Journal of Software Engineering and Applications, **2013, 6, 1-7**
http://dx.doi.org/10.4236/jsea.2013.64A001 Published Online April 2013 (http://www.scirp.org/journal/jsea)

A Bootstrapping Approach for Software Reliability Measurement Based on a Discretized NHPP Model

Shinji Inoue, Shigeru Yamada

Department of Social Management Engineering, Graduate School of Engineering, Tottori University, Tottori, Japan.
Email: ino@sse.tottori-u.ac.jp, yamada@sse.tottori-u.ac.jp

Received December 14th, 2012; revised January 17th, 2013; accepted January 26th, 2013

ABSTRACT

Discrete software reliability measurement has a proper characteristic for describing a software reliability growth process which depends on a unit of the software fault-detection period, such as the number of test runs, the number of executed test cases. This paper discusses discrete software reliability measurement based on a discretized nonhomogeneous Poisson process (NHPP) model. Especially, we use a bootstrapping method in our discrete software reliability measurement for discussing the statistical inference on parameters and software reliability assessment measures of our model. Finally we show numerical examples of interval estimations based on our bootstrapping method for the several software reliability assessment measures by using actual data.

Keywords: Software Reliability Measurement; Discretized NHPP Model; Nonparametric Bootstrapping Method; Regression Analysis; Bootstrap Confidence Intervals

1. Introduction

It is very important to measure reliability of a software product with accuracy in the final stage of software development process for shipping a highly reliable software product. A software reliability growth model [1-4] is known as one of the useful mathematical tools for quantitative measurement or assessment of software reliability. Generally in an actual testing-phase, we observe a software reliability growth process, in which software faults are detected and removed and the number of faults remaining in the software system is decreasing along with the test-execution time. The software reliability growth model describes the software reliability growth process, and measures the software reliability quantitatively by using software reliability assessment measures, which are derived by the software reliability growth model. A huge number of software reliability growth models were proposed so far for accurate software reliability assessment. Especially, there are discretized nonhomogeneous Poisson process (discretized NHPP) models, which have good fitting and prediction performance in software reliability assessment [5,6] because the discretized NHPP models have consistency with fault counting data, which are obtained by collecting information on the frequency of the software failure-occurrence or the number of detected faults during each constant testing-period. Estimating parameters in the discretized NHPP model from actual data is conducted by using the regression analysis based on a regression equation derived from a difference equation of the discretized NHPP model. After the parameter estimation, software reliability assessment is performed based on the software reliability assessment measures derived from the discretized NHPP model. This approach is based on the point estimation, which is better to use when we have enough number of data.

In recent years, it is very difficult to obtain enough number of data for the point estimation method due to the quick delivery of software development. In such case, it is better to conduct interval estimation for considering the uncertainty of the estimators being related to the model parameters and software reliability assessment measures. We often use asymptotic approximation approaches [7] for the interval estimation. However, we have some difficulty in mathematical manipulation for conducting the interval estimation even if we use the asymptotic approximation approach. For overcoming the problem above, the bootstrap method [8] was proposed. The bootstrapping method is known as one of the useful Monte Carlo methods for obtaining probability distributions for estimators by a resampling method. Recently,

the bootstrapping method is applied not only to software reliability analysis [9-11] but also optimal checkpoint replacement for hardware system [12].

In this paper, we discuss an interval estimation method for parameters and software reliability assessment measures of a discretized exponential software reliability growth model, which is one of the discretized NHPP models and has the simplest model-structure, by the bootstrapping method. And, we discuss several kinds of bootstrap confidence intervals for the interval estimations. Finally, we show numerical examples for our bootstrapping method for software reliability assessment based on the discretized exponential software reliability model and the bootstrap confidence intervals by using actual data.

2. Discretized Exponential NHPP Model

2.1. The Model

We briefly discuss the aspect of the discretized NHPP model by showing a discretized exponential software reliability growth model [5,6], which has the simplest mathematical structure. Now we define a discrete counting process $\{N_n, n=0,1,2,\cdots\}$ representing the cumulative number of faults detected up to *n*-th testing-period. And we can say that the discrete counting process $\{N_n, n=0,1,2,\cdots\}$ follows a discrete-time NHPP [13] if the process has the following property:

$$\begin{aligned}&\Pr\{N_n = x \mid N_0 = 0\}\\ &= \frac{\{\Lambda_n\}^x}{x!}\exp[-\Lambda_n] \quad (n, x = 0,1,2,\cdots),\end{aligned} \tag{1}$$

which is derived based on a continuous-time NHPP. In Equation (1), $\Pr\{A\}$ means the proba of event A. Λ_n is a mean value function of the discrete-time NHPP. The mean value function, Λ_n, represents the expected cumulative number of faults detected up to *n*th testing-period.

The discretized exponential software reliability growth model is a discrete analog of the original (continuous-time) exponential software reliability growth model. Let H_n denote the mean value function following the discretized exponential software reliability growth model. The discretized exponential software reliability growth model is given as

$$H_{n+1} - H_n = \delta\beta(\omega - H_n), \tag{2}$$

from the basic assumptions of the original exponential software reliability growth model. In Equation (2), δ represents the constant time-interval, ω the expected total number of potential faults to be detected in an infinitely long duration or the expected initial fault content, and β the fault detection rate per fault. Regarding the discretization method, we use the Hirota's bilinearization methods [14] for conserving the property of the continuous-time NHPP model. Solving the integrable difference equation in Equation (2), we can obtain an exact solution H_n as

$$H_n = \omega\left[1-(1-\delta\beta)^n\right] \quad (\omega > 0, \beta > 0). \tag{3}$$

As $\delta \to 0$, Equation (3) converges to an exact solution of the original continuous-time exponential software reliability growth model, which is derived by the differential equation.

The discretized exponential software reliability growth model in Equation (3) has two parameters, ω and β, which have to be estimated by using actual data. The parameter estimations of ω and β, $\hat{\omega}$ and $\hat{\beta}$, can be obtained by the following procedure using the method of least-squares. First of all, if we observed fault counting data (n, y_n) $(n=1,2,\cdots,N)$, where y_n represents the cumulative number of faults detected up to *n*th testing-period, we derive the following regression equation from Equation (2):

$$C_n = \alpha_0 + \alpha_1 D_n, \tag{4}$$

where

$$\begin{cases} C_n = H_{n+1} - H_n \equiv y_{n+1} - y_n \\ D_n = H_n \equiv y_n \\ \alpha_0 = \delta\omega\beta \\ \alpha_1 = -\delta\beta. \end{cases} \tag{5}$$

Based on the regression analysis, we can estimate $\hat{\alpha}_0$ and $\hat{\alpha}_1$, which are the estimations of α_0 and α_1 in Equation (4). Then, the parameter estimations, $\hat{\omega}$ and $\hat{\beta}$, can be obtained as

$$\begin{cases} \hat{\omega} = -\hat{\alpha}_0/\hat{\alpha}_1 \\ \hat{\beta} = -\hat{\alpha}_1/\delta. \end{cases} \tag{6}$$

C_n in Equation (4) is independent of δ because δ is not used in calculating C_n as showing Equation (5). Hence, we can obtain the same parameter estimates $\hat{\omega}$ and $\hat{\beta}$, respectively, when we choose any value of δ [5,6,15,16].

2.2. Software Reliability Assessment Measures

Software reliability assessment measures are useful in quantitative software reliability assessment. This paper discusses the expected number of remaining faults and the software reliability function, which are well-known software reliability assessment measures. The expected number of remaining faults, M_n, represents the expected number of undetected faults in the software system at arbitrary testing-period. Then, we have

$$M_n = \mathrm{E}[N_\infty - N_n] = \omega - \Lambda_n = \omega(1-\delta\beta)^n, \tag{7}$$

if we assume that N_n follows a discrete-time NHPP

with mean Λ_n in Equation (3). The software reliability function, $R(n,h)$, is defined as the probability that a software failure does not occur in the time-interval $(n, n+h]$ $(h=1,2,\cdots)$ given that the testing has been going Up to the *n*th testing-priod. Then, we have

$$\begin{aligned} R(n,h) &\equiv \Pr\{N_{n+h}-N_n=0 \mid N_n=x\} \\ &= \exp\left[-\{\Lambda_{n+h}-\Lambda_n\}\right] \\ &= \exp\left[-H_h(1-\delta\beta)^n\right]. \end{aligned} \tag{8}$$

3. Software Reliability Assessment Based on Bootstrapping Method

Ordinarily, the parameters of the discretized NHPP models are estimated by using the regression analysis based on the regression equation derived from the difference equation of the discretized NHPP models. However, it is difficult to discuss the statistical inference on software reliability assessment in the existing estimation approach because it is very difficult or complex to identify the probability distribution function for the estimators of parameter analytically. For overcoming a problems above, Kimura and Fujiwara [9,10] applied non-parametric bootstrap software reliability assessment methods for an incomplete gamma function-based software reliability growth model. Kaneishi and Dohi [11] discussed a parametric bootstrap method for software reliability assessment based on continuous-time NHPP models. In this paper, we apply a non-parametric bootstrap method to the discretized NHPP model for estimating model parameters and for obtaining information for the statistical inference on the parameters and software reliability assessment measures. Especially in this paper, we discuss five types of bootstrap confidence intervals for interval estimation of the model parameters and software reliability assessment measures.

3.1. Our Bootstrapping Method

As an example for discussing our bootstrapping method based on the discretized NHPP model, we apply the discretized exponential software reliability growth model. Our bootstrap method for software reliability assessment follows the following procedure:

Step 1: Estimate α_0 and α_1 in Equation (4) based on the linear regression scheme by using fault counting data. We indicate $\hat{\alpha}_0$ and $\hat{\alpha}_1$ as $\hat{\alpha}_{0(0)}$ and $\hat{\alpha}_{1(0)}$, respectively.

Step 2: Calculate the residual errors, $\hat{d}_i$ at each observation point by

$$\hat{d}_i = C_i - \left(\hat{\alpha}_{0(0)} + \hat{\alpha}_{1(0)} D_i\right) \quad (i=1,2,\cdots,N-1).$$

Step 3: Construct an empirical distribution function $\hat{F}$ by assuming the residual errors $\hat{d}_i$ follows the independent and identically probability distribution and putting mass $1/(N-1)$ at each ordered point

$$\left\{\hat{d}_{[1]}, \hat{d}_{[2]}, \cdots, \hat{d}_{[N-1]}\right\}.$$

Step 4: Set the total number of iteration B and let b $(b=1,2,\cdots,B)$ be the iteration count.

Step 5: Generate a bootstrap sample for the residual errors,

$$\hat{d}^*_{(b)} = \left\{\hat{d}^*_{b,1}, \hat{d}^*_{b,2}, \cdots, \hat{d}^*_{b,N-1}\right\}$$

by sampling with replacement from $\hat{F}$.

Step 6: Generate a bootstrap sample for

$$z^*_{(b)} = \left\{\left(y_1, C^*_{b,1}\right), \left(y_2, C^*_{b,2}\right), \cdots, \left(y_{N-1}, C^*_{b,N-1}\right)\right\}$$

by

$$C^*_{b,i} = \hat{\alpha}_{0(0)} + \hat{\alpha}_{1(0)} D_i + \hat{d}^*_{b,i}.$$

Step 7: Estimate $\alpha^*_{0(b)}$ and $\alpha^*_{1(b)}$ from the bootstrap sample $z^*_{(b)}$.

Step 8: Calculate parameters of the discretized exponential software reliability growth model by the following equation:

$$\omega^*_{(b)} = -\frac{\hat{\alpha}^*_{0(b)}}{\hat{\alpha}^*_{1(b)}}$$

$$\beta^*_{(b)} = -\frac{\hat{\alpha}^*_{1(b)}}{\delta}.$$

Step 9: Calculate software reliability assessment measures.

Step 10: Let $b = b+1$ and go back to Step 5 if. $b < B$

Step 11: We have B samples for $\hat{\omega}$, $\hat{\beta}$ and a software reliability assessment measures.

We can calculate the mean and the standard deviation for the model parameters and software reliability assessment measures by the Monte Carlo approximation, respectively.

3.2. Bootstrap Confidence Intervals

We discuss the following three typical bootstrap confidence intervals [17]: basic, standard normal, and percentile bootstrap confidence intervals. Further we discuss bootstrap-*t* and *BCa* methods [17,18] for deriving bootstrap confidence intervals considering with the asymmetric property and the bias and the skewness of the estimator of the parameter. Let θ be parameter of interest.

The basic bootstrap confidence interval is derived by using the quantile of the distribution of $\hat{\theta}^* - \hat{\theta}$, where $\hat{\theta}^*$ is the bootstrap statistic. We can approximate the α and $(1-\alpha)$ quantile denoting v_a and $v_{a\text{-}1}$, respec- tively, of the distribution of $\hat{\theta}-\theta$ by $\hat{\theta}^*_{[B\alpha]} - \hat{\theta}$ and $\hat{\theta}^*_{[B(1-\alpha)]} - \hat{\theta}$. Then,

$$\begin{aligned}1-2\alpha &= \Pr\left\{v_\alpha \le \hat{\theta}-\theta \le v_{1-\alpha}\right\}\\ &= \Pr\left\{\hat{\theta}-v_{1-\alpha} \le \theta \le \hat{\theta}-v_\alpha\right\}\\ &= \Pr\left\{2\hat{\theta}-\hat{\theta}^*_{[B(1-\alpha)]} \le \theta \le 2\hat{\theta}-\hat{\theta}^*_{[B\alpha]}\right\}.\end{aligned}$$

Thus, the $100(1-2\alpha)\%$ basic bootstrap confidence interval is given by

$$\left[2\hat{\theta}-\hat{\theta}^*_{[B(1-\alpha)]}, 2\hat{\theta}-\hat{\theta}^*_{[B\alpha]}\right]. \quad (9)$$

The standard normal bootstrap confidence interval is derived by assuming that the distribution of $\hat{\theta}-\theta$ can be approximated by the distribution of $\hat{\theta}^*-\hat{\theta}$ and $\hat{\theta}^*-\hat{\theta} \sim N\left(0, SD\left[\hat{\theta}\right]^2\right)$. That is,

$$1-2\alpha = \Pr\left\{w_\alpha \le \frac{\hat{\theta}^*-\hat{\theta}}{SD\left[\hat{\theta}\right]} \le w_{1-\alpha}\right\}.$$

Thus, we have the $100(1-2\alpha)\%$ standard normal bootstrap confidence interval as

$$\left[\hat{\theta}-w_{1-\alpha}SD\left[\hat{\theta}\right], \hat{\theta}-w_\alpha SD\left[\hat{\theta}\right]\right], \quad (10)$$

where w_α is $\Phi^{-1}(\alpha)$, which is the α quantile of the standard normal distribution. For example,

$$w_{1-0.025} = -w_{0.025} = 1.96.$$

The percentile bootstrap confidence interval is calculated from the empirical cumulative probability distribution function, which consists of the bootstrap iteration values: $\hat{\theta}^*_{(1)}, \hat{\theta}^*_{(2)}, \cdots, \hat{\theta}^*_{(B)}$. Then, the $100(1-2\alpha)\%$ percentile bootstrap confidence interval is calculated by

$$\left[\hat{\theta}^*_{[B\alpha]}, \hat{\theta}^*_{[B(1-\alpha)]}\right], \quad (11)$$

where $\hat{\theta}^*_{[B\alpha]}$ represents the α quantile of the empirical cumulative probability distribution function.

Further, we discuss a bootstrap-*t* method, which enables us to take into consideration the variance of $\hat{\theta}$ by deriving $T = \left(\hat{\theta}-\theta\right)/\hat{\sigma}$, where $\hat{\sigma}^2$ is the variance of $\hat{\theta}$. Letting u_α and $u_{(1-\alpha)}$ are the α and $(1-\alpha)$ quantile of T, we have

$$\begin{aligned}1-2\alpha &= \Pr\left\{u_\alpha \le \frac{\hat{\theta}-\theta}{\hat{\sigma}} \le u_{1-\alpha}\right\}\\ &= \Pr\left\{\hat{\theta}-\hat{\sigma}u_{1-\alpha} \le \theta \le \hat{\theta}-\hat{\sigma}u_\alpha\right\}.\end{aligned}$$

In above equation, we substitute u_α and $u_{(1-\alpha)}$, which are α and $(1-\alpha)$ quantile of T, into $T^*_{[B\alpha]}$ and $T^*_{[B(1-\alpha)]}$, which are the α and $(1-\alpha)$ quantile of $T^* = \left(\hat{\theta}^*-\hat{\theta}\right)/\hat{\sigma}^*$. Then, the $100(1-2\alpha)\%$ bootstrap-*t* confidence interval is derived as

$$\left[\hat{\theta}-\hat{\sigma}T^*_{[B(1-\alpha)]}, \hat{\theta}-\hat{\sigma}T^*_{[B\alpha]}\right]. \quad (12)$$

In Equation (12), $\hat{\sigma}$ is the standard deviation of $\hat{\theta}^*_{(b)}$ and $\hat{\sigma}^*$ is estimated by the bootstrap-*t* statistics T^*.

And we also discuss a *BCa* method for getting better bootstrap confidence interval with the asymmetric property, the bias, and the skewness of the probability distribution of the estimator. The *BCa* confidence interval can be given as

$$\left[\hat{G}^{-1}\left\{\Phi\left(z(\alpha)\right)\right\}, \hat{G}^{-1}\left\{\Phi\left(z(1-\alpha)\right)\right\}\right], \quad (13)$$

where

$$z(\alpha) = z_0 + \frac{z_0+z_\alpha}{1-a\left(z_0+z_\alpha\right)}.$$

In above equation, $z_0 = \Phi^{-1}\left\{\hat{G}\left(\hat{\theta}\right)\right\}$, in which $\hat{G}(\cdot)$ is the bootstrap distribution for the estimator. And a is the acceleration constant derived as

$$a = \frac{1}{6}\frac{\sum_{i=1}^{n}\left(\hat{\theta}_{(\cdot)}-\hat{\theta}_{(i)}\right)^3}{\left\{\sum_{i=1}^{n}\left(\hat{\theta}_{(\cdot)}-\hat{\theta}_{(i)}\right)^2\right\}^{3/2}},$$

where $\hat{\theta}_{(i)}$ is a jackknife iteration value, which is estimated by using the data, removed *i*th data and $\hat{\theta}_{(\cdot)} = \sum_{i=1}^{n}\hat{\theta}_{(i)}/n$.

4. Numerial Examples

We show numerical examples for our bootstrap software reliability assessment method based on the discretized exponential software reliability growth model.

We apply fault counting data: (n, y_n) $(n = 1, 2, \cdots, 25;\ y_{25} = 136)$ [1] and we set the total number of iteration $B = 2000$.

We first obtain

$$\hat{\alpha}_{0(0)} = 139.9564 \quad \text{and} \quad \hat{\alpha}_{1(0)} = 0.1133109$$

by the linear regression scheme from the actual data. Following to our bootstrapping method, we have 2000 bootstrap samples $\left\{z^*_{(1)}, z^*_{(2)}, \cdots, z^*_{(2000)}\right\}$. Then, we obtain bootstrap samples for ω and β. **Figures 1** and **2** show histograms for the bootstrap samples $\hat{\omega}^*_{(b)}$ and $\hat{\beta}^*_{(b)}$ to see bootstrap distributions of $\hat{\omega}$ and $\hat{\beta}$, respectively. And we have bootstrap samples for the software reliability assessment measures, such as the expected number of remaining fault at the termination time of the testing, $\hat{M}_{25}$, and the software reliability, $\hat{R}(25,1)$, respectively. These bootstrap samples of $\hat{M}^*_{25(b)}$ and $\hat{R}^*_{(b)}(25,1)$ are calculated by

$$\hat{M}^*_{25(b)} = \frac{\hat{\alpha}^*_{0(b)}}{\hat{\alpha}^*_{1(b)}}\left(1+\hat{\alpha}^*_{1(b)}\right)^{25}, \quad (14)$$

$$\hat{R}^*_{(b)}(25,1) = \exp\left[\frac{\hat{\alpha}^*_{0(b)}}{\hat{\alpha}^*_{1(b)}}\left\{1-\left(1+\hat{\alpha}^*_{1(b)}\right)^1\right\}\left(1+\hat{\alpha}^*_{1(b)}\right)^n\right], \quad (15)$$

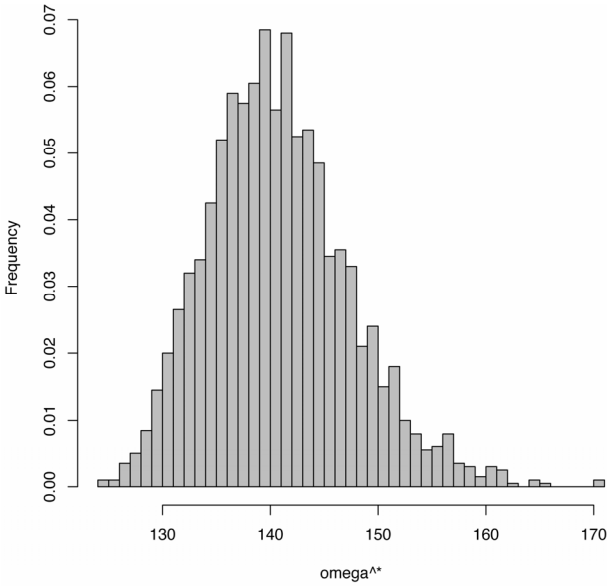

Figure 1. Bootstrap distribution of $\hat{\omega}$.

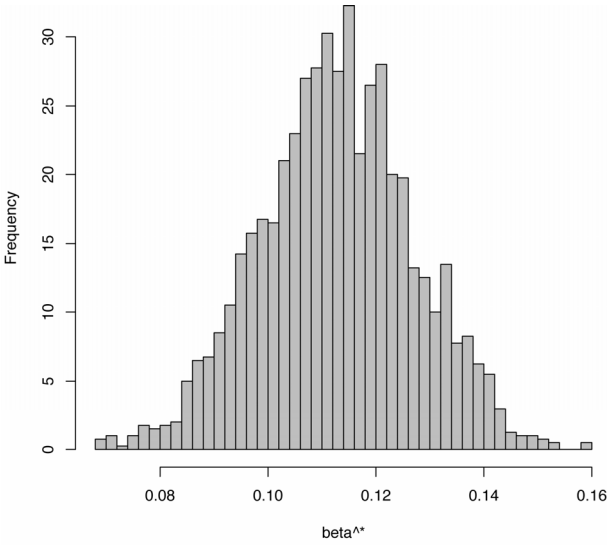

Figure 2. Bootstrap distribution of $\hat{\beta}$.

From Equations (7) and (8), respectively. **Figures 3** and **4** show histograms of the bootstrap samples for $\hat{M}^*_{25(b)}$ and $\hat{R}^*_{(b)}(25,1)$, respectively. Further, **Table 1** indicates the mean and the standard deviations of the estimators of $\hat{\alpha}_0$, $\hat{\alpha}_1$, $\hat{\omega}$, $\hat{\beta}$, $\hat{M}_{25}$, and $\hat{R}(25,1)$, respectively. And, **Figure 5** shows the estimated discretized exponential software reliability growth model, $\hat{H}_n$, in which we use the means of the bootstrap samples of $\hat{\omega}^*_{(b)}$ and $\hat{\beta}^*_{(b)}$ as the point estimations, respectively. The means of $\hat{\omega}^*_{(b)}$ and $\hat{\beta}^*_{(b)}$, which are denoted by $\bar{\omega}$ and $\bar{\beta}$, are calculated by

$$\bar{\omega} = \frac{1}{B}\sum_{b=1}^{B}\hat{\omega}^*_{(b)}, \tag{16}$$

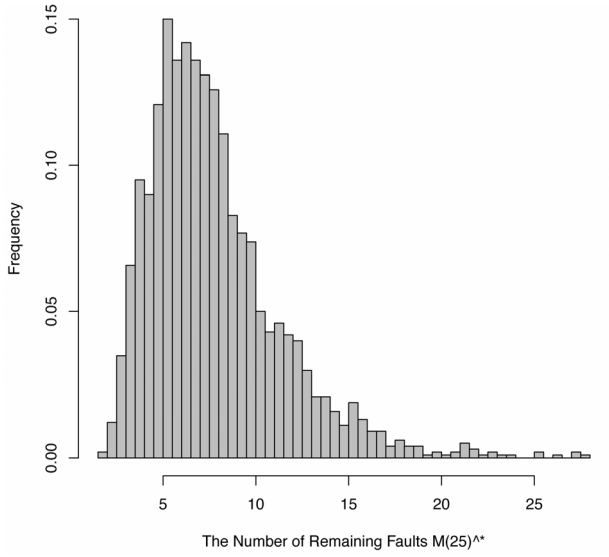

Figure 3. Bootstrap distribution of the expected number of remaining faults at $n = 25$, $\hat{M}_{25}$.

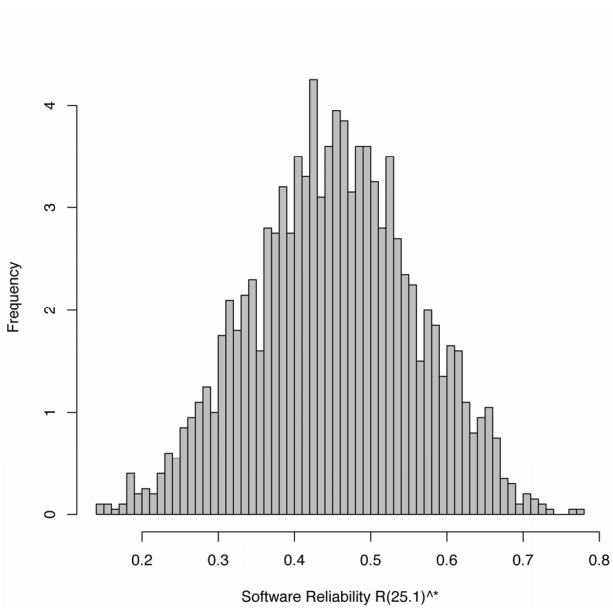

Figure 4. Bootstrap distribution of software reliability, $\hat{R}(25,1)$.

$$\bar{\beta} = \frac{1}{B}\sum_{b=1}^{B}\hat{\beta}^*_{(b)}, \tag{17}$$

respectively.

Further, **Table 2** shows the results of interval estimations based on the basic, standard normal, percentile, bootstrap-*t*, and *BCa* methods, respectively, with the 5% significance level $(\alpha = 0.025)$. From **Table 2**, we can see that the inappropriate confidence intervals for $\hat{M}_{25}$ are estimated in the basic, standard normal, and bootstrap-*t* confidence intervals because the number of re-

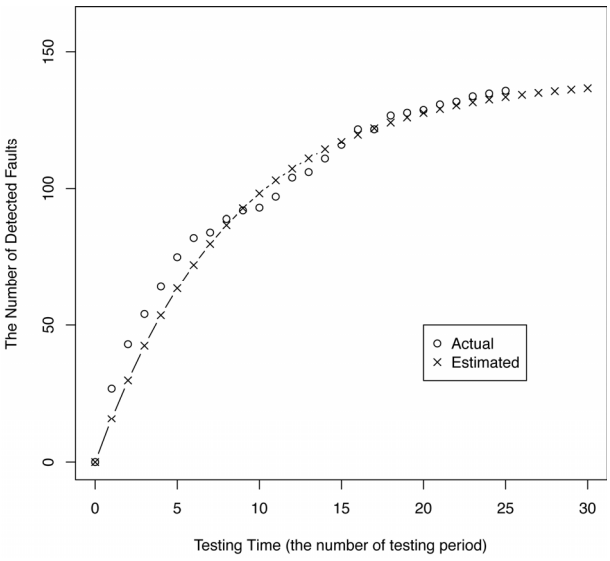

Figure 5. Estimated discretized exponential software reliability growth model, $\hat{H}_n$.

Table 1. Quantities of the bootstrap distribution.

	Mean	Standard Deviation
$\hat{\alpha}_0$	15.79443	1.500837
$\hat{\alpha}_1$	−0.1127263	0.01422163
$\hat{\omega}$	140.7685	6.609685
$\hat{\beta}$	0.1127263	0.01422163
$\hat{M}_{25}$	7.761994	3.59502
$\hat{R}(25,1)$	0.4514124	0.1072053

maining faults does not never take a negative value. And depending on the type of the bootstrap confidence interval, the results of interval estimations on $\hat{M}_{25}$ are notably different each other. These results are caused by assuming the symmetric distributions to derive these bootstrap confidence intervals. However, the probability distribution of an estimator follows an asymmetric distribution and the approximate accuracy is influenced by the bias and the skewness of the probability distribution for the estimator generally. As we show in **Figure 3**, we can say the bootstrap distribution for $\hat{M}_{25}$ follows an asymmetric distribution. On the other hand, the percentile bootstrap confidence interval give us an appropriate interval estimations on $\hat{M}_{25}$ because the interval estimation based on the percentile method is estimated based on only the bootstrap distribution, not assumed a symmetric distribution. Of course, the interval estimations based on the *BCa* method can be thought that we have more appropriate interval estimation on $\hat{M}_{25}$ because the *BCa* method is the improved estimation method for the percentile bootstrap confidence interval.

Table 2. Results of interval estimations based on bootstrap confidence intervals.

		Lower	Upper
$\hat{\omega}$	Basic	124.0544	150.3849
	Standard Normal	127.0015	152.9114
	Percentile	129.528	155.8585
	Bootstrap-*t*	122.9817	152.0545
	BCa	128.1697	153.2575
$\hat{\beta}$	Basic	0.08618762	0.141511
	Standard Normal	0.08543655	0.1411854
	Percentile	0.08511091	0.1404343
	Bootstrap-*t*	0.08372191	0.144191
	BCa	0.09073159	0.1481078
$\hat{M}_{25}$	Basic	−2.835469	10.84297
	Standard Normal	−0.1235227	13.96896
	Percentile	3.002461	16.6809
	Bootstrap-*t*	−5.021783	11.90131
	BCa	2.300986	13.80817
$\hat{R}(25,1)$	Basic	0.2567593	0.6711605
	Standard Normal	0.2462621	0.6665069
	Percentile	0.2416085	0.6560098
	Bootstrap-*t*	0.243411	0.6866959
	BCa	0.2794992	0.7010343

5. Conclusions

This paper discussed a bootstrap software reliability assessment method based on a discretized exponential software reliability growth model. And we discussed five types of bootstrapping confidence intervals for interval estimations of model parameters and several software reliability assessment measures.

In our numerical examples, we confirmed that our bootstrap approach gives probability distributions of each parameters and software reliability assessment measures numerically even if we do not derive these probability distributions analytically, and that we can obtain useful information in software reliability assessment, such as results of interval estimations on the model parameters, the number of remaining faults, and software reliability. This approach is very useful for the case that we cannot collect enough number of data and we have to conduct interval estimation for complex estimators. However, regarding bootstrap confidence intervals, we encountered a problem that we could not get appropriate interval estimations in the basic, standard normal, and bootstrap-*t*

confidence intervals for the number of remaining faults at the termination time of the testing. This problem was solved by using other bootstrap confidence intervals, such as the percentile and the *BCa* confidence intervals. In the future studies, we are going to apply our bootstrap approach for estimating optimal software release time and other practical software project management issues.

6. Acknowledgements

The second author is supported in part by the Grant-in-Aid for Scientific Research (C), Grant No. 22510150, from the Ministry of Education, Culture.

REFERENCES

[1] J. D. Musa, "A Theory of Software Reliability and Its Application," *IEEE Transactions on Software Engineering*, Vol. SE-1, No. 3, 1975, pp. 312-327. doi:10.1109/TSE.1975.6312856

[2] A. L. Goel, "Software Reliability Models: Assumptions, Limitations, and Applicability," *IEEE Transactions on Software Engineering*, Vol. SE-11, No. 12, 1985, pp. 1411-1423. doi:10.1109/TSE.1985.232177

[3] J. D. Musa, D. Iannio and K. Okumoto, "Software Reliability: Measurement, Prediction, Application," McGraw-Hill, New York, 1987.

[4] H. Pham, "Software Reliability," Springer-Verlag, Singapore, 2000.

[5] S. Inoue and S. Yamada, "Discrete Software Reliability Assessment with Discretized NHPP Models," *Computers & Mathematics with Applications*: *An International Journal*, Vol. 51, No. 2, 2006, pp. 161-170. doi:10.1016/j.camwa.2005.11.022

[6] S. Inoue and S. Yamada, "Integrable Difference Equations for Software Reliability Assessment and Their Applications," *International Journal of Systems Assurance Engineering and Management*, Vol. 1, No. 1, 2010, pp. 2-7. doi:10.1007/s13198-010-0005-x

[7] S. Yamada and S. Osaki, "Software Reliability Growth Modeling: Models and Applications," *IEEE Transactions on Software Engineering*, Vol. SE-11, No. 12, 1985, pp. 1431-1437. doi:10.1109/TSE.1985.232179

[8] B. Efron, "Bootstrap Methods: Another Look at the Jackknife," *The Annals of Statistics*, Vol. 7, No. 1, 1979, pp. 1-26. doi:10.1214/aos/1176344552

[9] M. Kimura, "A study on Bootstrap Confidence Intervals of Software Reliability Measures Based on an Incomplete Gamma Function Model," In: T. Dohi and W. Y. Yun, Eds., *Advanced Reliability Modeling II*, World Scientific, Singapore City, 2006, pp. 419-426.

[10] M. Kimura and T. Fujiwara, "A Bootstrap Software Reliability Assessment Method to Squeeze out Remaining Faults," In: T. H. Kim and H. Adeli, Eds., *Advances in Computer Science and Information Technology*, Springer-Verlag, Berlin-Heidelberg, 2010, pp. 435-446. doi:10.1007/978-3-642-13577-4_39

[11] T. Kaneishi and T. Dohi, "Parametric Bootstrapping for Assessing Software Reliability Measures," *Proceedings of the* 17*th IEEE Pacific Rim International Symposium on Dependable Computing*, 12-14 December 2010, pp. 1-9.

[12] S. Tokumoto, T. Dohi and W. Y. Yun, "Toward Development of Risk-Based Checkpointing Scheme via Parametric Bootstrapping," *Proceedings of the* 2012 *Workshop on Recent Advances in Software Dependability*, 19 November 2012, pp. 50-55.

[13] S. Yamada and S. Osaki, "Discrete Software Reliability Growth Models," *Journal of Applied Stochastic Models and Data Analysis*, Vol. 1, No. 1, 1985, pp. 65-77. doi:10.1002/asm.3150010108

[14] R. Hirota, "Nonlinear Partial Difference Equations. V. Nonlinear Equations Reducible to Linear Equations," *Journal of the Physical Society of Japan*, Vol. 46, No. 1, 1979, pp. 312-319. doi:10.1143/JPSJ.46.312

[15] D. Satoh, "A Discrete Gompertz Equation and a software Reliability Growth Model," *IEICE Transactions on Information and Systems*, Vol. E83-D, No. 7, 2000, pp. 1508-1513.

[16] D. Satoh, "A Discrete Bass Model and Its Parameter Estimation," *Journal of the Operations Research Society of Japan*, Vol. 44, No. 1, 2001, pp. 1-18.

[17] M. L. Rizzo, "Statistical Computing with R," Chapman and Hall/CRC, Boca Raton, 2008.

[18] B. Efron, "Better Bootstrap Confidence Intervals," *Journal of the American Statistical Association*, Vol. 82, No. 397, 1987, pp. 171-185. doi:10.1080/01621459.1987.10478410

