR (case based reasoning), which is studied in artificial intelligence field. Shepperd et al. [19] applied CBR to software development effort estimation. CBR selects a case similar to current issue from accumulated past cases, and applies solution of the case to the issue. CBR assumes similar issues can be solved by similar solution. Analogy based estimation assumes neighborhood (similar) projects (For example, development size and used programming language is similar) have similar effort, and estimates effort based on neighborhood projects' effort. Although ready-made estimation models such as COCOMO [2] can make estimation without stored software project dataset, analogy based estimation cannot estimate without it. It is a weak point of analogy based estimation, but it can be overcome by using public dataset.

Analogy based estimation uses $m\times n$ matrix shown in Table 1. In the matrix, $Proj_i$ is *i-th* project, $Metric_j$ is *j-th* variable, x_{ij} is a value of $Metric_j$ of $Proj_i$, fp_i is the development size (e.g. function point) of $Proj_i$, and y_i is the actual effort of $Proj_i$. We presume $Proj_a$ is estimated project, and $\hat{y}_a$ is the estimated value of y_a. Procedures of analogy based estimation consist of the three steps described below.

Step 1: Since each variable $Metric_j$ has different range of value, this step makes the ranges [0, 1]. The value x'_{ij}, normalized the value of x_{ij} is calculated by:

$$x'_{ij} = \frac{x_{ij} - \min(Mertic_j)}{\max(Metric_j) - \min(Metric_j)} \quad (1)$$

In the equation, $\max(Metric_j)$ and $\min(Metric_j)$ denote the maximum and minimum value of $Metric_j$ respectively.

Step 2: To find projects which are similar to estimated project $Proj_a$ (i.e. identifying neighborhood projects), distance between $Proj_a$ and other projects $Proj_i$ is calculated. Although various

Table 1. Dataset used by analogy based estimation

	Effort	*Size*	$Metric_1$	$Metric_2$	…	$Metric_j$	…	$Metric_n$
$Proj_1$	y_1	fp_1	x_{11}	x_{12}	…	x_{1j}	…	x_{1n}
$Proj_2$	y_2	fp_2	x_{21}	x_{22}	…	x_{2j}	…	x_{2n}
…	…	…	…	…		…		…
$Proj_i$	y_i	fp_i	x_{i1}	x_{i2}	…	x_{ij}	…	x_{in}
…	…	…	…	…		…		…
$Proj_m$	y_m	fp_m	x_{m1}	x_{m2}	…	x_{mj}	…	x_{mn}

measures (e.g. a measure directly handling nominal variables) are proposed [1], we applied Euclidean distance measure because it is widely used [21]. In the measure, short distance indicates two projects are similar. Distance Dist($Proj_a$, $Proj_i$) between $Proj_a$ and $Proj_i$ is calculated by:

$$\mathrm{Dist}(Proj_a, Proj_i) = \sqrt{\sum_{h=1}^{m} (x'_{ah} - x'_{ihh})^2} \quad (2)$$

Step 3: The estimated effort $\hat{y}_a$ of project $Proj_a$ is calculated by actual effort y_i of k neighborhood projects. While average of neighborhood projects' effort is generally used, we adopt size adjustment method, which showed high estimation accuracy in some researches [7][13][22]. Size adjustment method assumes effort y_i is s times (s is real number greater than 0) larger when development size fp_i is s times larger, and the method adjusts effort y_i based on ratio of estimated project's size fp_a and neighborhood project's size fp_i. Adjusted effort $adjy_i$ is calculated by equation (3), and estimated effort $\hat{y}_a$ is calculated by equation (4). In the equation, *Simprojects* denotes the set of k neighborhood projects which have top similarity with $Proj_a$.

$$adjy_i = y_i \times \frac{fp_a}{fp_i} \quad (3)$$

$$\hat{y}_a = \frac{\sum_{h \in Simprojects} adjy_h}{k} \quad (4)$$

3. OUTLIER DELETION METHOD

Outlier deletion method examines whether a case (project) in dataset is an outlier or not, and eliminates it from dataset when it is identified as an outlier. When software development effort is estimated, Cook's distance based deletion is widely applied before building a linear regression model to eliminate outliers (e.g., [12]). However, there are few researches which analyzed effects of outlier deletion methods [18], or proposed outlier deletion method suitable for analogy based estimation [6].

Seo et al. [18] proposed that LTS (least trimmed squares) based deletion and k-means based deletion are applied before effort estimation, and evaluated their effects by estimating development effort with linear regression model, neural network, and Bayesian network. However, they did not use analogy based estimation with a deletion method. Keung et al. [6] proposed Mantel's correlation based deletion. Although they analyzed which projects were eliminated, they did not compare it with other deletion method, estimating development effort with cross validation. Outlier deletion methods used in our research are explained below.

3.1 Cook's Distance Based Deletion

Cook's distance based deletion is used with multiple linear regression analysis, and identifies an outlier when the case greatly varies coefficient of the regression model. Cook's distance indicates how much residual of all cases varies when a certain case is omitted from model building. Large Cook's distance means the case greatly affects the model. A case is eliminated from dataset when Cook's distance is larger than 4 / n (n is the number of cases in the dataset). Although Cook's distance based deletion is used when linear regression model is built, we applied it to analogy based estimation, because it is widely used in many effort estimation researches.

3.2 Mantel's Correlation Based Deletion

Mantel's correlation based deletion identifies an outlier when a set of independent variables' values is similar, but dependent variable's value is not similar to other cases. The method is originally proposed in Analogy-X method [6] designed for analogy based estimation. Analogy-X method is (1) delivering a statistical basis, (2) detecting a statistically significant relationship and reject non-significant relationships, (3) providing simple mechanism for variable selection, (4) identifying abnormal data point (project) within a dataset, and (5) supporting sensitivity analysis that can detect spurious correlations in a dataset. We applied function (3) as outlier deletion method.

While ordinary correlation coefficient like Pearson's correlation denotes strength of relationship between two variables, Mantel's correlation does between two set of variables (i.e. a set of independent variables and a dependent variable). Mantel's correlation clarifies whether development effort (dependent variable) is similar or not, when project attributes like duration or development size (a set of independent variable) is similar. To settle Mantel's correlation, Euclidean distance based on independent variables and Euclidean distance based on dependent variable is calculated, and then correlation coefficient of them is calculated.

Mantel's correlation based deletion identifies outliers by the following procedure.

1. For all projects, Mantel's correlation r_i is calculated by excluding *i-th* project.
2. Average of r_i ($\bar{r}$), and standard deviation of r_i (rs) are calculated.
3. Leverage metric lm_i, impact of *i-th* project on $\bar{r}$ is calculated by the following equation:

$$lm_i = r_i - \bar{r} \quad (5)$$

4. lm_i is divided by rs, and when the value (standard score) is larger than 4, the project is eliminated from dataset.

3.3 Neighborhood's effort based deletion

Our method, neighborhood's effort based deletion identifies an outlier when effort of a project is extremely higher or lower than other neighborhood projects. As stated in section 2, procedure of analogy based estimation consists of range normalization (step 1), neighborhood projects selection (step 2), and estimated effort computation (step 3). When effort of neighborhood projects is not homogeneous in step 2, estimation accuracy gets low [16]. Focusing on the issue, our method identifies an outlier in step 2. Analogy based estimation assumes when characteristics (independent variables' values) of project is similar, effort (dependent variable's value) is also similar. Our method treats a project as an outlier when the project is not fit to the assumption. While existing deletion methods eliminates outliers from entire dataset before estimation, our method eliminates outliers after selecting neighborhood projects.

To identify an outlier, effort of each neighborhood project is compared to average of neighborhood projects' effort. However, when variance of neighborhood projects' effort is large, each project's deviation from the average effort is also large. So Z-score computation [9] is applied to standardize each neighborhood's effort before the comparison. In more detail, our method eliminates outliers from k neighborhood projects as follows. Note that although neighborhood project's effort is denoted by y_i, y_i signifies size adjusted effort, not actual effort when using size adjustment method.

1. Average of y_i ($\bar{y}$) and standard deviation of y_i (ys) is calculated.
2. Standardized effort y'_i is calculated by the following equation (Z-score):

$$y'_i = \frac{y_i - \bar{y}}{ys} \quad (6)$$

3. A project is identified as an outlier and eliminated when absolute value of y'_i is greater than threshold th (i.e. when deviation from $\bar{y}$ is greater than th times ys). Note that if all neighborhood projects are identified as outliers, no project is eliminated. Estimated effort $\hat{y}_a$ is calculated by the following equation:

$$\hat{y}_a = \frac{\sum_{h \in Eliminatedprojects} y_h}{k - d} \quad (7)$$

In the equation, *Eliminatedprojects* denotes a set of k neighborhood projects excluded d outlier projects. We set th as 1.65 (one sided 5% of standard normal distribution).

Some researches pointed out that when productivity (development size / effort) of neighborhood projects is not homogeneous, estimation accuracy with size adjustment method gets low [7][13]. Our method with size adjustment is regarded to eliminate outliers based on productivity. When size adjustment is used, our method eliminates a project whose adjusted effort $adjy_i$ is extremely higher or lower. From equation (3), $adjy_i$ is calculated by multiplying estimated project's size fp_a by the reciprocal productivity y_i / fp_i (y_i is effort of a neighborhood project, and fp_i is development size of its). fp_a is same for all neighborhood projects, and therefore it is regarded as a constant. Therefore, $adjy_i$ is regarded as productivity in our method.

Our method is totally different from neighborhood selection such as distance-based neighborhood selection. When neighborhoods are selected, only independent variables are used and dependent variable is not used, because the value of dependent variable of the target project is unknown. On the contrary, our method is used independent variable.

4. EXPERIMENT

4.1 Dataset

To evaluate existing deletion methods and our method, we compared estimation accuracy of analogy based estimation when each outlier deletion method is applied. We used ISBSG dataset [5], which is provided by International Software Benchmark Standard Group (ISBSG). It includes project data collected from software development companies in 20 countries, and the projects were carried out between 1989 and 2004.

We assumed estimation point is the end of project plan phase. So, only variables whose values were fixed at the point were used as independent variables, although 99 variables are recorded in the dataset. The independent variables are same as the previous study [11] (unadjusted function point, development type, programming language, and development platform). Development type, programming language, and development platform were transformed into dummy variables, because they are nominal scale variables.

ISBSG dataset includes low quality project data (Data quality ratings are also included in the dataset). So we extracted projects based on the previous study [11] (Data quality rating is A or B, and function point was recorded by IFUPG method, and so on). Also, we excluded projects which included missing values. As a result, we used 593 projects.

4.2 Evaluation criteria

To evaluate accuracy of effort estimation, we used average and median of *AE* (Absolute Error), *MRE* (Magnitude of Relative Error) [4], *MER* (Magnitude of Error Relative to the estimate) [8], and *BRE* (Balanced Relative Error) [14].

When x denotes actual effort, and $\hat{x}$ denotes estimated effort, each criterion is calculated by the following equations:

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

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

$$MER = \frac{|x - \hat{x}|}{\hat{x}} \tag{10}$$

$$BRE = \begin{cases} \dfrac{|x - \hat{x}|}{\hat{x}}, x - \hat{x} \geq 0 \\ \dfrac{|x - \hat{x}|}{\hat{x}}, x - \hat{x} < 0 \end{cases} \tag{11}$$

Lower value of each criterion indicates higher estimation accuracy. Intuitively, *MRE* means relative error to actual effort, and *MER* means relative error to estimated value. However, *MRE* and *MER* have biases for evaluating under and over estimation [3][10]. Maximum *MRE* is 1 even if terrible underestimate is occurred (For instance, when actual effort is 1000 person-hour, and estimated effort is 0 person-hour, *MRE* is 1). Similarly, maximum *MER* is smaller than 1 when overestimate is occurred. So in addition to *MRE* and *MER*, we adopted *BRE* whose evaluation is not biased both *MRE* and *MER* [15]. We did not use Pred(25) [4] which is sometimes used as an evaluation criterion, because Pred(25) is based on *MRE* and it has also bias for evaluating under estimation.

4.3 Experimental Procedure

Experimental procedure for existing deletion methods is follows:

1. Dataset is randomly divided into two equal set. One is treated as fit dataset, and the other is treated as test dataset. Fit dataset is used to compute estimated effort (regarded as past projects), and test dataset is used as estimation target (regarded as ongoing projects).
2. Outlier deletion method is applied to fit dataset, to eliminate outliers from fit dataset.
3. To decide neighborhood size *k*, estimation for fit dataset is performed, changing *k* from 1 to 20. After estimation, residual sum of square (It is same as sum of squares of *AE*, and widely used for estimation model selection [10]) is calculated, and *k* which shows smallest residual sum of square is adopted.
4. Estimation for test dataset is performed. *k* which is settled at step 3 is used.
5. Evaluation criteria are calculated by actual effort of test dataset and estimated effort.
6. Step 1 to 5 is repeated 10 times (As a result, 10 sets of fit dataset, test dataset, and evaluation criteria are made).

Experimental procedure for our method is follows:

1. Dataset is randomly divided into two equal set. One is treated as fit dataset, and the other is treated as test dataset.
2. Estimation for fit dataset is performed, changing *k* from 1 to 20. After estimation, residual sum of square is calculated, and *k* which shows smallest residual sum of square is adopted.
3. Estimation for test dataset is performed with our method. *k* which is settled at step 2 is used.
4. Evaluation criteria are calculated by actual effort of test dataset and estimated effort.
5. Step 1 to 4 is repeated 10 times.

5. RESULTS AND DISCUSSION

Experimental results are shown in Table 2 and Table 3. Table 2 denotes the values of evaluation criteria and deletion ratio when each outlier method is applied. Deletion ratio is defined as the number of deleted projects / the number of all projects. When our method is applied, it is defined as average of the number of deleted projects / neighborhood size. The values of evaluation criteria are average for 10 test dataset. Table 3 denotes differences of evaluation criteria between when each outlier deletion method is applied and not applied. Negative values mean evaluation criteria got worse by applying outlier deletion method. Table 3 also shows statistical test results for the difference by Wilcoxon signed-rank test (The values of evaluation criteria did not have normal distribution). Significant level was set as 5%, and italicized figures in the table signify there were significant differences.

When our method was applied, average *MER* got worse, but other criteria got better (Average *AE*, median *AE*, and median *MRE* showed significant difference). Especially, average *BRE* showed 20.8% improvement and median *BRE* did 5.3%. In case of Man-

Table 2. Evaluation criteria of each outlier deletion method

Outlier deletion method	Average *AE*	Median *AE*	Average *MRE*	Median *MRE*	Average *MER*	Median *MER*	Average *BRE*	Median *BRE*	Deletion ration
Not applied	3680	1607	166.93%	59.59%	91.27%	58.05%	210.36%	91.69%	0.00%
Neighborhood's effort	3252	1374	138.07%	55.98%	97.35%	56.67%	189.56%	86.36%	7.75%
Mantel's correlation	3549	1603	162.40%	60.40%	92.61%	58.48%	207.17%	92.01%	1.21%
Cook's distance	3371	1611	154.69%	62.50%	110.35%	58.80%	216.35%	99.05%	4.65%

Table 3. Difference of evaluation criteria of each outlier deletion method

Outlier deletion method		Average *AE*	Median *AE*	Average *MRE*	Median *MRE*	Average *MER*	Median *MER*	Average *BRE*	Median *BRE*
Neighborhood's effort	Difference	428	234	28.9%	3.6%	-6.1%	1.4%	20.8%	5.3%
	p-value	*0.03*	*0.00*	0.06	*0.03*	0.43	0.70	0.08	0.06
Mantel's correlation	Difference	131	4	4.5%	-0.8%	-1.3%	-0.4%	3.2%	-0.3%
	p-value	0.08	0.58	*0.01*	0.25	0.43	0.74	0.19	0.84
Cook's distance	Difference	309	-4	12.2%	-2.9%	-19.1%	-0.8%	-6.0%	-7.4%
	p-value	*0.01*	0.70	0.06	0.20	0.06	0.85	0.43	0.77

tel's correlation based deletion, four evaluation criteria out of eight were improved, and lowering of the other criteria were smaller than 1.3%. However, the extent of improvement was 3.2% on average *BRE*, and it was very small on median *BRE* (Only average *MRE* showed significant difference). When Cook's distance based deletion was applied, only two evaluation criteria out of eight were improved, and moreover, degradation of average *BRE* and median *BRE* were more than 6%.

We should carefully understand the results because we used only one dataset, but at least, our method shows high performance to eliminate outliers toward a certain kind of dataset (i.e. ISBSG dataset), and we could say that our method is promising. The effect of Mantel's correlation based deletion is not very strong when applied to ISBSG dataset. The result does not means Mantel's correlation based deletion is always not effective to eliminate outliers, but means it is not very effective to a certain kind of dataset. Applying Cook's distance based deletion made estimation accuracy lower. Cook's distance based deletion may be effective to other dataset, but it is not fit to a certain kind of dataset, and therefore we should consider whether Cook's distance based deletion apply or not before using analogy based estimation.

While existing deletion methods identifies outliers from whole dataset, our method does from neighborhood projects of the estimated project. The characteristics of our method may be effective especially when it is applied to dataset like ISBSG dataset which includes various projects.

6. CONCLUSIONS

In this research, we applied outlier deletion methods to analogy based software development effort estimation, and evaluated their effects. Also, we propose new outlier deletion method for analogy based estimation. While existing deletion methods eliminates outliers from entire dataset before estimation, our method does after neighborhood projects are selected by analogy based estimation. In our method, when the effort of the project is extremely higher or lower than other neighborhood projects, it is not used for effort estimation. In the experiment, we estimated development effort using ISBSG dataset, and in the results, our method is most effective, Mantel's correlation based deletion is not very effective, and Cook's distance based deletion made estimation accuracy lower. As future work, we will apply other deletion methods to other dataset and compare their effects to enhance reliability of our research.

7. ACKNOWLEDGMENTS

This work is being conducted as a part of the StagE project, The Development of Next-Generation IT Infrastructure, and Grant-in-aid for Young Scientists (B), 22700034, 2010, supported by the Ministry of Education, Culture, Sports, Science and Technology. We would like to thank Dr. Naoki Ohsugi for offering the collaborative filtering tool.

8. REFERENCES

[1] Angelis, L., and Stamelos, I. 2000. A Simulation Tool for Efficient Analogy Based Cost Estimation, *Empirical Software Engineering*, 5, 1, 35-68.

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

[3] Burgess, C., and Lefley, M. 2001. Can genetic programming improve software effort estimation? A comparative evaluation. *Journal of Information and Software Technology*, 43, 14, 863-873.

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

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

[6] Keung, J., Kitchenham, B., and Jeffery, R. 2008. Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation. *IEEE Trans. on Software Eng*. 34, 4, 471-484.

[7] Kirsopp, C., Mendes, E., Premraj, R., and Shepperd, M. 2003. An Empirical Analysis of Linear Adaptation Techniques for Case-Based Prediction. In *Proc. of International Conference Case-Based Reasoning*, Trondheim, Norway, June 2003, 231-245.

[8] Kitchenham, B., MacDonell, S., Pickard, L., and Shepperd, M. 2001. What Accuracy Statistics Really Measure. In *Proc. of IEE Software*, 148, 3, 81-85.

[9] Larsen, R., and Marx, M. 2000. *An Introduction to Mathematical Statistics and Its Applications*. Prentice Hall.

[10] Lokan, C. 2005. What Should You Optimize When Building an Estimation Model?, In *Proc. of International Software Metrics Symposium (METRICS)*, Como, Italy, Sep. 2005, 34.

[11] Lokan, C., and Mendes, E. 2006. Cross-company and single-company effort models using the ISBSG Database: a further replicated study, In *Proc. of the International Symposium on Empirical Software Engineering (ISESE)*, Rio de Janeiro, Brazil, Sep. 2006, 75-84.

[12] Mendes, E., Martino, S., Ferrucci, F., and Gravino, C. 2008. Cross-company vs. single-company web effort models using the Tukutuku database: An extended study. *The Journal of Systems and Software*, 81, 5, 673-690.

[13] Mendes, E., Mosley, N., and Counsell, S. 2003. A Replicated Assessment of the Use of Adaptation Rules to Improve Web Cost Estimation. In *Proc. of the International Symposium on Empirical Software Engineering (ISESE)*, Rome, Italy, September 2003, 100-109.

[14] Miyazaki, Y., Terakado, M., Ozaki, K., and Nozaki, H. 1994. Robust Regression for Developing Software Estimation Models. *Journal of Systems and Software*, 27, 1, 3-16.

[15] Mølokken-Østvold, K., and Jørgensen, M. 2005. A Comparison of Software Project Overruns-Flexible versus Sequential Development Models, *IEEE Trans. on Software Eng*. 31, 9, 754-766.

[16] Ohsugi, N., Monden, A., Kikuchi, N., Barker, M., Tsunoda, M., Kakimoto, T., and Matsumoto, K. 2007. Is This Cost Estimate Reliable? - the Relationship between Homogeneity of Analogues and Estimation Reliability. In *Proc. of the International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Madrid, Spain, September 2007, 384-392.

[17] Selby, R., and Porter, A. 1988, Learning from examples: generation and evaluation of decision trees for software resource analysis. *IEEE Trans. on Software Eng*. 14, 12, 743-757.

[18] Seo, Y., Yoon, K., and Bae, D. 2008. An Empirical Analysis of Software Effort Estimation with Outlier Elimination. In *proc. of the international workshop on Predictor models in software engineering (PROMISE)*, Leipzig, Germany, May 2008, 25-32.

[19] Shepperd, M., and Schofield, C. 1997. Estimating software project effort using analogies. *IEEE Trans. on Software Eng*. 23, 12, 736-743.

[20] Srinivasan, K., and Fisher, D. 1995. Machine Learning Approaches to Estimating Software Development Effort. *IEEE Trans. on Software Eng*. 21, 2, 126-137.

[21] Tosun, A., Turhan, B., and Bener, A. 2009. Feature weighting heuristics for analogy-based effort estimation models. *Expert Systems with Applications*, 36, 7, 10325-10333.

[22] Walkerden, F., and R. Jeffery. 1999. An Empirical Study of Analogy-based Software Effort Estimation. *Empirical Software Engineering*, 4, 2, 135-158.