Software Reliability Modeling with Periodic Debugging Schedule

Technical Report No. ASU/2013/5

Dated: 7^{th} November, 2013

Sudipta Das

SQC and OR Unit, Indian Statistical Institute

Anup Dewanji

Applied Statistics Unit, Indian Statistical Institute

Anshu Sobti

SQC and OR Unit, Indian Statistical Institute

And

Ashis Chakraborty

SQC and OR Unit, Indian Statistical Institute

Applied Statistics Unit

Indian Statistical Institute

Kolkata 700 108

Software Reliability Modeling

with Periodic Debugging Schedule

SUMMARY. In this article, we discuss continuous time periodic debugging of a software, in which errors are not corrected at the instants of their detection but at some pre-specified times. We describe maximum likelihood estimation for the model parameters assuming exponential distribution for the time between successive occurrence of an error, which can be used to estimate the reliability of the software. We also derive some asymptotic properties of the estimated model parameters, specially the number of errors initially present in the software. We investigate the final sample properties of the estimates, specially that of the number of errors, through simulation and also study the robustness of the estimate against deviation from the exponential assumption. Finally, we analyze a dataset obtained from the testing of a flight control software.

KEY WORDS: Continuous time scale, Periodic debugging, Jelinski-Moranda model, Failure intensity, Profile likelihood.

1 Introduction

Starting with Jelinski and Moranda [6], software reliability modeling has passed through many refinements (See Musa et al. [8], Nayak [10], Singpurwalla and Wilson [12], Lyu [7], Gokhale et al. [5], Dalal, 2002, among many others), however, keeping some of the basic underlying assumptions pretty much the same. Most of the models require a continuous time scale (like calendar or execution time or staff hours) with finite (unknown) number of errors, ν say, in the software and are specified by the failure (detection of an error) intensity. One major assumption made by all these models is that the errors are removed as and when they are detected. In many practical situations, it is not the case. Usually, there are some prefixed time points when debugging of errors, which are detected since the previous debugging, takes place. The software testing goes on till the last debugging time. It implies that, the errors are not removed as and when they are detected causing failures of the software during testing; only a record of these detected errors is maintained and these are removed, with certainty and introducing no other new error, at the subsequent scheduled debugging time. We call this design of software testing and data collection a 'periodic debugging schedule'. For example of such a design, we refer to a report (Dewanji et al, 2005) on a project sponsored by Indian Space Research Organization. This kind of periodic debugging schedule may be necessary when subsequent versions of the software are released at different times and testing continues with the most recent version [3].

In this work, we focus on estimating ν, by assuming that incidence of all errors follow independent identically distributed Poisson processes with a common but unknown rate λ. Under the Poisson assumption, the maximum likelihood estimates (MLEs) of ν and λ as well as their asymptotic distributions have been obtained. We show that, under some condition, $\hat{\nu}$ can be infinite. Because of the mathematical and computational simplicity of the estimator, we also study its sensitivity while data comes from some IFR and DFR distributions, other than exponential.

The article is organized as follows. In Section 2, we consider a continuous time scale Jelinski-Moranda type software reliability model to suit the periodic debugging schedule. A computational method for obtaining $\hat{\nu}$ is provided in Section 3. In Section 4, we discuss some asymptotic properties of $\hat{\nu}$, as $\nu \to \infty$. Section 5 reports results of a simulation study to investigate the properties of the estimates developed in Section 3. In Section 6, we illustrate our estimation method through the analysis of a real life data.

2 Modeling and Likelihood

The Jelinski-Moranda model ([6]) is credited with being the first and simplest software reliability model. In order to give a brief review of this model (since we propose to extend this model to the case of periodic debugging schedule), note that the following assumptions form the basis of the Jelinski-Moranda model:

1. Each of the ν errors in a software system will cause a failure after a time which is distributed exponentially.

2. The errors are independent of each other and equally likely to cause a failure.

3. The failure rate at any time is proportional to the current error content of a software system.

4. A detected error is corrected with certainty in a negligible amount of time and no new errors are introduced.

Assuming that there are initially ν errors in the software, each having exponential(λ) detection time, the failure rate is initially $\nu\lambda$ and remains constant at $\nu\lambda$ until the first error is detected. The failure rate decreases to $(\nu-1)\lambda$ after the first error is detected as well as corrected and decreases by an amount λ after each successive error detection and correction. In general, the failure rate during the ith time interval, between the $(i-1)$th failure and the ith failure, is given by $\lambda_i = (\nu - i + 1)\lambda$, for $i = 1, \cdots, r$, where r denotes the number of failures (or, detected errors) by the observation (or, testing) time τ, say.

In the context of periodic debugging, suppose there are k prefixed time points, $0 = t_0 < t_1 < \cdots < t_k < \infty$, for debugging; that is, at each of these time points, debugging, if necessary, is scheduled to take place. The software testing terminates at the last debugging time t_k. Suppose we observe the number of failures m_i (≥ 0) between t_{i-1} and t_i along with the identities of the corresponding errors, for $i = 1, \cdots, k$. Note that a particular error may be responsible for more than one failure during a time interval $(t_{i-1}, t_i]$; that is, the m_i failures may correspond to fewer than m_i distinct errors. Since we have the identities of the detected errors, let m_i^d (≥ 0) denote the number of distinct errors of the m_i failures which take place between t_{i-1} and t_i. These m_i^d errors are removed or debugged at time t_i, after which these errors do not appear in the subsequent time intervals. Let us write M_i to denote the cumulative number of distinct errors removed on or before the ith debugging time t_i, for $i = 0, 1, \cdots, k$, with $M_0 = 0$; that is, $M_i = \sum_{l=1}^{i} m_l^d$. Note that M_k is the number of detected errors by the last debugging time t_k. Therefore, since the failure rates during the period between successive debugging times remain the same and are assumed to be independent of time, the relevant observation from such a design of periodic debugging consists of $\{(m_i, m_i^d),\ i = 1, \cdots, k\}$. Let us also write H_i to denote the history of the testing process up to the ith debugging time t_i, for $i = 1, \cdots, k$, so that $H_i = \{(m_l, m_l^d),\ l = 1, \cdots, i\}$ and H_0 is empty. Note that, when the failure rates are not time-independent, then the times of different failures will also be a part of observation and, hence, of H_i's.

Clearly, since no debugging takes place between 0 and t_1, the occurrence of failures in $(0, t_1]$ follows a Poisson process with rate $\nu\lambda$. Therefore, the distribution of m_1 is Poisson with expectation $\nu\lambda t_1$. In general, given the history H_{i-1} up to time t_{i-1} including the number M_{i-1} of distinct errors debugged so far, the conditional distribution of m_i is

Poisson with expectation $(\nu - M_{i-1})\lambda\Delta t_i$, with $\Delta t_i = t_i - t_{i-1}$, for $i = 1, \cdots, k$. Therefore,

$$\begin{aligned} P[m_i|H_{i-1}] &= \frac{e^{-(\nu-M_{i-1})\lambda\Delta t_i}\left[(\nu - M_{i-1})\lambda\Delta t_i\right]^{m_i}}{m_i!} \\ &\propto e^{-(\nu-M_{i-1})\lambda\Delta t_i}\left[(\nu - M_{i-1})\lambda\right]^{m_i}. \end{aligned} \tag{1}$$

Also, it can be verified that the conditional probability of m_i^d, given m_i and H_{i-1}, is given by ([4], p102)

$$\begin{aligned} P[m_i^d|m_i, H_{i-1}] &= \frac{(\nu - M_{i-1})!}{(\nu - M_i)!(\nu - M_{i-1})^{m_i}}\left[\Sigma_{k=0}^{m_i^d}(-1)^k\frac{(m_i^d - k)^{m_i}}{k!(m_i^d - k)!}\right] \\ &\propto \frac{(\nu - M_{i-1})!}{(\nu - M_i)!(\nu - M_{i-1})^{m_i}}. \end{aligned} \tag{2}$$

Using (1) and (2), the likelihood function can be written as

$$\begin{aligned} L(\nu, \lambda) &= \prod_{i=1}^{k} P[m_i, m_i^d|H_{i-1}] \\ &= \prod_{i=1}^{k} P[m_i|H_{i-1}]P[m_i^d|m_i, H_{i-1}] \\ &\propto \prod_{i=1}^{k}\left[e^{-(\nu-M_{i-1})\lambda\Delta t_i}\lambda^{m_i} \times \frac{(\nu - M_{i-1})!}{(\nu - M_i)!}\right] \\ &= e^{-\lambda\sum_{i=1}^{k}(\nu-M_{i-1})\Delta t_i}\lambda^m \times \frac{\nu!}{(\nu - M_k)!}, \end{aligned} \tag{3}$$

where $m = \sum_{i=1}^{k} m_i$.

3 Maximum Likelihood Estimation

As discussed earlier, we consider ν as the parameter of primary interest and present a profile likelihood method to obtain the MLE of ν. However, by this method, λ is also estimated as a by-product. Note that, when $M_k = 0$, the likelihood is $\exp[-\lambda\nu t_k]$, which gives the MLE of ν as zero for any λ. So, we consider the case $M_k \geq 1$.

The likelihood (3) gives the MLE of λ, for a given ν, as

$$\hat{\lambda}(\nu) = \frac{m}{\sum_{i=1}^{k}(\nu - M_{i-1})\Delta t_i}. \tag{4}$$

Substituting $\hat{\lambda}(\nu)$ in (3), the profile likelihood of ν is

$$L(\nu) \propto \frac{1}{\left(\sum_{i=1}^{k}(\nu - M_{i-1})\Delta t_i\right)^m} \times \frac{\nu!}{(\nu - M_k)!}.$$

Hence,

$$\frac{L(\nu+1)}{L(\nu)} = \left(\frac{\nu+1}{\nu+1-M_k}\right) \frac{(\sum_{i=1}^{k}(\nu - M_{i-1})\Delta t_i)^m}{(\sum_{i=1}^{k}(\nu - M_{i-1})\Delta t_i + t_k)^m}, \text{ for } \nu \geq M_k.$$

If $\frac{L(\nu+1)}{L(\nu)} \leq 1$ for all ν, then the MLE of ν is $\hat{\nu} = M_k$. While estimating ν, there can be situations when $\hat{\nu} = \infty$; that is, $\frac{L(\nu+1)}{L(\nu)} \geq 1$ for all ν. The following proposition states a necessary and sufficient condition for infinite $\hat{\nu}$.

Proposition:- The MLE $\hat{\nu} = \infty$ if and only if $M_k = m$ and $m \geq 1 + 2B$, where $B = \frac{1}{t_k}\Sigma_{i=1}^{k} M_{i-1}\Delta t_i$.

Proof:- Write

$$g(\nu) = \frac{L(\nu)}{L(\nu+1)} = \left(1 - \frac{M_k}{\nu+1}\right)\left(1 + \frac{1}{\nu - B}\right)^m.$$

Note that $m \geq M_k \geq B$. Write

$$f(\nu) = \log g(\nu) = \log\left(1 - \frac{M_k}{\nu+1}\right) + m\log\left(1 + \frac{1}{\nu - B}\right).$$

Treating ν as a continuous variable, we have

$$\frac{d}{d\nu}f(\nu) = \frac{(M_k - m)\nu^2 + (M_k - 2BM_k - 2m + mM_k)\nu + (B^2M_k - BM_k - m + mM_k)}{(\nu+1)(\nu+1-M_k)(\nu-B)(\nu+1-B)}.$$

Now, we investigate the slope of $f(\nu)$, equivalently the slope of $g(\nu)$. We consider the following three exhaustive cases.

Case 1: $M_k = m$ and $m \geq 1 + 2B$. Then

$$f'(\nu) = \frac{m(m-1-2B)\nu + m(B^2 - B - 1 + m)}{(\nu+1)(\nu+1-m)(\nu-B)(\nu+1-B)}$$

is always positive. Therefore, $f(\nu)$, equivalently $g(\nu)$, is always increasing and $\lim_{\nu\to\infty} g(\nu) = 1$. Hence, $g(\nu)$ can not go above 1, else it can not come back to 1. Thus, $L(\nu) < L(\nu+1)$, for all $\nu \geq M_k$, implying $\hat{\nu} = \infty$.

Case 2: $M_k = m$ and $m < 1{+}2B$. The leading term of $f'(\nu)$ (that is, $\frac{m(m-1-2B)\nu}{(\nu+1)(\nu+1-M_k)(\nu-B)(\nu+1-B)}$) is negative. Therefore, for large ν, $f(\nu)$, equivalently, $g(\nu)$, is decreasing, and $\lim_{\nu\to\infty} g(\nu) = 1$. Hence, there exists a ν_0 such that $g(\nu) > 1$, for all $\nu \geq \nu_0$. Thus, $\hat{\nu} < \infty$.

Case 3: $M_k \neq m$ (that is, $m > M_k$). Then, the leading term of $f'(\nu)$ (that is, $\frac{(M_k-m)\nu^2}{(\nu+1)(\nu+1-M_k)(\nu-B)(\nu+1-B)}$) is also negative. By a similar argument, as in Case 2, we can claim that $\hat{\nu} < \infty$.

Hence, the proof is completed. The proposition indicates that the function $g(\nu)$ has only one maximum. Therefore, under the conditions of Cases 2 and 3, we compute $g(\nu)$, for $\nu = M_k, M_k{+}1, \ldots$ one by one and the MLE of ν is given by $\hat{\nu} = \min\{\nu \geq M_k : g(\nu) \geq 1\}$. The MLE of λ can be obtained as $\hat{\lambda} = \hat{\lambda}(\hat{\nu})$, using (4).

Note that, when k=1, (that is, there is only one period of debugging), then this necessary and sufficient condition for $\hat{\nu} = \infty$ reduces to $M_1 = m \geq 1$ (see Nayak [9]).

4 Asymptotic Results

In order to derive the asymptotic properties of the estimates, $\hat{\lambda}$ and $\hat{\nu}$ (say) of λ and ν, respectively, obtained by the method of Section 3, following Dewanji et al. ([2]), let the ν errors be labeled as $1, \cdots, \nu$ and let X_j denote the (unknown) observation from the jth error. If the jth error is not detected, let us write $X_j = 0$; otherwise, X_j consists of the debugging time t_{i_j} (say) of the jth error and the number m_j^* (say) of times it appears between t_{i_j-1} and t_{i_j}. The X_j's are clearly hypothetical since the labeling $1, \cdots, \nu$ is not observed; however, the X_j's are independent and identically distributed with the probability distribution given by

$$p_X(x_j;\lambda) = \begin{cases} e^{-\lambda t_k}, & \text{if } x_j = 0 \\ \frac{\lambda^{m_j^*}(t_{i_j}-t_{i_j-1})^{m_j^*}}{m_j^*!} e^{-\lambda t_{i_j}}, & \text{otherwise.} \end{cases}$$

The joint distribution of $(X_1, \cdots, X_\nu)$ is, therefore, $L^*(\nu,\lambda) = \prod_{j=1}^{\nu} p_X(x_j;\lambda)$. Note that the observed data is a function of the unobserved X_j's. Hence, the observed likelihood

(3) can be obtained from the joint distribution of the X_j's and is given by

$$\frac{\nu!}{(\nu - M_k)!} \prod_{j=1}^{\nu} p_X(x_j; \lambda).$$

Following Dewanji et al. ([2]), define

$$\begin{aligned} V_{1j} &= \frac{d \ \log \ p_X(x_j; \lambda)}{d \ \lambda} \\ &= \begin{cases} -t_k, & \text{if } x_j = 0 \\ \frac{m_j^*}{\lambda} - t_{i_j}, & \text{otherwise.} \end{cases} \end{aligned}$$

and

$$V_{2j} = \begin{cases} 1, & \text{if } x_j \neq 0 \\ -\frac{1-e^{-\lambda t_k}}{e^{-\lambda t_k}}, & \text{otherwise,} \end{cases}$$

for $j = 1, \cdots, \nu$. Now,

$$\begin{aligned} E[V_{1j}] &= -t_k e^{-\lambda t_k} + \Sigma_{i_j=1}^{k} \Sigma_{m_j^*=1}^{\infty} \left(\frac{m_j^*}{\lambda} - t_{i_j} \right) \lambda^{m_j^*} \frac{(t_{i_j} - t_{i_j-1})^{m_j^*}}{m_j^*!} e^{-\lambda t_{i_j}} \\ &= -t_k e^{-\lambda t_k} + \Sigma_{i_j=1}^{k} \left(t_{i_j} e^{-\lambda t_{i_j}} - t_{i_j-1} e^{t_{i_j-1}} \right) \\ &= 0 \end{aligned}$$

and

$$\begin{aligned} Var[V_{1j}] &= E[-\frac{d^2}{d\lambda^2} \ \log \ p_X(x_j; \lambda)] \\ &= \Sigma_{i_j=1}^{k} \Sigma_{m_j^*=1}^{\infty} \left(\frac{m_j^*}{\lambda^2} \right) \lambda^{m_j^*} \frac{(t_{i_j} - t_{i_j-1})^{m_j^*}}{m_j^*!} e^{-\lambda t_{i_j}} \\ &= \Sigma_{i_j=1}^{k} \frac{(t_{i_j} - t_{i_j-1}) e^{-\lambda t_{i_j}}}{\lambda} \Sigma_{m_j^*=1}^{\infty} \frac{\lambda^{m_j^*-1} (t_{i_j} - t_{i_j-1})^{m_j^*-1}}{(m_j^* - 1)!} \\ &= \frac{1}{\lambda} \Sigma_{i_j=1}^{k} \left(t_{i_j} - t_{i_j-1} \right) e^{-\lambda t_{i_j-1}} \\ &= I(\lambda), \text{ (say).} \end{aligned}$$

Similarly,

$$\begin{aligned} E[V_{2j}] &= -\frac{1 - e^{-\lambda t_k}}{e^{-\lambda t_k}} e^{-\lambda t_k} + \Sigma_{i_j=1}^{k} \Sigma_{m_j^*=1}^{\infty} \lambda^{m_j^*} \frac{(t_{i_j} - t_{i_j-1})^{m_j^*}}{m_j^*!} e^{-\lambda t_{i_j}} \\ &= 0 \end{aligned}$$

and

$$
\begin{aligned}
Var[V_{2j}] &= \left(-\frac{1-e^{-\lambda t_k}}{e^{-\lambda t_k}}\right)^2 e^{-\lambda t_k} + \Sigma_{i_j=1}^{k}\Sigma_{m_j^*=1}^{\infty}\lambda^{m_j^*}\frac{(t_{i_j}-t_{i_j-1})^{m_j^*}}{m_j^*!}e^{-\lambda t_{i_j}} \\
&= (1-e^{-\lambda t_k})^2 e^{\lambda t_k} + (1-e^{-\lambda t_k}) \\
&= \frac{1-e^{-\lambda t_k}}{e^{-\lambda t_k}}.
\end{aligned}
$$

The covariance between V_{1j} and V_{2j} is

$$
\begin{aligned}
Cov[V_{1j}, V_{2j}] &= t_k\left(\frac{1-e^{-\lambda t_k}}{e^{-\lambda t_k}}\right)e^{-\lambda t_k} + +\Sigma_{i_j=1}^{k}\Sigma_{m_j^*=1}^{\infty}\left(\frac{m_j^*}{\lambda}-t_{i_j}\right)\lambda^{m_j^*}\frac{(t_{i_j}-t_{i_j-1})^{m_j^*}}{m_j^*!}e^{-\lambda t_{i_j}} \\
&= t_k\left(1-e^{-\lambda t_k}\right) + \Sigma_{i_j=1}^{k}\left(t_{i_j}e^{-\lambda t_{i_j}} - t_{i_j-1}e^{t_{i_j-1}}\right) \\
&= t_k
\end{aligned}
$$

We write

$$
\begin{aligned}
u_{1\nu} &= \nu^{-1/2}\sum_{j=1}^{\nu}V_{1j} \\
&= \nu^{-1/2}\left(-(\nu-M_k)t_k + \Sigma_{j=1}^{M_k}(\frac{m_j*}{\lambda}-t_{i_j})\right) \\
&= \nu^{-1/2}\left(\frac{m}{\lambda}-\nu t_k + M_k t_k - \Sigma_{j=1}^{M_k}t_{i_j}\right) \\
&= \nu^{-1/2}\left(\frac{m}{\lambda}-\nu t_k + \Sigma_{j=1}^{k}M_{j-1}\Delta t_{i_j}\right)
\end{aligned}
$$

and

$$
\begin{aligned}
u_{2\nu} &= \nu^{-1/2}\sum_{j=1}^{\nu}V_{2j} \\
&= \nu^{-1/2}\left((\nu-M_k)\left(-\frac{1-e^{-\lambda t_k}}{e^{-\lambda t_k}}\right)-M_k\right) \\
&= \nu^{-1/2}\left(\frac{M_k-\nu+\nu e^{-\lambda t_k}}{e^{-\lambda t_k}}\right),
\end{aligned}
$$

so that, $1-\nu^{-1/2}u_{2\nu} = (\nu-M_k)/(\nu e^{-\lambda t_k})$. Thus, by the central limit theorem,

$$
\begin{pmatrix} u_{1\nu} \\ u_{2\nu} \end{pmatrix} \stackrel{d}{=} \nu^{-1/2}\sum_{j=1}^{\nu}\begin{pmatrix} V_{1j} \\ V_{2j} \end{pmatrix} \xrightarrow{L} N(0,\Sigma^{-1}) \text{ as } \nu\to\infty,
$$

where

$$\Sigma = \begin{pmatrix} I(\lambda) & t_k \\ t_k & \left(1-e^{-\lambda t_k}\right)/e^{-\lambda t_k} \end{pmatrix}^{-1}.$$

We get,

$$\log L(\nu,\lambda) = m\log\lambda + \Sigma_{j=0}^{M_k-1}\log(\nu-j) - \lambda\Sigma_{i=1}^{k}(\nu-M_{i-1})\Delta t_i$$

from (3).

Thus, for bounded (a,b), following the technique of Dewanji et al. ([2]), we consider

$$\begin{aligned}
& \log L(\nu+\nu^{1/2}a, \lambda+\nu^{-1/2}b) - \log L(\nu,\lambda) \\
= \; & m\log\left(1+\frac{\nu^{-1/2}b}{\lambda}\right) + \Sigma_{j=0}^{M_k-1}\log\left(1+\frac{\nu^{1/2}a}{\nu-j}\right) - \nu^{-1/2}b\Sigma_{i=1}^{k}\left(\nu-M_{i-1}\right)\Delta t_i - \left(\lambda+\nu^{-1/2}b\right)\nu^{1/2}at_k \\
= \; & m\log\left(1+\frac{\nu^{-1/2}b}{\lambda}\right) - \nu^{1/2}bt_k + \nu^{-1/2}b\Sigma_{i=1}^{k}M_{i-1}\Delta t_i - \lambda\nu^{1/2}at_k - abt_k \\
& +a\nu^{1/2}\Sigma_{j=0}^{M_k-1}\frac{1}{\nu-j} - \frac{a^2\nu}{2}\Sigma_{j=0}^{M_k-1}\frac{1}{(\nu-j)^2} + \frac{a^3\nu^{3/2}}{3}\Sigma_{j=0}^{M_k-1}\frac{1}{(\nu-j)^3} + \cdots \\
= \; & m\left(\frac{\nu^{-1/2}b}{\lambda} - \frac{\nu^{-1}b^2}{2\lambda^2} + \frac{\nu^{-3/2}b^3}{3\lambda^3} - \cdots\right) - \nu^{1/2}(b+\lambda a)t_k + \nu^{-1/2}b\Sigma_{i=1}^{k}M_{i-1}\Delta t_i - abt_k \\
& +a\nu^{1/2}\left(\log\nu - \log(\nu-M_k) + O(\nu^{-1})\right) - \frac{a^2}{2}\left(\frac{M_k}{\nu-M_k} + O(\nu^{-1})\right) + \frac{a^3}{3}O(\nu^{-1/2}) + \cdots \\
= \; & m\left(\frac{\nu^{-1/2}b}{\lambda} - \frac{\nu^{-1}b^2}{2\lambda^2} + \frac{\nu^{-3/2}b^3}{3\lambda^3} - \cdots\right) - \nu^{1/2}(b+\lambda a)t_k + \nu^{-1/2}b\Sigma_{i=1}^{k}M_{i-1}\Delta t_i - abt_k \\
& -a\nu^{1/2}\log\left(e^{-\lambda t_k}(1-\nu^{-1/2}u_{2\nu}) + O(\nu^{-1})\right) - \frac{a^2}{2}\left(\frac{(1-e^{-\lambda t_k})+\nu^{-1/2}u_{2\nu}e^{-\lambda t_k}}{(1-\nu^{-1/2}u_{2\nu})e^{-\lambda t_k}} + O(\nu^{-1})\right) \\
& +\frac{a^3}{3}O(\nu^{-1/2}) + \cdots \\
= \; & \nu^{-1/2}b\left(\frac{m}{\lambda} - \nu t_k + \Sigma_{i=1}^{k}M_{i-1}\Delta t_i\right) - \frac{mb^2}{2\nu\lambda^2} - abt_k - a\nu^{1/2}\log\left(1-\nu^{-1/2}u_{2\nu}\right) \\
& -\frac{a^2}{2}\left(\frac{(1-e^{-\lambda t_k})/e^{-\lambda t_k}+\nu^{-1/2}u_{2\nu}e^{-\lambda t_k}}{1-\nu^{-1/2}u_{2\nu}}\right) + O(\nu^{-1/2}) \\
= \; & bu_{1\nu} - \frac{1}{2}b^2I(\lambda) - abt_k + au_{2\nu} - \frac{a^2}{2}\frac{1-e^{-\lambda t_k}}{e^{-\lambda t_k}} + o_p(1), \qquad (5)
\end{aligned}$$

since,

$$I(\lambda) \;=\; E\left[-\frac{\delta^2}{\delta\lambda^2}\log p_x(x_j,\lambda)\right]$$

$$
\begin{aligned}
&= E\left[-\nu^{-1}\Sigma_{j=1}^{\nu}\frac{\delta^2}{\delta\lambda^2}\log p_x(x_j,\lambda)\right] \\
&= E\left[\frac{m}{\nu\lambda^2}\right]. \qquad (6)
\end{aligned}
$$

Note that,

$$
\begin{aligned}
\Sigma_{j=0}^{r-1}(\nu-j)^{-1} &= \log\nu - \log(\nu-r) + O(\nu^{-1}), \\
\nu\Sigma_{j=0}^{r-1}(\nu-j)^{-2} &= \frac{r}{\nu-r} + O(\nu^{-1}), \\
\nu^{3/2}\Sigma_{j=0}^{r-1}(\nu-j)^{-3} &= O(\nu^{-1/2}).
\end{aligned}
$$

Then, using the argument of Sen and Singer ([11], p207), as in Dewanji et al. ([2]), we differentiate (5) to obtain

$$
\begin{aligned}
\frac{\delta}{\delta a}\left(\log L(\nu+\nu^{1/2}a,\lambda+\nu^{-1/2}b)-\log L(\nu,\lambda)\right) &= u_{2\nu} - a\frac{1-e^{-\lambda t_k}}{e^{-\lambda t_k}} - bt_k = 0 \\
\frac{\delta}{\delta b}\left(\log L(\nu+\nu^{1/2}a,\lambda+\nu^{-1/2}b)-\log L(\nu,\lambda)\right) &= -at_k + u_{1\nu} - bI(\lambda) = 0.
\end{aligned}
$$

Hence, we obtain $(\hat{a_\nu},\hat{b_\nu})$ as the solution of

$$
\begin{aligned}
u_{1\nu} &= \hat{a}t_k + \hat{b}I(\lambda) \\
u_{2\nu} &= \hat{a}\frac{1-e^{-\lambda t_k}}{e^{-\lambda t_k}} + \hat{b}t_k
\end{aligned}
$$

so that

$$
\begin{pmatrix} \hat{b} \\ \hat{a} \end{pmatrix} = \Sigma \begin{pmatrix} u_{1\nu} \\ u_{2\nu} \end{pmatrix},
$$

equivalently

$$
\begin{pmatrix} \nu^{1/2}(\hat{\lambda}-\lambda) \\ \nu^{-1/2}(\hat{\nu}-\nu) \end{pmatrix} = \Sigma \begin{pmatrix} u_{1\nu} \\ u_{2\nu} \end{pmatrix}.
$$

Now, $[u_{1\nu},u_{2\nu}] \xrightarrow{L} N(0,\Sigma^{-1})$. Thus, we have the following result.

Result 1: As $\nu\to\infty$,

$$
[\nu^{1/2}(\hat{\lambda}-\lambda),\nu^{-1/2}(\hat{\nu}-\nu)] \xrightarrow{L} N(0,\Sigma).
$$

The covariance matrix Σ can be consistently estimated by using $\hat{\lambda}$ in place of λ and by estimating $I(\lambda)$ by $-\hat{\nu}^{-1}\partial^2 \log L(\hat{\nu}, \hat{\lambda})/\partial\lambda^2 \; = \frac{m}{\hat{\nu}\hat{\lambda}^2}$.

In particular, the variance of $\hat{\nu}$ can be consistently estimated by

$$\hat{\nu}\left[\frac{1-e^{-\hat{\lambda}t_k}}{e^{-\hat{\lambda}t_k}} - \frac{t_k^2\hat{\nu}\hat{\lambda}^2}{m}\right]^{-1}. \tag{7}$$

5 A Simulation Study

In order to assess the performance of the estimators of ν and λ through simulation, we consider four values of ν, namely, $\nu = 100, 500, 1000$ and 5000 with $t_k = 10$, while time between successive debugging is 1, so that number of debugging k is equal to 10. We consider three choices of the parameter λ as 0.2303, 0.1609 and 0.0916 resulting in $\bar{F}(t_k, \lambda) = 0.1$, 0.2 and 0.4, respectively, reflecting different extent of non-detection (or, missingness). This results in $4 \times 3 = 12$ different parameter configurations. For each such configuration, we simulate 10,000 datasets. For each dataset, excluding the cases when $M_k = m$ and $m \geq 1 + 2B$, we compute $\hat{\nu}$, $\hat{\lambda}$ and the standard error of $\hat{\nu}$ using (7), which is denoted by $s(\hat{\nu})$. As noted in Section 3, if $M_k = m$ and $m \geq 1 + 2B$, then $\hat{\nu}$ is infinite.

Since ν is the parameter of primary interest, we report the simulation results regarding the performance of $\hat{\nu}$ in Table 1. In addition to the average of $\hat{\nu}$ and $s(\hat{\nu})$ over the 10,000 simulations, we also present the sample standard error of $\hat{\nu}$, denoted by $sse(\hat{\nu})$, defined as the sample standard deviation of 10,000 estimates of ν. The estimated coverage probability, denoted by CP, is computed as the proportion of times (out of 10,000 simulations) the asymptotic 95% confidence interval, obtained through the normal approximation of $\hat{\nu}$ (see Result 1) and using (7), contains the true ν. For the sake of comparison, we also present relative bias and relative standard error defined as $(\hat{\nu} - \nu)/\nu$ and $s(\hat{\nu})/\nu$, respectively.

Note that the average standard error under $s(\hat{\nu})$ and the sample standard error under

$sse(\hat{\nu})$ are very close specially for large ν and small $\bar{F}(t_k, \lambda)$. This gives evidence for convergence to the asymptotic variance of $\hat{\nu}$ in (7). As expected, the estimator $\hat{\nu}$ seems to perform better with respect to relative bias, relative standard error and CP with increasing ν and decreasing $\bar{F}(t_k, \lambda)$. The MLE is nearly unbiased in all cases. Also, the CP values are close to 0.95, specially for large ν, suggesting convergence to normality as given in Result 1.

Table 1: Simulation results on the MLE of ν with corresponding standard errors and estimated coverage probability for $t_k = 10$.

ν	$\bar{F}(t_k, \lambda)$	$\hat{\nu}$	$(\hat{\nu} - \nu)/\nu$	$s(\hat{\nu})$	$sse(\hat{\nu})$	$s(\hat{\nu})/\nu$	CP
100	0.1	99.68	-0.0033	5.61	5.26	0.056	0.894
	0.2	99.75	-0.0025	11.09	10.33	0.111	0.889
	0.4	110.19	0.1019	74.98	57.68	0.750	0.864
500	0.1	499.38	-0.0012	11.68	11.56	0.023	0.939
	0.2	499.62	-0.0008	22.60	22.43	0.045	0.939
	0.4	504.89	0.0098	60.68	58.33	0.121	0.932
1000	0.1	999.56	-0.0005	15.95	16.37	0.016	0.953
	0.2	999.51	-0.0005	31.18	31.59	0.031	0.948
	0.4	1003.93	0.0039	82.33	80.09	0.082	0.939
5000	0.1	4999.04	-0.0002	36.88	36.55	0.007	0.945
	0.2	4998.41	-0.0003	71.00	70.36	0.014	0.945
	0.4	5005.08	0.0010	176.38	176.02	0.035	0.949

To study the effect of the number and size of debugging intervals, we have carried out another simulation study with fixed $t_k = 10$, while the time Δ between successive debugging is varied as 1, 2, 5 and 10 with the corresponding number k of debugging being 10, 5, 2 and 1. The probability of non-detection $\bar{F}(t_k, \lambda)$ is kept fixed at 0.2 corresponding to $\lambda = 0.1609$. The same simulation exercise as before is carried out in 10,000 repetitions and the results are presented in Table 2.

Table 2: Simulation results with varying number of debugging and $\bar{F}(t_k, \lambda) = 0.2$ for $t_k = 10$.

ν	Δ	$\hat{\nu}$	$(\hat{\nu} - \nu)/\nu$	$s(\hat{\nu})$	$s(\hat{\nu})/\nu$	CP
100	1	99.75	-0.0025	11.096	0.111	0.889
	2	99.66	-0.0034	9.94	0.099	0.903
	5	100.05	0.0005	7.86	0.079	0.936
	10	99.69	-0.0031	6.53	0.065	0.945
500	1	499.62	-0.0008	22.60	0.045	0.939
	2	500.02	4.08E-05	20.52	0.041	0.944
	5	499.65	-0.0007	16.75	0.034	0.946
	10	499.87	-0.0003	14.52	0.029	0.951
1000	1	999.51	-0.0005	31.18	0.031	0.948
	2	999.48	-0.0005	28.75	0.029	0.945
	5	999.56	-0.0004	23.76	0.024	0.946
	10	999.67	-0.0003	20.25	0.020	0.955
5000	1	4998.41	-0.0003	71.00	0.014	0.945
	2	4999.70	-5.946E-05	63.96	0.013	0.949
	5	4999.16	-0.0002	53.64	0.011	0.946
	10	5000.66	0.0001	45.47	0.009	0.949

The average standard error and the sample standard error turn out to be very close, as before, and so only the $s(\hat{\nu})$'s are reported. The estimator $\hat{\nu}$ seems to perform better with respect to relative bias, relative standard error and CP with increasing ν, as before, and the efficiency in terms of relative standard error seems to increase with decreasing number k of debugging intervals. This is an interesting observation in the sense that better efficiency for smaller k seems to be counter-intuitive. In fact, with decreasing k, there are more replications of failure time from an error resulting in more information and, hence, better efficiency. Therefore, a single debugging schedule at the end of testing at time t_k, as in [10], seems to be the most efficient design. It can also be easily proved that the asymptotic variance of $\hat{\nu}$ for k debugging intervals, given by

$$\nu \left[\frac{1 - e^{-\lambda t_k}}{e^{-\lambda t_k}} - \frac{\lambda t_k^2}{\Sigma_{i=1}^{k} (t_i - t_{i-1}) e^{-\lambda t_{i-1}}} \right]^{-1},$$

is minimum when $k = 1$. However, as remarked in Section 1, a schedule of more than one debugging intervals may be necessary due the market demand for software release.

It is of interest to investigate the robustness of the estimator $\hat{\nu}$, obtained by using exponential failure time distribution as in Section 3, while the true failure time distribution is not exponential. We have carried out an extensive simulation study by generating failure time from Weilbull and Gamma distributions having both increasing and decreasing failure rate and with different probability of non-detection. The results (partially reported in Tables 3 and 4.) indicate sensitiveness of the estimator to violation of the assumption of exponential failure time distribution with the performance being worse for larger probability of non-detection, as expected. Interestingly, it appears from the simulation results that $\hat{\nu}$ over-(under-)estimates ν when the true failure time distribution has increasing(decreasing) failure rate (see [1] for similar result).

Table 3: Empirical evaluation of the estimators for Weibull distribution for failure rate is 0.9 and $t_k = 10$.

ν	$\bar{F}(t_k, \lambda)$	$\hat{\nu}$	$(\hat{\nu} - \nu)/\nu$	$s(\hat{\nu})$	$sse(\hat{\nu})$	$s(\hat{\nu})/\nu$	CP
100	0.1	94.81	-0.0519	4.09	3.25	0.041	0.554
	0.2	91.02	-0.1623	6.96	6.16	0.070	0.5701
	0.4	83.77	-0.162	19.82	15.75	0.198	0.537
500	0.1	477.66	-0.0447	8.88	7.47	0.018	0.224
	0.2	459.19	-0.0817	15.38	13.93	0.031	0.227
	0.4	420.71	-0.1586	34.38	31.87	0.068	0.315
1000	0.1	956.24	-0.0438	12.65	10.596	0.013	0.057
	0.2	919.14	-0.0809	21.78	19.73	0.022	0.049
	0.4	841.47	-0.1586	47.39	44.76	0.047	0.126

Table 4: Empirical evaluation of the estimators for Weibull distribution for failure rate is 1.1 and $t_k = 10$.

ν	$\bar{F}(t_k, \lambda)$	$\hat{\nu}$	$(\hat{\nu} - \nu)/\nu$	$s(\hat{\nu})$	$sse(\hat{\nu})$	$s(\hat{\nu})/\nu$	CP
100	0.1	105.06	0.0506	7.46	7.77	0.075	0.984
	0.2	111.60	0.1160	23.42	17.82	0.234	0.981
	0.4	125.04	0.2504	53.03	55.0	0.530	0.959
500	0.1	530.31	0.06062	16.43	17.61	0.033	0.653
	0.2	558.92	0.1178	35.80	36.31	0.072	0.737
	0.4	639.60	0.2792	124.39	110.41	0.249	0.989
1000	0.1	1062.08	0.0621	23.17	25.00	0.023	0.236
	0.2	1118.23	0.1182	48.54	51.07	0.049	0.300
	0.4	1266.49	0.2665	153.55	148.69	0.154	0.663

6 An Example

For illustration, we consider a dataset obtained from testing of a flight control software analyzed by Dewanji et al. ([3]) using a different kind of reliability modeling without the parameter ν representing the initial number of errors in the software. The testing was done in four phases consisting of 187, 227, 187 and 267 runs with 2, 1, 3 and 0 failures, respectively, without any repetition. Debugging was done at the end of each phase. Treating each simulation run as one unit of time for the sake of illustration, we have a periodic debugging schedule with $k = 4$ and $t_1 = 187$, $t_2 - t_1 = 227$, $t_3 - t_2 = 187$ and $t_4 - t_3 = 267$. We also have $m = 6$, $M_1 = 2$, $M_2 = 3$, $M_3 = M_4 = 6$ and $B = 3.015$.

Since we have $6 = m < 1 + 2B = 7.03$, the profile likelihood method gives MLE of ν as $\hat{\nu} = 9$ with standard error 6.43. Since ν seems to be small, the asymptotic results of Section 4, and hence the standard error in (7), may be in doubt. We, therefore, use a parametric bootstrap method to estimate the sampling distribution of $\hat{\nu}$. Based on 10,000 bootstrap samples, the symmetric 95% confidence interval for ν turns out to be

[3,24].

The quantity of primary interest is, however, the reliability of the software for an additional τ (say) units of time which is given by $e^{-(\nu-M_k)\lambda\tau}$ and is estimated by $e^{-(\hat{\nu}-M_k)\hat{\lambda}\tau}$. For τ =1 and 5, the estimates of this reliability are 0.9965 and 0.9828, respectively. Application of delta method to obtain the standard error of this estimate may be questionable since one of the parameters, ν, takes discrete values and also as ν seems to be rather small. However, treating ν to be continuous and using delta method, corresponding standard errors are 0.0024 and 0.0117. The standard errors are also obtained on the basis of 10,000 parametric bootstrap samples using $\hat{\lambda} = 0.0012$ and $\hat{\nu} = 9$ as 0.0020 and 0.0098, respectively, which are not very different from those obtained by using delta method. Note that, in the analysis of Dewanji et al. ([3]) of the same data, the reliability for an additional unit of time ($\tau = 1$) is estimated as 0.9979 with standard error 0.0026.

Acknowledgments:- The authors are thankful to Professor Debasis Sengupta for many valuable comments.

References

[1] Anup Dewanji, Subrata Kundu, and Tapan K. Nayak. Nonparametric estimation of the number of components of a superposition of renewal processes. *Journal of Statistical Planning and Inference*, 142:27102718, 2012.

[2] Anup Dewanji, Tapan K. Nayak, and Pranab K. Sen. Estimating the number of components of a system of superimposed renewal processes. *Sankhyā: The Indian Journal of Statistics Series A*, pages 486–499, 1995.

[3] Anup Dewanji, Debasis Sengupta, and Ashis Kumar Chakraborty. A discrete time model for software reliability with application to a flight control software. *Applied Stochastic Models in Business and Industry*, 27:723–731, 2011.

[4] W. Feller. *An Introduction to Probability Theory and Its Application*, volume 1. Wiley Eastern Limited, New Delhi, 1968.

[5] Swapna S. Gokhale, Peter N. Marinos, and Kishor S. Trivedi. Important milestones in software reliability modeling. *SEKE*, pages 345–352, 1996.

[6] Z. Jelinski and P. Moranda. Software reliability research. *Statistical Computer Performance Evaluation, Academic Press*, pages 465–484, 1972.

[7] Michael R. Lyu. *Handbook of software reliability engineering.*, volume 3. CA: IEEE computer society press, 1996.

[8] John D. Musa, Anthony Iannino, and Kazuhira Okumoto. *Software Reliabilty.* McGraw-Hill, New York, 1987.

[9] Tapan K. Nayak. Estimating population size by recapture sampling. *Biometrika*, 75:113–120, 1988.

[10] Tapan K. Nayak. A note on estimating the number of errors in a system by recapture sampling. *Statistics & Probability Letters*, 7.3:191–194, 1988.

[11] P. K. Sen and J. M. Singer. *Large sample methods in statistics an introduction with application.* Chapman and Hall, London, 1993.

[12] Nozer D. Singpurwalla and Simon P. Wilson. Software reliability modeling. *International Statistical Review*, 62(3):289–317, December 1994.