Incorporating Expert Judgment into Regression Models of Software Effort Estimation

Masateru Tsunoda
Faculty of Information Sciences and Arts
Toyo University
Saitama, Japan
tsunoda@toyo.jp

Akito Monden
Graduate School of Information Science
Nara Institute of Science and Technology
Nara, Japan
akito-m@is.naist.jp

Jacky Keung
Department of Computing
Hong Kong Polytechnic University
Kowloon, Hong Kong
Jacky.Keung@comp.polyu.edu.hk

Kenichi Matsumoto
Graduate School of Information Science
Nara Institute of Science and Technology
Nara, Japan
matumoto@is.naist.jp

***Abstract*—One of the common problems in building an effort estimation model is that not all the effort factors are suitable as predictor variables. As a supplement of missing information in estimation models, this paper explores the project manager's knowledge about the target project. We assume that the experts can judge the target project's productivity level based on his/her own expert knowledge about the project. We also assume that this judgment can be further improved, because using the expert's judgment solely could incur subjective perception. This paper proposes a regression model building/selection method to address this challenge. In the proposed method, a fit dataset for model building is divided into two or three subsets by project productivity, and an estimation model is built on each data subset. The expert judges the productivity level of the target project and selects one of the models to be used. In the experiment, we used three datasets to evaluate the produced effort estimation models. In the experiment, we adjusted the error rate of the judgment and analyzed the relationship between the error rate and the estimation accuracy. As a result, the judgment-incorporating models produced significantly higher estimation accuracy than the conventional linear regression model, where the expert's error rate is less than 37%.**

Keywords: Software Effort Estimation, Project Management, Expert Judgment, Stratification, Productivity, Estimation error

I. INTRODUCTION

In today's rapid software development environment, software systems grow in size and complexity. Software project management activities such as staffing, scheduling and project progress management are becoming increasingly important to avoid project failure (cost overrun and/or delayed delivery). As a basis of project management, effort estimation plays a fundamental role in the decision making process; therefore, accurate effort estimation is essential to software development project success.

To date, various estimation models that use past projects' historical data have been proposed. One of the most commonly used estimation models is a linear regression model, which represents the relationship between the dependent variable (i.e. effort) and independent variables such as functional size, architecture, programming language, and so on.

A major challenge in building an accurate estimation model is that not all of the effort estimators are available as predictor variables. For example, non-functional requirements greatly affect the ultimate development effort; however, such information is often not available in the historical project datasets. To provide supplement of missing information in estimation models, this paper focuses on the project manager's knowledge and perception about the target project. Generally, experts have more complimentary information than the datasets, such as non-functional requirements, which are not available in quantitative forms in the datasets. In this paper, we assume that the experts are able to determine the target project's productivity level (either high or low, or either high, middle or low) using his/her own knowledge about the project. On the other hand, we also assume that this subjective judgment can be invalid, and this paper also attempts to improve the estimation accuracy using regression models.

To incorporate the expert's judgment on productivity level into the model, this paper proposes the following model building/estimation procedure. First, the dataset of past projects is divided into two or three subsets by productivity, and an estimation model is built on each data subset. Then, the expert judges the productivity level of the estimation target project and chooses one of the models to be used. In this respect, it is not clear how the error rate of expert's judgment affects the final estimation accuracy. So, we experimentally change the error rate of the judgment and analyze the relationship between the error rate and the estimation accuracy.

Section 2 explains our proposed effort estimation method. Section 3 describes the experimental setting, results of the experiment and discussion. Section 4 explains related work, and Section 5 concludes the paper.

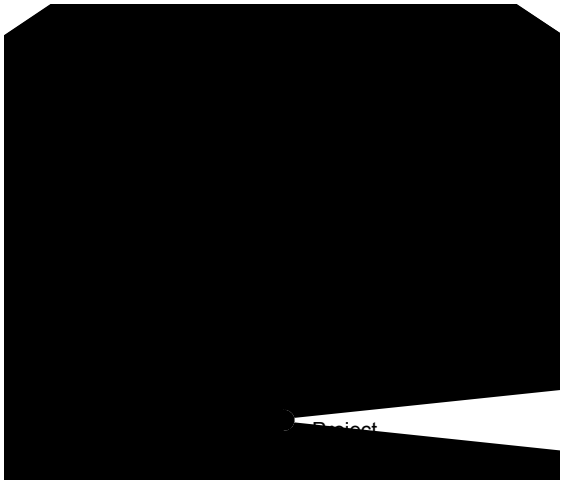


Figure 1. Procedure of the expert judgment incorporated estimation (two-level model)

II. ESTIMATION INCORPORATING EXPERT JUDGMENT

The proposed method uses regression model and the productivity level given by a project manager. Figure 1 illustrates a procedure of our method. Our method consists of four steps. In the first step, a fit dataset (for model building) is divided into subsets by productivity level. In the second step, an estimation model is built on each subset. In the third step, a project manager judges the productivity level of the target project. Lastly, the model is then selected based on the judgment and the selected model estimates the effort. The detail of this procedure is as follows:

1) Dividing dataset: The dataset used to build an estimation model is divided into subsets by productivity. The productivity is defined as the ratio of functional size to effort. We propose two-productivity-level model and three-productivity-level model (i.e. dividing a dataset into two subsets or three subsets). This is because we presume judging the productivity in two or three levels is not difficult for a project manager, but by more than three levels is difficult. When using the two-level model, median of productivity of the dataset is used, and low productivity subset yields high productivity subset are being made. When using the three-level model, first quartile and third quartile of productivity is used, and this resulting low productivity, medium productivity and high productivity subsets are being made. We do not use 33 percentile and 66 percentile because we presume judging particulary low (first quartile) or high (third quartile) productivity project is convinient than judging slightly low (33 percentile) or high (66 percentile) productivity project.

2) Building estimation models: An estimation model is built on the each subset made in step 1. For instance, when the two-level model is applied, two estimation models (an estimation model on the high productivity subset and a model on the low productivity subset) are built. We assume that the estimation model is built using linear regression. When effort is denoted as y, and independent variables such as functional size are denoted as x_1, x_2, and x_3, the effort estimation model based on the linear regression model is denoted as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon \quad (1)$$

In the equation, β_0 is an intercept, β_1, β_2, β_3 are partial regression coefficients, and ε is an error term. Logarithmic transformation is often used when an effort estimation model is built. For example, when logarithmic transformation is applied to x_1, x_2 and y, and not to x_3, the model is denoted as:

$$y = x_1^{\beta 1}\, x_2^{\beta 2}\, e^{\beta_3 x_3 + \beta_0 + \varepsilon} \quad (2)$$

The equation is then transformed as:

$$y = x_2^{\beta 2}\, e^{\beta_3 x_3 + \beta_0}\, x_1^{\beta 1}\, e^{\varepsilon} \quad (3)$$

Where functional size is denoted as x_1, and $x_2^{\beta 2}\, e^{\beta_3 x_3 + \beta_0}$ implies productivity. The model refined by productivity makes the variance of productivity smaller. It helps to infer the coefficients β_2 and β_3, and the intercept β_0 accurately. Additionally, the accurate inference of them helps the inference of the coefficient β_0.

3) Judging productivity: When a project manager judges productivity of a target project, the two-level model is applied, he/she provides the productivity of the project to be high or low level, and compared with past projects. Similarly, when the three-level model is applied, he/she judges the productivity of the project to be high, middle or low. In judgment, the project manager determines the factors seems to affecting productivity the most such as non-functional requirements of the system.

4) Estimating effort: Based on the judgment in step 3, one of the estimation models built in step 2 is selected, and the effort of the target project is estimated. For example, when the two-level model is applied, and the expert selected the high productivity model, the effort is estimated by the selected model.

Instead of dividing the dataset by productivity, the variable which denotes productivity level is applicable, to incorporate the expert's judgment into the estimation model. We applied dividing the dataset because more flexible model is made by the divided dataset than the variable.

III. EXPERIMENT

First, to evaluate the effect of incorporating expert judgment into the estimation model, we experimentally clarify the estimation accuracy of our judgment-incorporating models, comparing to the conventional model, i.e. an effort estimation model based on linear regression without incorporating expert judgment. Then, to evaluate the influence of misjudgment, we change the error rate of expert judgment, and compare the accuracy with the conventional model.

A. Datasets

We used the ISBSG [7], Kitchenham [12], and Desharnais datasets [6]. We assume that the estimation phase is at the end of the project planning. So, only variables whose values were fixed at this point were used as independent variables. Logarithmic transformation was applied to the effort and the

function point in the datasets. Nominal scale variables were transformed into dummy variables (e.g. if the variable has *n* categories, it is transformed into *n* - 1 dummy variables).

The ISBSG dataset is provided by the International Software Benchmark Standard Group (ISBSG), and it includes project data collected from software development companies in 20 countries [7]. The dataset (Release 9) includes 3026 projects that were carried out between 1989 and 2004, and 99 variables were recorded. The ISBSG dataset includes low quality project data (Data quality ratings are also included in the dataset).

We extracted projects based on the previous study [14] (Data quality rating is A or B, function point was recorded by the IFPUG method, and so on). Also, we excluded projects that included missing values (listwise deletion). As a result, we used 593 projects. The variables used in our experiment are FP, language type, development type, and development platform. They are same as the previous study [14].

The Kitchenham dataset includes 145 projects of a single outsourcing company, shown by Kitchenham et al. in their paper [12]. We selected 135 projects that do not include missing values. FP and Development type were chosen as the independent variables, and inadequate variables for effort estimation (e.g. estimated effort by a project manager) were eliminated. Development type was transformed into dummy variables.

The Desharnais dataset includes 88 projects of 1980's, collected from a single Canadian company by Desharnais [6]. The dataset is available at the PROMISE Repository [3]. We used 77 projects that do not have missing values. FP, adjustment factor, experience of team, experience of manager, and language were used as independent variables, and development year and duration were not used. Also, the adjusted function point, the number of transactions, and the number of entities were not used to avoid multicollinearity. Programming language was transformed into dummy variables which reflect different development environments.

The variance of productivity of the ISBSG dataset (collected from multi companies) was the largest, and the variance of the Desharnais dataset (collected from a single company) was the smallest in the datasets.

B. Evaluation criteria

To evaluate the accuracy of effort estimation, we used the conventional metrics such as *MRE* (Magnitude of Relative Error) [5], *MER* (Magnitude of Error Relative to the estimate) [11], and *BRE* (Balanced Relative Error) [15]. Especially, *MRE* is widely used to evaluate the effort estimation accuracy [18] (The residual sum of squares is not widely used for the evaluation). A lower value of each criterion indicates higher estimation accuracy. Intuitively, *MRE* means error relative to actual effort, and *MER* means error relative to estimated effort. However, *MRE* and *MER* have biases for evaluating under and over estimation [4][13]. The maximum *MRE* is 1 even if an extreme underestimate occurs (For instance, when the actual effort is 1000 person-hour, and the estimated effort is 0 person-hour, *MRE* is 1). Similarly, maximum *MER* is smaller than 1 when an overestimate occurs. So we employed *BRE* whose evaluation is not biased as is both *MRE* and *MER* [16], and we evaluated the judgment-incorporating models based on mainly *BRE* (*MRE* and *MER* were used for reference).

C. Procedure of Experiment

In the proposed method, we assume and allow misjudgment occurs, that is, the expert's judgment of productivity is incorrect and thus he/she cannot select the right model. In the experiment, we generate *n*% misjudgment in estimation, and compute the estimation accuracy when the percentage varies. We assume that judging the productivity to be high or low is easy for a project manager when the actual productivity is extremely high or low. On the contrary, when the actual productivity is close to the border line that divides the high and low productivity classes, then misjudgment can be easily occurred.

Figure 2 explains the misjudged projects generated in our experiment. The fit dataset is used to build estimation models (regarded as past projects), and the test dataset is used as the estimation target (regarded as ongoing projects). In the figure, projects in datasets are ordered by productivity, and the number in parentheses indicates productivity of each project. Based on the assumption explained in the previous paragraph, misjudgment of the target project in the test dataset occurs when productivity of the project is close to the borderline of productivity. To generate *n*% misjudgment of target projects, we choose top *n*% projects whose productivity is close to the borderline in the test dataset, and consider they are misjudged.

Figure 3 shows the experimental procedure when the two-level model is used. Details of the procedure are as follows:

1. A dataset is randomly divided into five equal sets.
2. One subset is treated as a test dataset, and the others are treated as a fit dataset (five-fold cross validation).
3. Productivity (the ratio of functional size to effort) of projects included in the fit dataset is computed, and the fit dataset is divided into subsets by the productivity.
4. Estimation model based on linear regression is built on each subset. When building the model, stepwise variable selection based on AIC is applied.
5. Productivity levels of all projects in the test dataset are

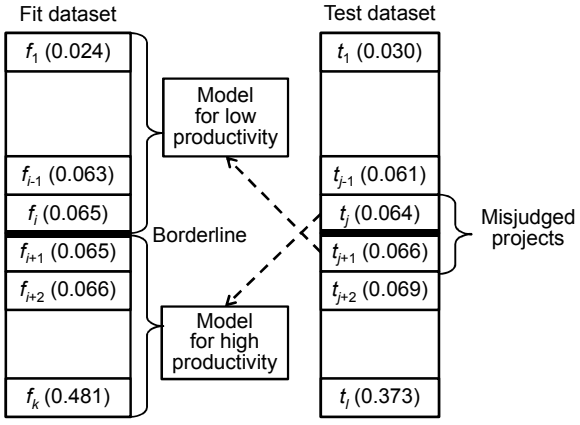


Figure 2. Misjudgment of target projects

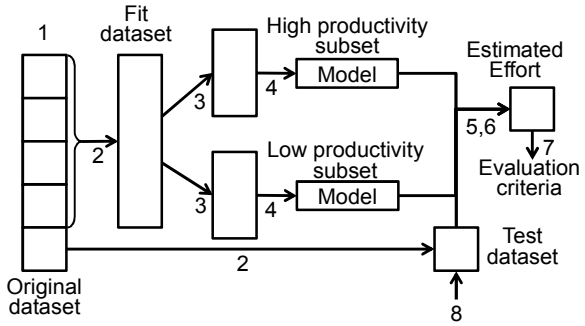


Figure 3. Experimental procedure (two-level model)

settled by their productivity and the misjudgment rate (initially the misjudgment rate is zero.)

6. For each projects in the test dataset, one of the models in step 4 is selected based on the productivity levels settled in step 5, and the effort of the project is estimated.
7. Evaluation criteria are calculated based on the actual effort and the estimated effort.
8. Increase misjudged projects in the test dataset, and Steps 5 to 7 are repeated
9. Steps 2 to 8 are repeated, selecting other subset as fit dataset and test dataset.
10. Steps 1 to 9 are repeated four times, to increase the number of trials.

We also compute the estimation accuracy of the conventional model (without incorporating expert judgment), by skipping step 3, 5 and 8 (The reason for skipping these steps is that the conventional model does not handle productivity, and therefore the judgment and selection steps are not needed). In the experiment, we call it no-level model. To compare the accuracy of our method with the no-level model statistically, we use Wilcoxon signed-rank test at a significance level of 0.05. In the following tables, italic face means the difference was statistically significant at 0.05 level, and the number in parentheses indicates p-value of the test.

D. Results and Discussion

Table 1 shows the accuracy of the no-level model and the proposed models. In the table, no-level indicates the conventional model. There is significantly difference between the no-level model and the proposed models. As shown in the tables, all criteria were greatly improved and there are statistically different. We conclude that if there is no misjudgment, the estimation accuracy can be greatly improved by incorporating expert judgment into the estimation model.

Next, we further explored the influence of misjudgment rate. Table 2 shows the limit rate when all criteria of the judgment-incorporating models were smaller than the no-level model. When the misjudgment rate was smaller than the limit rate, the accuracy of the judgment-incorporating models was higher than the no-level model. Although all evaluation criteria were not statistically different, if we were consider all criteria, when the misjudgment rate was smaller than 36% (average of limit rate on three datasets), performance of the two-level models was higher than the no-level model on average. Similarly, when the misjudgment rate was smaller than 36%, performance of the three-level model was higher the no-level model.

As explained in the evaluation criteria section, we consider that the *BRE* is the most preferable measure in this experiment. So we analyzed the limit rate, by focusing on the *BRE*. The limit rate is shown in Table 3. In the table, "smaller (significant)" means that the average or median *BRE* of the judgment-incorporating models was significantly smaller (higher accuracy) than the no-level model, when misjudgment rate was smaller than the limit rate. "Smaller (not significant)" means average or median *BRE* of the judgment-incorporating models was smaller than the no-level model, when misjudgment rate was smaller than the limit rate, but there was no statistically difference. "Average" is the average of the limit rate on the three datasets.

On average, when the misjudgment rate on the two-level model was smaller than 37%, performance of the model was higher than no-level model because median *BRE* was not lower than the no-level model, and average *BRE* was statistically smaller. Similarly, when the misjudgment rate on the three-level model was smaller than 43%, performance of the model was higher. So, the answer of RQ2 is that when using the two-level model, the acceptable error rate is 37% on average, and it is 43% on average when using the three-level model.

Finally, we compared the two-productivity-level model with the three-level model. Figure 4 shows the change of average *BRE* and median *BRE* on each dataset. When the misjudgment rate was same, three-level model showed higher accuracy than two-level model, except for median *BRE* on Desharnais dataset, when misjudgment rate was high. However, the result does not mean performance of the three-level model is higher than the two-level model, because there is no evidence that the misjudgment rate of the two-level model and the three-level model are same. So, as our future work, we need to clarify actual misjudgment rate of project managers.

IV. RELATED WORK

Apart from the model based effort estimation, expert judgment based effort estimation is proposed, and its practical guidelines are discussed [8]. However, while benefits of the model based estimation and expert judgment based estimation are different, using both approaches is preferable to avoid risks in estimation [9]. Our approach exploits the both benefits of expert judgment and the model based estimation, to enhance the estimation accuracy. Some model based estimation (such as COCOMO [2]) uses variables based on expert judgment such as team skill. However, there is no model that includes highly project-specific but important information such as a serious conflict between the project leader and a best programmer [9]. Our approach can include in the model any kind of information that affects the productivity as far as the project manager is aware of it.

There is a research which uses expert judgment to build a model. Baker [1] proposed an estimation tool which supports experts to build an effort estimation model. However, it uses the conventional estimation model, and therefore it does not directly include the expert's knowledge in the model.

TABLE I. ACCURACY OF THE JUDGMENT-INCORPORATING MODELS

(a) ISBSG dataset

Levels	Average *MRE*	Median *MRE*	Average *MER*	Median *MER*	Average *BRE*	Median *BRE*
No	1.04	0.51	0.84	0.57	1.45	0.86
Two	*0.67* (0.000)	*0.39* (0.000)	*0.57* (0.000)	*0.42* (0.000)	*0.89* (0.000)	*0.54* (0.000)
Three	*0.45* (0.000)	*0.32* (0.000)	*0.42* (0.000)	*0.30* (0.000)	*0.58* (0.000)	*0.38* (0.000)

(b) Kitchenham dataset

Levels	Average *MRE*	Median *MRE*	Average *MER*	Median *MER*	Average *BRE*	Median *BRE*
No	0.69	0.38	0.58	0.41	0.92	0.51
Two	*0.44* (0.000)	*0.26* (0.000)	*0.38* (0.000)	*0.27* (0.000)	*0.55* (0.000)	*0.32* (0.000)
Three	*0.31* (0.000)	*0.20* (0.000)	*0.28* (0.000)	*0.21* (0.000)	*0.38* (0.000)	*0.23* (0.000)

(c) Desharnais dataset

Levels	Average *MRE*	Median *MRE*	Average *MER*	Median *MER*	Average *BRE*	Median *BRE*
No	0.36	0.28	0.38	0.29	0.47	0.34
Two	*0.24* (0.000)	*0.20* (0.000)	*0.26* (0.000)	*0.19* (0.000)	*0.30* (0.000)	*0.21* (0.000)
Three	*0.22* (0.000)	*0.18* (0.000)	*0.23* (0.000)	*0.18* (0.000)	*0.27* (0.000)	*0.20* (0.000)

TABLE II. LIMIT OF THE MISJUDGMENT RATE (BASED ON ALL CRITERIA)

(a) ISBSG dataset

Levels	Misjudgment rate	Average *MRE*	Median *MRE*	Average *MER*	Median *MER*	Average *BRE*	Median *BRE*
Two	33%	*0.78* (0.000)	0.48 (0.231)	*0.74* (0.000)	*0.50* (0.014)	*1.11* (0.000)	*0.67* (0.002)
Three	33%	*0.57* (0.000)	*0.43* (0.001)	0.82 (0.701)	*0.38* (0.000)	*1.02* (0.000)	*0.53* (0.000)

(b) Kitchenham dataset

Levels	Misjudgment rate	Average *MRE*	Median *MRE*	Average *MER*	Median *MER*	Average *BRE*	Median *BRE*
Two	41%	*0.54* (0.000)	0.36 (0.202)	0.57 (0.452)	0.38 (0.165)	*0.77* (0.000)	0.48 (0.729)
Three	41%	*0.48* (0.000)	0.35 (0.064)	*0.52* (0.027)	0.36 (0.123)	*0.68* (0.000)	0.44 (0.105)

(c) Desharnais dataset

Levels	Misjudgment rate	Average *MRE*	Median *MRE*	Average *MER*	Median *MER*	Average *BRE*	Median *BRE*
Two	33%	*0.30* (0.000)	0.27 (0.261)	*0.34* (0.002)	0.26 (0.083)	*0.39* (0.000)	0.31 (0.123)
Three	33%	*0.31* (0.030)	*0.24* (0.014)	*0.33* (0.008)	*0.23* (0.007)	*0.40* (0.000)	*0.27* (0.006)

Similar notion with our assumption that subjective judgment includes error is pointed out by Kitchenham et al. [10]. They pointed out five sources of estimation uncertainty. Assumption error is one of the sources, and indicates error of a model's input parameters. For example, difference of estimated functional size and actual size is the assumption error. The subjective judgment is regarded as the assumption error.

In our method, dividing the dataset by productivity is somewhat akin to the analogy-based estimation (ABE) [17], which estimates effort using similar projects. Whereas ABE uses only available information in the historical project dataset, our method uses not only the available information but also expert judgment. Our method is also applicable to ABE, and it may improve estimation accuracy of ABE, because it was difficult to predict productivity of a target project by ABE in our preliminary analysis.

V. CONCLUSIONS

This work incorporates the expert judgment method to compliment software effort estimation model. We proposed a composite approach of model building, which has four steps. The first step uses a fit dataset divided into two or three subsets by productivity. The second step builds an estimation model on each subset. A project manager judges the productivity level of the target project in the third step. Lastly, the model is selected based on the expert judgment, and the selected model produce better effort estimates, which has been evaluated using an empirical experiment using real project datasets.

In the experiment, three dataset were used and compared the accuracy of the judgment-incorporating models with the conventional model. We changed the error rate of the judgment and analyze the relationship between the rate and the accuracy. The result showed that when the two-productivity-level model is used, the acceptable error rate is 37% on average, and when the three-productivity-level model is used, the rate is 43% on average. We conclude that the judgment-incorporating models are useful for effort estimation given its improved prediction performance. One of important findings of this work is that incorporating the expert judgment is effective to improve the accuracy of the effort estimation model, even if misjudgment occurred to some extent. Future studies will be carried out to clarifying actual misjudgment rate of project managers, and applying the judgment-incorporating models to other estimation method such as the analogy-based estimation. This work is important to the research of combining expert-based software effort estimation in software engineering.

REFERENCES

[1] D. Barker, A Hybrid Approach to Expert and Model Based Effort Estimation, , Master Thesis, West Virginia University, 2007.

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

[3] G. Boetticher, T. Menzies, and T. Ostrand, PROMISE Repository of empirical software engineering data, West Virginia University, Department of Computer Science, 2007.

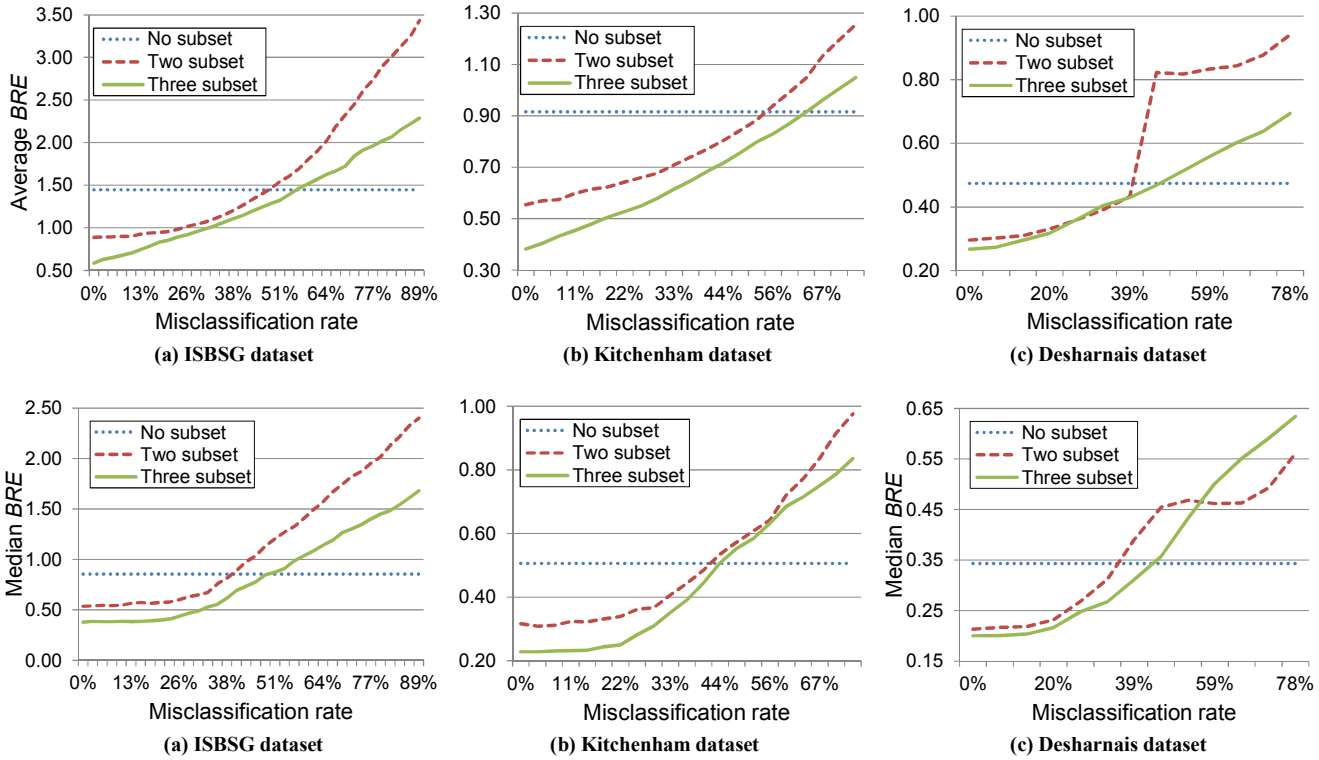
TABLE III. LIMIT OF THE MISJUDGMENT RATE (BASED ON *BRE*)

(a) Two-level model

		ISBSG	Kitchenham	Desharnais	Average
Average *BRE*	**Smaller (Significant)**	41% (0.002)	48% (0.036)	39% (0.006)	43%
	Smaller (Not significant)	46% (0.368)	52% (0.216)	39% (0.006)	46%
Median *BRE*	**Smaller (Significant)**	33% (0.002)	37% (0.021)	26% (0.001)	32%
	Smaller (Not significant)	38% (0.430)	41% (0.729)	33% (0.123)	37%

(b) Three-level model

		ISBSG	Kitchenham	Desharnais	Average
Average *BRE*	**Smaller (Significant)**	48% (0.048)	52% (0.012)	33% (0.000)	44%
	Smaller (Not significant)	54% (0.409)	63% (0.985)	46% (1.000)	54%
Median *BRE*	**Smaller (Significant)**	41% (0.015)	37% (0.001)	33% (0.006)	37%
	Smaller (Not significant)	48% (0.784)	41% (0.105)	39% (0.349)	43%



(a) ISBSG dataset **(b) Kitchenham dataset** **(c) Desharnais dataset**

Figure 4. Transitions of average *BRE* and median *BRE* on each dataset

[4] C. Burgess, and M. Lefley, "Can genetic programming improve software effort estimation? A comparative evaluation," Journal of Information and Software Technology, vol. 43, no. 14, pp. 863–873, 2001.

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

[6] 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.

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

[8] M. Jørgensen, "Practical Guidelines for Expert-Judgment-Based Software Effort Estimation," IEEE Software, vol. 22 no. 3, pp. 57–63, 2005.

[9] M. Jørgensen, B. Boehm, and S. Rifkin, "Software Development Effort Estimation: Formal Models or Expert Judgment?," IEEE Software, vol. 26 no. 2, pp. 14–19, 2009.

[10] B. Kitchenham, and S. Linkman, "Estimates, Uncertainty, and Risk," IEEE Software, vol. 14 no. 3, pp. 69–74, 1997.

[11] B. Kitchenham, S. MacDonell, L. Pickard, and M. Shepperd, "What Accuracy Statistics Really Measure," In Proceedings of IEE Software, vol. 148, no. 3, pp. 81–85, 2001.

[12] B. Kitchenham, S. Pfleeger, B. McColl, and S. Eagan, "An Empirical Study of Maintenance and Develop-ment Estimation Accuracy," Journal of Systems and Software, vol. 64, no. 1, pp. 57–77, 2004.

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

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

[15] 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.

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

[17] M. Shepperd, and C. Schofield, "Estimating software project effort using analogies," IEEE Transaction on Software Engineering, vol. 23, no. 12, pp. 736–743, 1997.

[18] 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.