Bin-Based Estimation of the Amount of Effort for Embedded Software Development Projects with Support Vector Machines

Kazunori Iwata, Elad Liebman, Peter Stone, Toyoshiro Nakashima, Yoshiyuki Anan and Naohiro Ishii

Abstract In this paper we study a bin-based estimation method of the amount of effort associated with code development. We investigate the following 3 variants to define the bins: (1) the same amount of data in a bin (SVM same #), (2) the same range for each bin (SVM same range) and (3) the bins made by Ward's method (SVM Ward). We carry out evaluation experiments to compare the accuracy of the proposed SVM models with that of the ε-SVR using Welch's t-test and effect sizes. These results indicate that the methods SVM same # (1) and SVM Ward (3) can improve the accuracy of estimating the amount of effort in terms of the mean percentage of predictions that fall within 25 % of the actual value.

K. Iwata (✉)
Department of Business Administration, Aichi University,
4-60-6, Hiraike-cho, Nakamura-ku, Nagoya, Aichi 453-8777, Japan
e-mail: kazunori@vega.aichi-u.ac.jp

K. Iwata · E. Liebman · P. Stone
Department of Computer Science, The University of Texas at Austin,
2317 Speedway, Stop D9500, Austin, TX 78712-1757, USA

E. Liebman
e-mail: eladlieb@cs.utexas.edu

P. Stone
e-mail: pstone@cs.utexas.edu

T. Nakashima
Department of Culture-Information Studies, Sugiyama Jogakuen University,
17-3 Moto-machi, Hoshigaoka, Chikusa-ku, Nagoya, Aichi 464-8662, Japan
e-mail: nakasima@sugiyama-u.ac.jp

Y. Anan
Base Division, Omron Software Co., Ltd., Higashiiru, Shiokoji-Horikawa,
Shimogyo-ku, Kyoto 600-8234, Japan
e-mail: yoshiyuki_anan@oss-g.omron.co.jp

N. Ishii
Department of Information Science, Aichi Institute of Technology,
1247 Yachigusa, Yakusa-cho, Toyota, Aichi 470-0392, Japan
e-mail: ishii@aitech.ac.jp

R. Lee (ed.), *Computer and Information Science 2015*,
Studies in Computational Intelligence 614, DOI 10.1007/978-3-319-23467-0_11

1 Introduction

Growth and expansion of the information-based society has resulted in increased use of a wide variety of information products using embedded software systems. The functionality of such products is becoming ever more complex [8, 14], and because of the focus on reliability, guaranteeing product quality is particularly important. Such software represents an important fraction of the budget of businesses and government. It is, therefore, increasingly important for embedded software development companies to realize efficient development methods while guaranteeing delivery time and product quality, and maintaining low development costs [3, 13, 15, 16, 22, 23, 25]. Estimating the amount of effort (man-days cost) requirements for new software projects and guaranteeing product quality are especially important because the amount of effort is directly related to cost, while product quality affects the reputation of the corporation. Considerable attention has been given to various development, management, testing, and reuse techniques, as well as real-time operating systems, tools, and other elements in the embedded software field. However, there has been little research on the relationship between the scale of the development, the amount of effort, and the number of errors using data accumulated from past projects [12, 17, 18]. Thus far, to study the task of effort prediction, the well-known NASA software project data-set has been used [2, 18].

In our formulation of the problem, rather than treat the task of predicting effort as a regression task and predicting a continuous value of effort for code samples, we instead identify blocks of effort, which we refer to as bins, and treat these as labels, which we try to predict, thus treating the problem as a classification task (predicting the correct effort bin for a code sample). In previous work, we investigated the estimation of total effort and errors using artificial neural networks (ANN), and showed that ANN models are superior to regression analysis models for predicting effort and errors in new projects [9, 10]. We also proposed a method to estimate intervals of the number of errors using a support vector machine (SVM) and ANNs [11].

However, these models used a naive method to create bins, which have the same range. In this paper, we propose a novel bin-based estimation method for the amount of effort for embedded software development projects with SVMs, and investigate 3 methods for bin identification. This is crucial to our general framework, since in order to predict an appropriate interval of the amount of effort in a project, it is important to correctly define the intervals (i.e. prediction labels).

In addition, the effectiveness of the SVM (and SVR) using the function depends on selection of the kernel parameter (γ) and soft margin parameter (C) [5]. ε is important for ε-SVR to estimate values effectively. We use three dimensional grid search to select the best combination of them.

We perform extensive evaluations to compare the accuracy of the proposed SVM models with that of the ε-SVR [17] using 10-fold cross-validation as well as by means of Welch's t-test [21, 26] and effect sizes [4, 7]. Our results show that the proposed models can improve the accuracy of estimating the amount of effort in terms of the mean percentage of predictions that fall within 25 % of the actual value.

2 Related Work

2.1 Support Vector Regression

One of the prominent algorithms that has been employed to predict development effort associated with software projects is ε-Support Vector Regression (SVR) [17]. The Support Vector Regression algorithm (SVR) uses the same principles as the canonical Support Vector Machine for classification with a few minor differences [19]. One prominent variant, ε-Support Vector Regression (ε-SVR), uses an ε-insensitive loss function to solve the regression problem and find a closest fitting curve [20].

ε-SVR tries to find a continuous function such that the maximum number of data points lie within the ε-wide insensitivity tube. While previous work did use this approach, it did not probe the optimization of parameters which are crucial to the performance of ε-SVR and similar algorithms, as we do in this paper in Sect. 3.4.

The proposed method to optimize parameters improves the mean magnitude of relative error (*MMRE*: Eq. (3)) from 0.165 [5] to 0.149 by leave-one-out cross-validation (LOOCV) [18]. On the other hand, our proposed SVM models in this paper for the data indicate 0.226 as *MMRE*, because of a small number of data points and independent variables. The number of data points is 18 and that of independent variables is 2.

2.2 Artificial Neural Networks

In earlier papers, we showed that ANN models are superior to regression analysis models for predicting effort and errors in new projects [9]. In addition, we proposed a method for reducing this margin of error [10]. However, methods using ANNs have reached the limit in their improvement, because these methods estimate an appropriate value using what is known as point estimation in statistics. Therefore, we propose in this paper a method for reducing prediction errors using bin-based estimation provided by SVMs. The results of comparison using an ANN are shown in Sect. 4.3. We find out the number of optimal hidden node by 10-fold cross-validation in the comparison. The results demonstrate that the proposed method can estimate the amount of effort better than ANNs.

2.3 Our Contribution

The algorithms proposed in previous work tend to estimate the amount of effort accurately. However, we maintain that this is to some extent an illusion—the NASA software project data set includes the small number of data points, and the dispersion in depended and independent variables is not large. In a more sophisticated approach like the one we propose, a small data set makes it difficult to create appropriate bins: performing regression is easier than bin-based estimation in the case of low dispersion. Our target data sets, however, are large, and manifest a high extent of variability. Specifically, the amount of effort (the dependent variable) is within a certain range, but the values of independent variables are highly variable. In this case, it is difficult for a regression approach to estimate the amount of effort accurately. Therefore, we propose an approach for creating some kind of bins for projects of which the amount of effort is within a certain range to reduce the influence of such dispersion in independent variables.

3 Bin-Based Estimation Models for the Amount of Effort

3.1 Original Data Sets

Using the following data from a large software company, we created bin-based estimation models to estimate the amount of planning effort (*Eff*).

Eff: "The amount of effort", which indicates man-days cost in a review process for software development projects.

V_{new}: "Volume of newly added", which denotes the number of steps in the newly generated functions of the target project.

V_{modify}: "Volume of modification" denoting the number of steps modified or added to existing functions to use the target project.

V_{survey}: "Volume of original project", which denotes the original number of steps in the modified functions, and the number of steps deleted from the functions.

V_{reuse}: "Volume of reuse" denoting the number of steps in functions of which only an external method has been confirmed and which are applied to the target project design without confirming the internal contents.

3.2 Data Selection for Creating Models

To estimate an appropriate binning for the amount of effort in a project, it is important to eliminate outliers. Figures 1 and 2 show the distributions of the amount of effort with bin intervals of 500 and 10, respectively. These distributions confirm that data

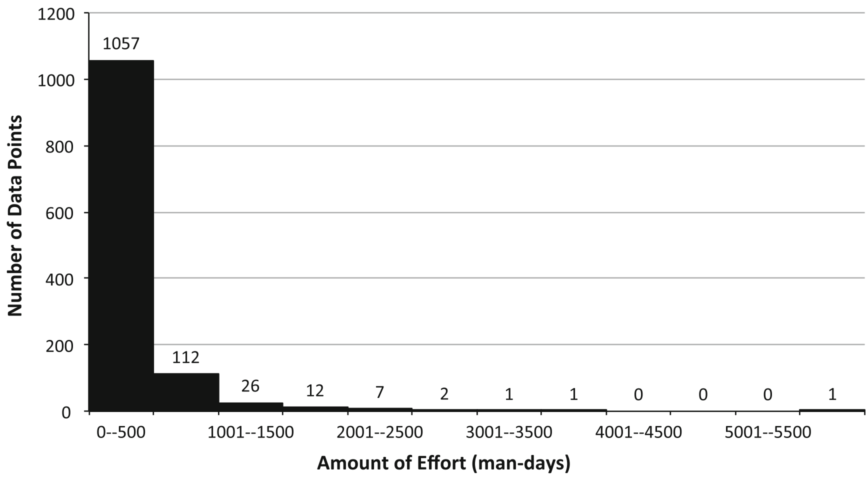

Fig. 1 Distribution of the amount of effort (bins interval 500)

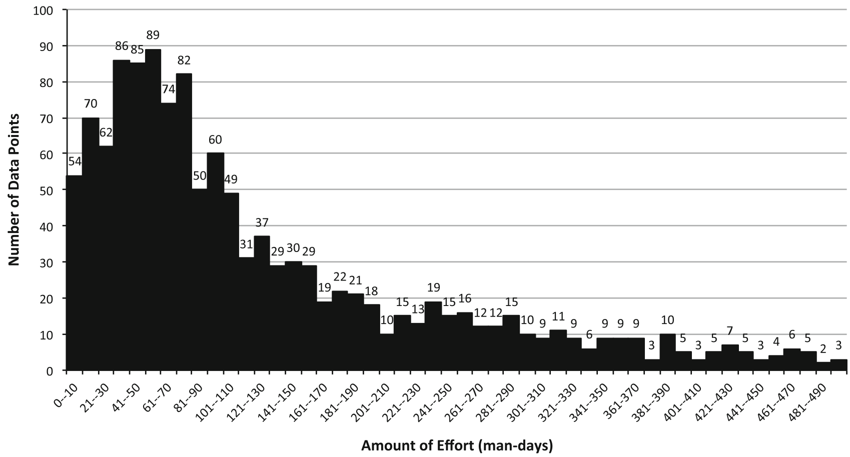

Fig. 2 Distribution of the amount of effort (bins interval 10)

points with less than 500 man-days of effort account for approximately 86.7 % of the total amount of effort. Considering the conditions outlined above, we use the data points which have less than 500 man-days of effort. The distribution of the amount of effort with a bin interval of 10 is shown in Fig. 2. The histogram in this figure has 50 bins and 1057 projects, and our models estimate an appropriate bin for each project.

3.3 General Architecture

SVMs [5, 6] are also supervised learning models. They construct a hyperplane or set of hyperplanes in a high or infinite dimensional space for classification. A good classification can be achieved by the hyperplane with the largest distance to the closest training data point of any class. It often happens, however, that the discrimination sets are not linearly separable in a finite dimensional space. Hence, the SVM maps the original finite dimensional space into a much higher dimensional space in which separation is easier by defining them in terms of a kernel function selected to suit the problem. We use a radial basis function as the kernel function, because this is a popular kernel function for use in SVMs. The corresponding feature space using the function is a Hilbert space of infinite dimensions. Moreover, the effectiveness of the SVM using the function depends on selection of the kernel parameter (γ) and soft margin parameter (C) [5].

The reason why we use SVMs instead of SVRs is that a method to estimate intervals of the number of errors using a support vector machine (SVM) and ANNs showed the better results than these of ANNs for regression and regression analysis [11].

3.3.1 Grouping into Bins for SVM

A representative value of a bin is used as the estimated amount of effort in a project. Therefore, to estimate an appropriate bin of the amount of effort in a project, it is important to define the clusters. We create the following 3 types of bins. A representative value of a cluster is the median of the bin.

- The same amount of data in a bin (SVM same #).
- The same range for each bin (SVM same range).
- The bins made by Ward's method [24] (SVM Ward).

Figure 3 shows the example of same # and same range bins. The target data to be grouped is 15, 20, 30, 40, 50, 70, 80, 90 and 100. The amount of data in each bin is three in the same #. The data belong to the first bin are 15, 20 and 30. The same

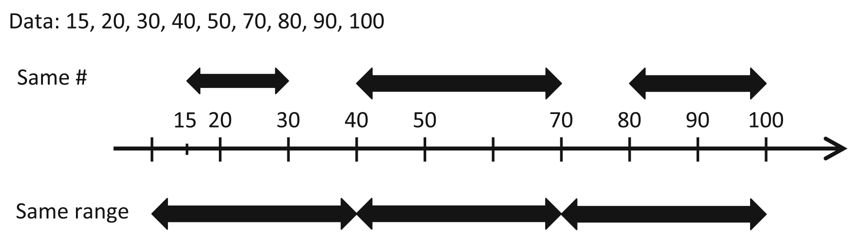

Fig. 3 Example of bins

range adopt 29 as the range. The first bin is [11, 40] and includes 15, 20, 30, 40. If a representative value is the median of each bin, these of the same # are 20, 50 and 90. Correspondingly, these of the same range are 35, 60 and 90.

The accuracy of the estimation depends on the number of bins. Hence, we select the best number of bins with cross-validation and 3D grid-search shown in Sect. 3.4.

3.4 Parameter Selection Using Cross-Validation and 3D Grid-Search

The performance of SVM depends on the choice of the regularization parameters γ and C. The best combination of γ and C is often selected by a grid search with exponentially increasing sequences thereof. In addition, we search for the best number of bins or the most appropriate ε. Hence, we have to define a three-dimensional grid to adapt them using grid-search. The ε and the number of bins are selected with linearly increasing sequences in the three-dimensional grid-search. Figure 4 shows an example of the three-dimensional grid-search. Firstly, the parameters are searched for in the search space $g_1, g_2, \ldots, g_7, g_8$ according to the sparse grid. The cuboid $g'_1, g'_2, \ldots, g'_7, g'_8$ indicating the best combination is found. Next, the cuboid is used as the new search space and partitioned into new grids. Typically, each distinct combination of parameters is checked using cross-validation to avoid over-fitting. We perform 10-fold cross-validation to find the best combination.

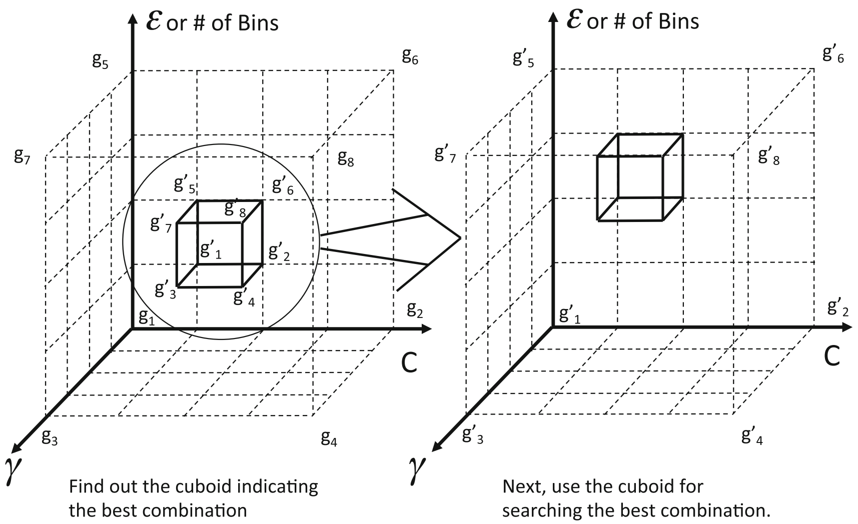

Fig. 4 Example of 3D grid-search

4 Evaluation Experiment

4.1 Evaluation Criteria

The following 6 criteria are used as the performance measures for the effort estimation models [18]. Equations (1) and (3) are, the smaller the value of each evaluation criterion is, the higher is the accuracy. On the other hand, the larger the value of *MPRED*(*25*) is, the higher is the relative accuracy. The value of $\frac{\widehat{X}-X}{X}$ is regarded as 1, if X is equal to 0 in the calculation of *MARE* and *SDRE*. The accuracy value is expressed as X, while the representative value in the estimated bin is expressed as $\widehat{X}$. A representative value is the median of the bin in this paper. Therefore, if a model could estimate appropriate bins for all projects, *MAE* and *MMRE* would not be 0. For example, if the accuracy value is 13 and the estimated bin is $(11, 20]$, $\widehat{X}$ is 15.5 ($(11+20)/2$) and *MAE* and *MMRE* are equal to 2.5 and 0.1613, respectively. The amount of data is expressed as n.

1. Mean of absolute errors (*MAE*).
2. Standard deviation of absolute errors (*SDAE*).
3. Mean magnitude of relative errors (*MMRE*).
4. Standard deviation of relative errors (*SDRE*).
5. *MPRED*(*25*) is the mean percentage of predictions that fall within 25 % of the actual value.
6. *SDPRED*(*25*) is the standard deviation of predictions that fall within 25 % of the actual value.

$$MAE = \frac{1}{n}\sum |\widehat{X} - X| \tag{1}$$

$$SDAE = \sqrt{\frac{1}{n-1}\sum \left(|\widehat{X} - X| - MAE\right)^2} \tag{2}$$

$$MMRE = \frac{1}{n}\sum \left|\frac{\widehat{X} - X}{X}\right| \tag{3}$$

$$SDRE = \sqrt{\frac{1}{n-1}\sum \left(\left|\frac{\widehat{X} - X}{X}\right| - MARE\right)^2} \tag{4}$$

4.2 Data Used in Evaluation Experiment

We performed 10-fold cross validation on data from 1057 real projects in the evaluation experiment. The original data were randomly partitioned into 10 equal sized subsamples (with each subsample having data from 105 or 106 projects). One of the subsamples was used as the validation data for testing the model, while the remaining nine subsamples were used as training data. The cross-validation process was repeated ten times with each of the ten subsamples used exactly once as validation data.

4.3 Results and Discussion

For each model, the experimental results of the 10-fold cross validation are shown in Tables 1, 2 and 3.

We compared the accuracy of the proposed models with that of the ε-SVR using Welch's t-test [26] and effect sizes [4, 7]. A Student's t-test [21] is used to test the null hypothesis that the means of two normally distributed populations are equal. Welch's t-test is used when the variances of the two samples are assumed to be different to test the null hypothesis that the means of two normally distributed populations are equal if the two sample sizes are equal [1]. Given the t-value and degrees of freedom, a p-value can be found using a table of values from the Student's t-distribution. If the p-value is smaller than or equal to the significance level, the null hypothesis is rejected. The null hypothesis in our experiment is interpreted as "there is no difference between the means of the estimation errors (or the mean percentage) for the proposed model and ε-SVR". Effect size measures either the sizes of associations or the sizes of differences. Cohen provided rules of thumb for interpreting these effect sizes,

Table 1 Experimental results (absolute errors) for estimating the amount of effort

	MAE	*SDAE*	95 % Confidence interval
SVM same #	37.546	38.437	[35.226, 39.866]
SVM same range	40.568	41.689	[38.052, 43.084]
SVM ward	38.311	40.384	[35.874, 40.748]
ε-SVR	36.669	39.403	[34.291, 39.047]
ANN model	84.169	60.449	[80.521, 87.817]

Table 2 Experimental results (relative errors) for estimating the amount of effort

	MMRE	*SDRE*	95 % Confidence interval
SVM same #	0.65355	1.0157	[0.59225, 0.71485]
SVM same range	0.74389	1.3956	[0.65966, 0.82812]
SVM ward	0.68157	1.1862	[0.60998, 0.75316]
ε-SVR	0.71025	2.0037	[0.58932, 0.83118]
ANN model	0.96687	0.082109	[0.96191, 0.97183]

Table 3 Experimental results (PRED(25)) for estimating the amount of effort

	MPRED(*25*)	*SDPRED*(*25*)	95 % Confidence interval
SVM same #	0.36558	0.05924	[0.32320, 0.40796]
SVM same range	0.31064	0.03924	[0.28257, 0.33871]
SVM Ward	0.35707	0.04098	[0.32775, 0.38639]
ε-SVR	0.30305	0.04505	[0.27082, 0.33528]
ANN model	0.0038005	0.000024074	[0.0037833, 0.0038177]

suggesting that Cohen's d of |0.1| represents a 'small' effect size, |0.3| represents a 'medium' effect size and |0.5| represents a 'large' effect size.

The results of the t-test and Cohen's d for *MAE*, *MMRE* and *MPRED*(*25*) in estimating the amount of effort are given in Tables 4, 5 and 6. The underlined p-values in the tables indicates statistically significant differences between the type of bin and ε-SVR. In addition, the underlined Cohen's d values in the tables mean the effect size is large.

Tables 1 and 4 indicate that the method of SVM same range cannot improve the accuracy to estimate the amount of effort than that of ε-SVR in *MAE* and the others have the same estimating accuracy as ε-SVR. The Tables 2 and 5 mean that the proposed methods have the same estimating accuracy as ε-SVR in *MMRE*. The results for *MPRED*(*25*) indicate that statistically significant differences between SVM same # and ε-SVR, and SVM Ward and ε-SVR. In addition, SVM same # and SVM ward improve about 6.252 % $(= \sqrt{(0.05924^2 + 0.04505^2)/2} \times 1.188)$ and 5.400 % $(= \sqrt{(0.04098^2 + 0.04505^2)/2} \times 1.254)$ in terms of *MPRED*(*25*), respectively.

Table 4 Results of t-test for *MAE* between each type of bin and ε-SVR

	SVM same #	SVM same range	SVM ward
t-value	0.5180	2.210	0.9462
p-value	0.6045	<u>**0.02723**</u>	0.3422
Cohen's d	0.02253	0.09612	0.04115

Table 5 Results of t-test for *MMRE* between each type of bin and ε-SVR

	SVM same #	SVM same range	SVM ward
t-value	0.8206	0.4479	0.4004
p-value	0.4210	0.6543	0.6889
Cohen's d	0.03569	0.01948	0.01741

Table 6 Results of t-test for $MPRED(25)$ between each type of bin and ε-SVR

	SVM same #	SVM same range	SVM Ward
t-value	3.082	0.4017	2.805
p-value	**0.006835**	0.6927	**0.01178**
Cohen's d	**1.188**	0.1797	**1.254**

It is evident from these results that the methods SVM same # and SVM Ward can improve the accuracy of estimating the amount of effort in terms of the mean percentage of predictions that fall within 25 % of the actual value. However, the methods and SVM same range cannot improve the mean of absolute errors and the mean magnitude of relative errors. The cause of the results is several large errors for estimating in proposed methods. Despite the usefulness of the mean to investigate the accuracy of models, outliers have the biggest effect on the mean.

5 Conclusion

In this paper we have discussed a bin-based estimation method for the amount of effort with SVMs and investigated the following three approaches for defining suitable bins: (1) the same amount of data in a bin (SVM same #), (2) the same range for each bin (SVM same range) and (3) the bins made by Ward's method (SVM Ward). We have carried out evaluation experiments to compare the accuracy of the proposed SVM model with that of the ε-SVR using 10-fold cross-validation as well as by means of Welch's t-test and effect sizes. The results in estimating the amount of effort have indicated statistically significant differences between SVM same # and ε-SVR, and SVM Ward and ε-SVR in terms of $MPRED(25)$. In addition, SVM same # and SVM ward have improved $MPRED(25)$ about 6.252 % and 5.400 %, respectively. These results have exhibited that the methods SVM same # and SVM Ward can improve the accuracy of estimating the amount of effort in terms of the mean percentage of predictions that fall within 25 % of the actual value.

Our future research includes the following:

1. Having implemented a model to estimate the final amount of effort in new projects, we plan to estimate the amount of effort at various stages in the project development process (e.g. halfway).
2. We intend to employ a more complex method to improve the overall prediction accuracy.
3. Since outliers can be detrimental to our model, more refined approaches to outlier detection may be beneficial to our framework.
4. Overall, more data is needed to further support our work.

Acknowledgments A portion of this work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CNS-1330072, CNS-1305287), ONR (21C184-01), AFRL (FA8750-14-1-0070), and AFOSR (FA9550-14-1-0087).

References

1. Aoki, S.: In testing whether the means of two populations are different (in Japanese) (2007). http://aoki2.si.gunma-u.ac.jp/lecture/BF/index.html
2. Bailey, J.W., Basili, V.R.: A meta-model for software development resource expenditures. In: Proceedings of the 5th International Conference on Software Engineering, ICSE'81, pp. 107–116. IEEE Press, Piscataway (1981). http://dl.acm.org/citation.cfm?id=800078.802522
3. Boehm, B.: Software engineering. IEEE Trans. Softw. Eng. **C-25**(12), 1226–1241 (1976)
4. Cohen, J.: Statistical Power Analysis for the Behavioral Sciences, 2nd edn. Routledge, New York (1988). http://www.worldcat.org/isbn/0805802835
5. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
6. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
7. Cumming, G.: The new statistics: why and how. Psychol. Sci. **25**(1), 7–29 (2014)
8. Hirayama, M.: Current state of embedded software (in Japanese). J. Inf. Process. Soc. Jpn. (IPSJ) **45**(7), 677–681 (2004)
9. Iwata, K., Nakashima, T., Anan, Y., Ishii, N.: Error estimation models integrating previous models and using artificial neural networks for embedded software development projects. In: Proceedings of 20th IEEE International Conference on Tools with Artificial Intelligence, pp. 371–378 (2008)
10. Iwata, K., Nakashima, T., Anan, Y., Ishii, N.: Improving accuracy of an artificial neural network model to predict effort and errors in embedded software development projects. In: Lee, R., Ma, J., Bacon, L., Du, W., Petridis M. (eds.) Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. Studies in Computational Intelligence, vol. 295, pp. 11–21. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13265-0_2
11. Iwata, K., Nakashima, T., Anan, Y., Ishii, N.: Estimating interval of the number of errors for embedded software development projects. Int. J. Softw. Innov. (IJSI) **2**(3), 40–50 (2014). doi:10.4018/ijsi.2014070104
12. Kemerer, C.F.: An empirical validation of software cost estimation models. Commun. ACM **30**(5), 416–429 (1987). doi:10.1145/22899.22906
13. Komiyama, T.: Development of foundation for effective and efficient software process improvement (in Japanese). J. Inf. Process. Soc. Jpn. (IPSJ) **44**(4), 341–347 (2003)
14. Nakamoto, Y., Takada, H., Tamaru, K.: Current state and trend in embedded systems (in Japanese). J. Inf. Process. Soc. Jpn. (IPSJ) **38**(10), 871–878 (1997)
15. Nakashima, S.: Introduction to model-checking of embedded software (in Japanese). J. Inf. Process. Soc. Jpn. (IPSJ) **45**(7), 690–693 (2004)
16. Ogasawara, H., Kojima, S.: Process improvement activities that put importance on stay power (in japanese). J. Inf. Process. Soc. Jpn. (IPSJ) **44**(4), 334–340 (2003)
17. Oliveira, A.L.: Estimation of software project effort with support vector regression. Neurocomputing **69**(1315), 1749–1753 (2006). doi:10.1016/j.neucom.2005.12.119. http://www.sciencedirect.com/science/article/pii/S0925231205004492
18. Shin, M., Goel, A.: Empirical data modeling in software engineering using radial basis functions. IEEE Trans. Softw. Eng. **26**(6), 567–576 (2000). doi:10.1109/32.852743
19. Smola, A., Scholköpf, B.: A tutorial on support vector regression. Stat. Comput. **14**(3), 199–222 (2004). doi:10.1023/B:STCO.0000035301.49549.88

20. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Stat. Comput. **14**(3), 199–222 (2004). doi:10.1023/B:STCO.0000035301.49549.88
21. Student: The probable error of a mean. Biometrika **6**(1), 1–25 (1908)
22. Takagi, Y.: A case study of the success factor in large-scale software system development project (in Japanese). J. Inf. Process. Soc. Jpn. (IPSJ) **44**(4), 348–356 (2003)
23. Tamaru, K.: Trends in software development platform for embedded systems (in Japanese). J. Inf. Process. Soc. Jpn. (IPSJ) **45**(7), 699–703 (2004)
24. Ward, J.H.: Hierarchical grouping to optimize an objective function. J. Am. Stat. Assoc. **58**(301), 236–244 (1963). doi:10.1080/01621459.1963.10500845
25. Watanabe, H.: Product line technology for software development (in Japanese). J. Inf. Process. Soc. Jpn. (IPSJ) **45**(7), 694–698 (2004)
26. Welch, B.L.: The generalization of student's problem when several different population variances are involved. Biometrika **34**(28), 28 (1947)