数理解析研究所講究録
第 1939 巻 2015 年 256-261

ソフトウェア信頼性評価のための チェンジポイントモデルに基づいた MTBF の推定に関する一考察
On MTBF Estimation Based on a Change-Point Model for Software Reliability Assessment

鳥取大学・大学院工学研究科 井上 真二, 山田 茂
Shinji Inoue and Shigeru Yamada
Graduate School of Engineering,
Tottori University

1 Introduction

Software reliability growth model [5] is known as one of the mathematical tools for quantitative software reliability assessment. Among the software reliability growth models proposed so far, nonhomogeneous Poisson process (NHPP)-based models, which are called *NHPP models*, are widely applied to practical software reliability assessment due to their applicability and simple mathematical structure. It is known that there exists a case that the characteristic of the software failure-occurrence or the fault-detection phenomenon changes notably in an actual testing-phase of a software development process due to changes of some factors which are related to the software reliability growth process. And such changes influence on the accuracy of software reliability assessment based on the software reliability growth model. Generally, testing-time when such changes are observed is ordinarily called a change-point [6]. Taking the effect of change-point on the software reliability growth process into consideration in the software reliability modeling is one of the important issues for accurate software reliability assessment.

From the background mentioned above, software reliability growth models with the effect of change-point have been proposed so far [2, 3, 6, 7]. These models might contribute the accuracy improvement of software reliability assessment based on a software reliability growth model. However, it is known that the probability distribution of the software failure-occurrence time-interval has *improper* or *defective* property [1] especially in the NHPP models and the NHPP-based change-point models in which the total number of detectable faults is assumed to be finite. This property leads to inconvenience situation in quantitative software reliability assessment. That is, we cannot precisely derive the mean time between software failures (MTBF or MTBSF). Therefore, we usually use the instantaneous or cumulative MTBF [5] as the substitute measures for the proper MTBF. We apply the mending method proposed by Grottke and Trivedi [1] to our NHPP-based change-point modeling framework [3], and proposed an all-stage truncated change-point modeling framework. Finally, we show numerical examples of application of our proposed change-point model to software reliability assessment, and discuss the importance of using the proper MTBF in software reliability assessment by using actual fault counting data.

2 NHPP Modeling Framework

Let $\{N(t),\ t \geq 0\}$ denote a counting process representing the total number of faults detected up to testing-time t. From the basic assumptions [4], the probability that m faults are detected up to testing-time t is derived as

$$\Pr\{N(t)=m\} = \sum_{n} \binom{n}{m} \{F(t)\}^{m}\{1-F(t)\}^{n-m} \Pr\{N_0=n\}, \tag{1}$$

in which N_0 represents the initial fault content. From Eq. (1), we can derive an NHPP with mean value function $\mathrm{E}[N(t)] \equiv \Lambda(t) = \omega F(t)$ if we assume that N_0 follows a Poisson distribution with mean ω. The software reliability function $R(x \mid t)$, which means the probability that a software failure does not occur in the time-interval $(t, t+x]$, is derived as

$$R(x \mid t, N(t) = i-1) = \exp\left[-\omega\{F(x+t) - F(x)\}\right]. \tag{2}$$

We should note that the corresponding probability distribution function $G(x \mid t) \equiv 1 - R(x \mid t)$ has the following properties:

$$\lim_{x \to 0} G(x \mid t) = 1 - \lim_{x \to 0} R(x \mid t) = 0, \tag{3}$$

$$\lim_{x \to \infty} G(x \mid t) = 1 - \lim_{x \to 0} R(x \mid t) = 1 - \exp\left[-\omega(1 - F(t))\right]. \tag{4}$$

Therefore, the probability distribution function $G(x \mid t)$ is *defective* or *improper.* Such defective distribution implies that there is the possibility no failure will occur at all. This might be unrealistic especially for the first software failure-occurrence time distribution, $F(x \mid 0)$, except for thoroughly tested software. And, we cannot precisely derive the MTBF.

Further, the conditional distribution of the number of faults remaining $N_0 - N(t)$ given that $N(t) = i-1$ follows Poisson distribution with mean $\omega(1 - F(t))$ because

$$\begin{aligned}\Pr\{N_0 = n \mid N(t) = i-1\} &= \frac{\Pr\{N(t) = i-1 \mid N_0 = n\}\Pr\{N_0 = n\}}{\sum_{k=i-1}^{\infty} \Pr\{N(t) = i-1 \mid N_0 = k\}\Pr\{N_0 = k\}} \\ &= \frac{\{\omega(1-F(t))\}^{n-(i-1)}}{\{n-(i-1)\}!} \exp\left[-\omega(1-F(t))\right] \qquad (n \geq i-1).\end{aligned} \tag{5}$$

From Eq. (5), we can see that $\Pr\{N_0 = i-1 \mid N(t) = i-1\} = \exp\left[-\omega(1-F(t))\right]$. This equation is the same as Eq. (4). This implies that no additional faults are left in the software. These properties of the NHPP models are uncomfortable for us in realistic software reliability modeling.

3 Change-Point Modeling

Now we define the following stochastic quantities being related to our modeling approach in this paper: X_i: the i-th software failure-occurrence time before change-point ($X_0 = 0, \quad i = 0, 1, 2, \cdots$), S_i: the i-th software failure-occurrence time-interval before change-point ($S_i = X_i - X_{i-1}, \quad S_0 = 0, \quad i = 1, 2, \cdots$), Y_i: the i-th software failure-occurrence time after change-point ($Y_0 = 0, \quad i = 0, 1, 2, \cdots$), T_i: the i-th software failure-occurrence time-interval after change-point ($T_i = Y_i - Y_{i-1}, \quad T_0 = 0, \quad i = 1, 2, \cdots$).

We assume that the stochastic quantities before and those after change-point have the following relationships: $Y_i = \alpha(X_i), T_i = \alpha(S_i), J_i(\alpha^{-1}(t)) = K_i(t)$, respectively, where $\alpha(t)$ is a testing-environmental function representing the relationship between the stochastic quantities of the software failure-occurrence times or time-intervals before change-point and those after change-point, $J_i(t)$ and $K_i(t)$ the probability distribution functions with respect to the random variables S_i and T_i, respectively. In our change-point modeling, we assume that the testing-environmental function is given as $\alpha(t) = \alpha t \quad (\alpha > 0)$, where α is the proportional constant representing the relative magnitude of the effect of change-point on the software reliability growth process. Suppose that n faults have been detected up to change-point and their fault-detection times from the test-beginning ($t = 0$) have been observed as $0 < x_1 < x_2 < \cdots < x_n < \tau$, where τ represents change-point. Then, the probability distribution function of T_1, a random variable representing the time-interval from the change point to the 1-st software failure-occurrence after change-point, can be derived as

$$\begin{aligned}\bar{K}_1(t) \equiv \Pr\{T_1 > t\} &= \frac{\Pr\{S_{n+1} > \tau - x_n + t/\alpha\}}{\Pr\{S_{n+1} > \tau - x_n\}} \\ &= \frac{\exp[-\{\Lambda_B(\tau + t/\alpha) - \Lambda_B(x_n)\}]}{\exp[-\Lambda_B(\tau) - \Lambda_B(x_n)]},\end{aligned} \tag{6}$$

where $\bar{K}_1(t)$ indicates the cofunction of the probability distribution function $K_1(t) \equiv \Pr\{T_1 \le t\}$, i.e., $\bar{K}_1(t) \equiv 1 - K_1(t)$, and $\Lambda_B(t)(\equiv \omega K_1(t))$ represents the expected number of faults detected up to change-point, i.e., a mean value function for the NHPP before change-point. From Eq. (6), the expected number of faults detected up to $t \in (\tau, \infty]$ after change-point, $M_A(t)$, can be formulated as

$$\begin{aligned} \Lambda_A(t) &= -\log \Pr\{T_1 > t - \tau\} = -\log \bar{K}_1(t-\tau) \\ &= \Lambda_B\left(\tau + \frac{t-\tau}{\alpha}\right) - \Lambda_B(\tau). \end{aligned} \tag{7}$$

Then, the expected number of faults detected up to testing-time t ($t \in (0, \infty], 0 < \tau < t$) [3] can be derived as

$$\Lambda(t) = \begin{cases} \Lambda_1(t) = \Lambda_B(t) & (0 \le t \le \tau) \\ \Lambda_2(t) = \Lambda_B(\tau) + \Lambda_A(t) = \Lambda_B\left(\tau + \frac{t-\tau}{\alpha}\right) & (\tau < t). \end{cases} \tag{8}$$

From Eq. (8), we can see that an NHPP-based software reliability growth model with change-point can be developed by assuming a suitable probability distribution function for the software failure-occurrence time before change-point.

4 Mending NHPP Models

Grottke and Trivedi [1] considered to use the zero-truncated Poisson distribution:

$$\Pr\{N_0 = n \mid N_0 > 0\} = \frac{\Pr\{N_0 = n, N_0 > 0\}}{\Pr\{N_0 > 0\}} = \frac{\omega^n}{n!}\frac{e^{-\omega}}{1 - e^{-\omega}} \qquad (n = 1, 2, \cdots), \tag{9}$$

where N_0 follows Poisson distribution with mean ω, for the probability distribution of the initial fault content. Applying Eq. (9) to Eq. (1), we have

$$\begin{cases} \Pr\{N(t) = 0\} = \dfrac{\exp\left[\omega(1 - F(t)\right] - 1}{e^{\omega} - 1}, \\ \Pr\{N(t) = m\} = \dfrac{\{\omega F(t)\}^m}{m!}\dfrac{e^{-\omega F(t)}}{1 - e^{-\omega}} \qquad (m = 1, 2, \cdots). \end{cases} \tag{10}$$

And the mean value function $\Lambda(t)$ can be derived as

$$\Lambda(t) = \sum_{m=1}^{\infty} m \frac{\{\omega F(t)\}^m}{m!}\frac{e^{-\omega F(t)}}{1 - e^{-\omega}} = \frac{\omega F(t)}{1 - e^{-\omega}}. \tag{11}$$

Eqs. (10) and (11) are called the *first-stage truncated model* from the point of view of CTMC software reliability modeling. Actually, the probability distribution of the first software failure-occurrence time satisfies: $G_{X_1}(0 \mid 0) = 0$ and $G_{X_1}(\infty \mid 0) = 1$ because

$$R_{X_1}(x \mid 0, N(0) = 0) = \frac{\exp\left[\omega(1 - F(x))\right] - 1}{e^{\omega} - 1}. \tag{12}$$

And the hazard rates for the $X_2, X_3, \cdots$ are $\omega f(t)$, which means the probability distribution of the other software failure time-interval are defective.

For developing the *all-stage truncated model*, Grottke and Trivedi [1] considers the conditional probability distribution of $N_0 \mid N(t) = i-1$ in Eq. (5). Substituting Eq. (9) into Eq. (5), we have

$$
\begin{aligned}
&\Pr\{N_0 = n \mid N(t) = i-1\} \\
&= \frac{\Pr\{N(t) = i-1 \mid N_0 = n\}\Pr\{N_0 = n\}}{\sum_{k=i-1}^{\infty} \Pr\{N(t) = i-1 \mid N_0 = k\}\Pr\{N_0 = k\}} \\
&= \left[\binom{n}{i-1} F(t)^{i-1}\{1-F(t)\}^{n-(i-1)} \frac{\omega^n}{n!}\frac{e^{-\omega}}{1-e^{-\omega}}\right] \\
&\qquad\qquad \Big/ \left[\sum_{k=i-1}^{\infty} \binom{k}{i-1} F(t)^{i-1}\{1-F(t)\}^{k-(i-1)} \frac{\omega^k}{k!}\frac{e^{-\omega}}{1-e^{-\omega}}\right] \\
&= \frac{\{\omega(1-F(t))\}^{n-(i-1)}}{(n-(i-1))!} \frac{1}{\exp[\omega(1-F(t))]-1} \qquad (n \geq i-1).
\end{aligned} \tag{13}
$$

We should note that

$$
\Pr\{N_0 = n \mid N(0) = 0\} = \frac{\omega^n}{n!}\frac{e^{-\omega}}{1-e^{-\omega}}, \tag{14}
$$

which is the same as Eq. (9), when $i-1=0$ and $t=0$ in Eq. (13). Further, the software reliability function for the all-stage truncated model can be obtained as

$$
\begin{aligned}
&R(x \mid t, N(t) = i-1) \\
&= \sum_{n=i}^{\infty} \Pr\{N(t+x) - N(t) = 0 \mid N_0 = n, N(t) = i-1\}\Pr\{N_0 = n \mid N(t) = i-1\} \\
&= \sum_{i=n}^{\infty} \left\{\frac{1-F(t+x)}{1-F(t)}\right\}^{n-(i-1)} \frac{\{\omega(1-F(t))\}^{n-(i-1)}}{(n-(i-1))!} \frac{1}{\exp[\omega(1-F(t))]-1} \\
&= \frac{\exp[\omega(1-F(t+x))]-1}{\exp[\omega(1-F(t))]-1} \qquad (n \geq i-1).
\end{aligned} \tag{15}
$$

We should note that Eq. (15) tends to zero for $x \to \infty$ and tends to 1 for $x \to 0$. Eq. (15) implies that at least one faults will be eventually detected during infinite testing-time because the probability distributions of the software failure-occurrence time-interval are *non-defective* or *proper.*

5 Mending Change-Point Models

In the all-stage truncated NHPP model, in which the all transition rates are identical, we can derive the mean value function by using the following equation:

$$
\Lambda(t) = -\ln[R(x \mid 0, N(0) = 0)] = \ln\left[\frac{e^{\omega}-1}{\exp[\omega(1-F(t))]-1}\right]. \tag{16}
$$

In Eq. (16), we can see $\Lambda(t) \to \infty$ for $t \to \infty$.

We can derive a mean value function after change-point for the all-stage truncated change-point model as

$$
\begin{aligned}
\Lambda_A(t) &= \Lambda_B\left(\tau + \frac{t-\tau}{\alpha}\right) - \Lambda_B(\tau) \\
&= \ln\left[\frac{e^{\omega}-1}{\exp[\omega\{1-F(\tau+\frac{t-\tau}{\beta})\}]-1}\right] - \ln\left[\frac{e^{\omega}-1}{\exp[\omega\{1-F(\tau)\}]-1}\right],
\end{aligned} \tag{17}
$$

by substituting Eq. (16) into Eq. (7). Then, we have the all-stage truncated change-point model as

$$
\Lambda(t) = \begin{cases}
\Lambda_1(t) = \ln\left[\dfrac{e^{\omega}-1}{\exp[\omega\{1-F(t)\}]-1}\right] & (0 \leq t \leq \tau), \\
\Lambda_2(t) = \ln\left[\dfrac{e^{\omega}-1}{\exp[\omega\{1-F(\tau+\frac{t-\tau}{\beta})\}]-1}\right] & (\tau < t),
\end{cases} \tag{18}
$$

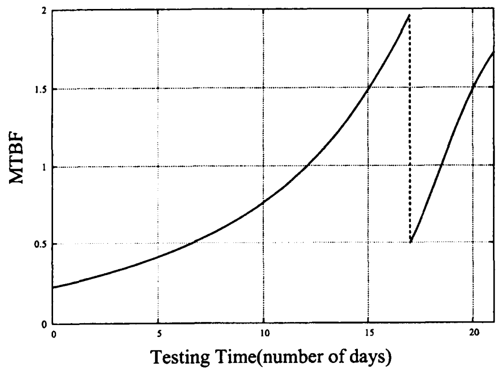

Fig 1 : Estimated MTBF with the effect of change-point, $\widehat{MTBF}(t)$. ($\tau = 17$)

from Eqs. (8) and (17).

6 Numerical Examples

We show numerical examples of application of our change-point modeling to software reliability assessment by using actual fault counting data: $(t_k, y_k)(k = 0, 1, 2, \cdots, 21;\ t_{21} = 21\ \text{(days)}, y_{21} = 39;\ \tau_1 = 17)$. This actual data was fault counting data collected from actual testing-phases for the Windows version software and the change-point was generated by changing the tester and increasing the test personnel.

As one of the examples, we now develop an all-stage truncated change-point model in which the software failure-occurrence time distribution follows an exponential distribution: $F(t) = 1 - \exp[-bt]$. This model can be derived as

$$\Lambda(t) = \begin{cases} \Lambda_1(t) = \ln\left[\dfrac{e^{\omega} - 1}{\exp[\omega \exp[-bt]] - 1}\right] & (0 \le t \le \tau), \\ \Lambda_2(t) = \ln\left[\dfrac{e^{\omega} - 1}{\exp[\omega \exp\{-b(\tau + \frac{t-\tau}{\beta})\}] - 1}\right] & (\tau < t), \end{cases} \tag{19}$$

based on the modeling framework in Eq. (18).

Applying the actual data mentioned above, we estimate the parameters ω, b, and β by using the method of maximum likelihood based on the NHPP. Consequently, we respectively obtained $\widehat{\omega} = 39.5061$, $\widehat{b} = 0.1147$, and $\widehat{\beta} = 0.2516$, in which $\widehat{\omega}$, $\widehat{b}$, and $\widehat{\beta}$ are the estimations of ω, b, and β, respectively. Fig. 1 show the time-dependent behavior of the estimated MTBF, which is derived by

$$MTBF(t) = \int_0^{\infty} R(x \mid t)\mathrm{d}x, \tag{20}$$

by using Eq. (15). For comparing the time-dependent behavior of MTBF with it of the cumulative MTBF, which is widely applied as the substitution of the MTBF in the conventional NHPP models, we additionally show the time-dependent behavior of the estimated cumulative MTBF in Fig. 2. The cumulative MTBF is calculated as $MTBF_C(t) = t/\Lambda(t)$. From Figs. 1 and 2, we can say that the time-dependent behaviors of the MTBF and the cumulative MTBF are obviously different each other. This implies the importance of applying the proper MTBF to software reliability assessment. Actually, $\widehat{MTBF}(21) = 1.726$ (days) and $\widehat{MTBF}_C(21) = 0.537$ (days). From these results, we can say that the cumulative MTBF does not work well as the substitution measure for the MTBF.

7 Conclusion

Our change-point model has a useful properties that the probability distributions of the software failure-occurrence time-interval are non-defective and we can analytically derive the proper MTBF, which is

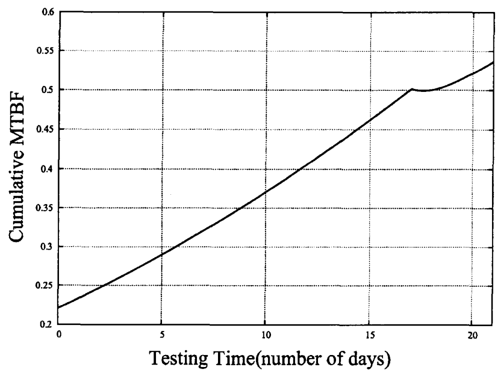

Fig 2 : Estimated cumulative MTBF with the effect of change-point, $\widehat{MTBF_C}(t)$. ($\tau = 17$)

one of the typical reliability assessment measures. Change-point models proposed so far do not have such useful properties. Further we showed numerical examples of application of our all-stage truncated change-point model to software reliability assessment by using actual fault counting data. Especially, we compared the time-dependent behaviors of the cumulative MTBF and the proper MTBF, and obviously showed their differences. In our further studies, we need to check the fitting and predictive performances of our all-stage truncated change-point model by applying a lot of actual data.

Acknowledgement

This research was supported in part by the Grant-in-Aid for Scientific Research (C), Grant No. 22510150, from the Ministry of Education, Culture, Sports, Science and Technology of Japan and the Telecommunications Advancement Foundation.

References

[1] M. Grottke and K.S. Trivedi, "On a method for mending time to failure distributions," *Proceedings of the 2005 International Conference on Dependable Systems and Networks (DNS'05)*, Yokohama, Japan, 28 June–1 July, 2005.

[2] C.Y. Huang, "Performance analysis of software reliability growth models with testing-effort and change-point," *Journal of System and Software*, Vol. 76, No. 2, pp. 181–194, 2005.

[3] S. Inoue and S. Yamada, "Software reliability measurement with effect of change-point," *International Journal of System Assurance Engineering and Management*, Vol. 2, No. 2, pp. 155–162, 2011. (DOI 10.1007/s13198-011-0070-9)

[4] N. Langberg and N.D. Singpurwalla, "A unification of some software reliability models," *SIAM Journal on Scientific Computing*, Vol. 6, No. 3, pp. 781–790, 1985.

[5] S. Yamada. Software Reliability Modeling — Fundamentals and Applications —. Springer-Verlag, Tokyo, 2013.

[6] M. Zhao, "Change-point problems in software and hardware reliability," *Communications in Statistics —Theory and Methods*, Vol. 22, No. 3, pp. 757–768, 1993.

[7] F.Z. Zou, "A change-point perspective on the software failure process," *Software Testing, Verification and Reliability*, Vol. 13, No. 2, pp. 85–93, 2003.