www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 12 Dec. 2016, Page No. 19552-19560

On a Software Reliability Growth Model with Log Logistic failure time distribution

Balakrishna, K. Harishchandra
Department of Statistics
Vijaya Evening College, Bengalur, Karnataka, Indian- 560 004
E-mail: sbk8060@rediffmail.com

Department of Statistics
Bangalore University, Bengalur, Karnataka, India-560 056
E-mail: harish.jbc@gmail.com

Abstract

Non Homogeneous Poisson process models with expected number of faults detected in given testing time are proposed in the literature. These models show either constant, monotonic increasing or monotonic decreasing failure occurrence rate per fault. In this article we propose a software reliability model in which the distribution of time between two failures assumed to be log logistic distribution. The model can capture increasing/decreasing nature of failure rate. The parameters of the model are estimated using maximum likelihood method. A simulation study and real data are used to verify the model.

Keywords: *Software reliability, log-logistic distribution, fault detection process, failure occurrence rate per fault, Maximum likelihood estimation.*

1. Introduction

A computer is an electronic device used for quick and accurate function of any phenomena. Today computers have been widely used for control of many complex systems. The efficiency of a computer depends on its two main ingredients namely hardware and the software. The quality of hardware and software of a system can be described by many parameters such as complexity, portability, maintainability, availability, reliability, etc. The Reliability of the computer both in terms of hardware and software is of very much essential for development of computer field. Software Reliability received great attention to deal with statistical problems. Software reliability is defined as *the probability of failure –free operation of a computer program in a specified environment for a specified period of time* (Musa and Okumoto, 1982).

During the past numerous software reliability models have been developed by the researchers to provide useful information about how to improve software reliability. For a detailed review of software reliability models see Lyu (1996) , Pham (2006), Musa et. al. (1987). Goel and Okumoto (1979), Pham (2006) and Zhang and Pham (2000), have discussed some software reliability models with only fault detection processes with the assumption of perfect and immediate fault correction. Yamada, et.al. (1984)

have studied an S-shaped software reliability growth model. Harishchandra and Manjunatha (2010a, 2010b) have discussed the maximum likelihood estimation of the parameters of a software reliability model including fault detection and correction processes. Harishchandra et.al. (2016) have studied a software reliability model assuming that whenever a software error occurs a multiple number of errors are detected and the model includes a correction process as well.

In this article we propose a software reliability growth model in which the time between two failures assumed to be log logistic distribution. The motivation to choose this distribution is the inadequacy of existing models to describe the nature of failure process.

2. Finite Failure NHPP Models

Non Homogeneous Poisson Process group of models provides an analytical framework for describing the software failure process during testing. The main issue of NHPP models is to estimate $m(t)$, the expected number of faults experienced up to a certain point of time, which is also called mean value function. These models differ in mean value function $m(t)$. The NHPP models can be further classified into two categories, namely finite failure and infinite failure categories. Finite failure NHPP models assumes that the expected number of faults experienced detected given infinite amount of testing time will be finite, where as in Infinite failure models assume that the expected number of faults detected given infinite amount of testing time will be infinite. The parameters of the finite failure NHPP model are described as under.

If $N(t)$ be the cumulative number of faults detected by the time t , $F(t) = P[T \leq t]$ is the distribution function and a denote the expected number of faults that would be detected in a given infinite testing time, then the mean value function is given by

$$m(t) = E[N(t)] = aF(t) \tag{1}$$

The failure intensity function is given by

$$\lambda(t) = aF'(t) \tag{2}$$

The reliability function is given by

$$R(t) = P[T \geq t] = 1 - P[T \leq t] = 1 - F(t)$$

The hazard rate or failure occurrence rate per fault of the software is given by

$$h(t) = \frac{F'(t)}{1 - F(t)} \tag{3}$$

The failure intensity function can also expressed in terms of hazard rate as

$$\lambda(t) = aF'(t) = [a - m(t)]\frac{F'(t)}{1 - F(t)} = [a - m(t)]h(t) \tag{4}$$

In the Goel-Okumoto (GO) model failure occurrence rate per fault of the software $h(t)$ is assumed to be time independent that is constant b. In generalized GO model or Weibull model the nature of the failure occurrence rate per fault is determined by the parameter γ. If $\gamma < 1$ then the failure occurrence rate per fault is increasing and if $\gamma > 1$ then the failure occurrence rate per fault is decreasing. S-shaped SRGM proposed by Yamada et.al has an increasing failure occurrence rate per fault. The expressions of $m(t)$, $\lambda(t)$ and $h(t)$ are presented in 0

Table 1.

Coverage function	$m(t)$	$\lambda(t)$	$h(t)$
Goel Okumoto	$a(1-e^{-bt})$	abe^{-bt}	b
Weibull	$a\left(1-e^{-bt^{\gamma}}\right)$	$ab\gamma t^{\gamma-1}e^{-bt^{\gamma}}$	$ab\gamma t^{\gamma-1}$
S-shaped	$a[1-(1+bt)e^{-bt}]$	ab^2te^{-bt}	$\frac{b^2t}{1+bt}$

3. Models with semi infinite interval distribution

In some of the failure data sets, the rate at which individual faults manifest themselves as testing progresses can also exhibit an increasing/decreasing behavior. An increasing/decreasing trend exhibited by the failure occurrence rate per fault cannot be captured by the models.

Harishchandra and Manjunath developed a SRGM in which behavior of failure occurrence rate per fault i.e., hazard function is described by generalized inverse exponential distribution. The hazard rate function $h(t)$ of the generalized inverse exponential SRGM is given as

$$h(t) = \frac{\alpha b}{t^2}\left(\frac{e^{-b/t}}{1-e^{-b/t}}\right) \tag{5}$$

The corresponding mean value function $m(t)$ and failure intensity function $\lambda(t)$ are

$$m(t) = a[1-(1-e^{-bt})^{\alpha}] \tag{6}$$

$$\lambda(t) = \frac{a\alpha b}{t^2}e^{-bt}(1-e^{-bt})^{\alpha-1} \tag{7}$$

The software reliability is the conditional probability that the i^{th} software failure does not occur between $(t, t+x], (x \geq 0)$ on the condition that the $(i-1)th$ software failure has occurred at testing time t, is given by

$$R(x|t) = exp\left\{a\left[\left(1-e^{-b/(t+x)}\right)^{\alpha} - (1-e^{-bt})^{\alpha}\right]\right\} \tag{8}$$

4. Models with log logistic distribution

DOI: 10.18535/ijecs/v5i12.53

In this section we use log logistic distribution to construct a SRG model which capture the increasing/decreasing nature of the hazard function. The increasing/decreasing behavior of the failure occurrence rate per fault can be captured by the hazard of the log logistic distribution.

If t is the time between two failures then the probability density function of log logistic distribution is given by

$$f(t) = \frac{\left(\frac{\beta}{\alpha}\right)\left(\frac{t}{\alpha}\right)^{\beta-1}}{\left[1+\left(\frac{t}{\alpha}\right)^{\beta}\right]^{2}},$$

$$t \geq 0\ \alpha \geq 0\ is\ scale\ parameter\ and\ \beta \geq 0\ is\ shape\ parameter \tag{9}$$

The corresponding distribution function, mean value function, failure intensity function and the hazard function of the software reliability growth model with log logistic distribution are given by.

$$R(x|t) = exp\left\{a\left[\left(1-e^{-b/(t+x)}\right)^{\alpha} - (1-e^{-bt})^{\alpha}\right]\right\} \tag{10}$$

$$F(t) = \frac{1}{1+\left(\frac{t}{\alpha}\right)^{-\beta}}, \tag{11}$$

$$m(t) = \frac{a\left(\frac{t}{\alpha}\right)^{\beta}}{1+\left(\frac{t}{\alpha}\right)^{\beta}}, \tag{12}$$

$$\lambda(t) = \frac{a\left(\frac{\beta}{\alpha}\right)\left(\frac{t}{\alpha}\right)^{\beta-1}}{\left[1+\left(\frac{t}{\alpha}\right)^{\beta}\right]^{2}} \tag{13}$$

$$and\quad h(t) = \frac{\left(\frac{\beta}{\alpha}\right)\left(\frac{t}{\alpha}\right)^{\beta-1}}{1+\left(\frac{t}{\alpha}\right)^{\beta}} \tag{14}$$

4.1. Software Reliability

Let X_i be the time between $(i-1)^{th}$ and i^{th} failure. Then $T_i = \sum_{j=1}^{i} X_j, i = 1,2,\cdots$ represent time of occurrence of i^{th} failure. The conditional probability that the i^{th} software failure does not occur between $(t, t+x], (x \geq 0)$ on the condition that $(i-1)^{th}$ software failure has occurred at testing time t, is given by

$$R(x|t) = Pr\{X_i > x|T_{i-1} = t\} = exp\{-m(t+x) - m(t)\} \tag{15}$$

On substituting the (12) in (15)

$$R(x|t) = exp\left\{-\left\{\frac{a\left(\frac{t+x}{\alpha}\right)^{\beta}}{1+\left(\frac{t+x}{\alpha}\right)^{\beta}} - \frac{a\left(\frac{t}{\alpha}\right)^{\beta}}{1+\left(\frac{t}{\alpha}\right)^{\beta}}\right\}\right\} \tag{16}$$

To predict the future reliability, we have to estimate the parameters involved in the above expression. We use Maximum likelihood method to estimate the parameters of the model in the coming section.

4.2. Estimation of parameters.

We obtain the maximum likelihood estimators of the parameters using two types of data namely interval domain data and time domain data.

4.2.1.Estimation of parameters using Interval domain data.

Let y_i , $i = 1,2,\dots,n$ be the observed number of software faults detected upto the testing time t_i, $(0 = t_0 < t_1 < \dots < t_n)$. Then the likelihood function for the interval domain data is

$$L = \prod_{i=1}^{n} P[N(t_i) = y_i] = \prod_{i=1}^{n} \frac{e^{-[m(t_i)-m(t_{i-1})]}[m(t_i) - m(t_{i-1})]^{(y_i - y_{i-1})}}{(y_i - y_{i-1})!} \tag{17}$$

Where $t_0 = 0$ and $n_0 = 0$ Taking natural logarithm for (17) we get log likelihood function

$$lnL = \sum_{i=1}^{n}(n_i - n_{i-1})ln[m(t_i) - m(t_{i-1})] - m(t_n) \tag{18}$$

Substituting the mean value function (12) in (18)

$$lnL = \sum_{i=1}^{n}(y_i - y_{i-1})ln\left[\frac{a\left(\frac{t_i}{\alpha}\right)^{\beta}}{1+\left(\frac{t_i}{\alpha}\right)^{\beta}} - \frac{a\left(\frac{t_{i-1}}{\alpha}\right)^{\beta}}{1+\left(\frac{t_{i-1}}{\alpha}\right)^{\beta}}\right] - \frac{a\left(\frac{t_n}{\alpha}\right)^{\beta}}{1+\left(\frac{t_n}{\alpha}\right)^{\beta}} \tag{19}$$

This can also be presented as,

$$lnL = lna\sum_{i=1}^{n}(y_i - y_{i-1}) + \sum_{i=1}^{n}(y_i - y_{i-1})ln\left[\frac{{t_i}^{\beta}}{\alpha^{\beta} + {t_i}^{\beta}} - \frac{{t_{i-1}}^{\beta}}{\alpha^{\beta} + {t_{i-1}}^{\beta}}\right] - \frac{a{t_n}^{\beta}}{\alpha^{\beta} + {t_n}^{\beta}} \tag{20}$$

Taking partial derivative of (25) with respect to a, α and β and equating them to zero we get the system of equations (21) - (23)

$$a = \frac{\left(\alpha^{\beta} + {t_n}^{\beta}\right)\sum_{i=1}^{n}(y_i - y_{i-1})}{{t_n}^{\beta}} \qquad (21) \tag{21}$$

$$\frac{at_n{}^{\beta}}{\left(\alpha^{\beta}+t_n{}^{\beta}\right)^2}=\sum_{i=1}^{n}\left[\frac{(y_i-y_{i-1})\left[\left(\alpha^{\beta}+t_{i-1}{}^{\beta}\right)^2 t_i{}^{\beta}-\left(\alpha^{\beta}+t_i{}^{\beta}\right)^2 t_{i-1}{}^{\beta}\right]}{\left(\alpha^{\beta}+t_i{}^{\beta}\right)\left(\alpha^{\beta}+t_{i-1}{}^{\beta}\right)\left[\left(\alpha^{\beta}+t_{i-1}{}^{\beta}\right)t_i{}^{\beta}-\left(\alpha^{\beta}+t_i{}^{\beta}\right)t_{i-1}{}^{\beta}\right]}\right]$$

(22) (22)

$$\frac{at_n{}^{\beta}(lnt_n-ln\alpha)}{\left(\alpha^{\beta}+t_n{}^{\beta}\right)^2}=\sum_{i=1}^{n}\left[\frac{(y_i-y_{i-1})\left[\left(\alpha^{\beta}+t_{i-1}{}^{\beta}\right)^2(lnt_i-ln\alpha)t_i{}^{\beta}-\left(\alpha^{\beta}+t_i{}^{\beta}\right)^2(lnt_{i-1}-ln\alpha)t_{i-1}{}^{\beta}\right]}{\left(\alpha^{\beta}+t_i{}^{\beta}\right)\left(\alpha^{\beta}+t_{i-1}{}^{\beta}\right)\left[\left(\alpha^{\beta}+t_{i-1}{}^{\beta}\right)t_i{}^{\beta}-\left(\alpha^{\beta}+t_i{}^{\beta}\right)t_{i-1}{}^{\beta}\right]}\right]$$

(23) (23)

The equations (21) - (23) does not provide closed form expression for the maximum likelihood estimation $a, \alpha\ and\ \beta$. However we can use the numerical method to obtain MLEs of $a, \alpha\ and\ \beta$. If $\hat{a}, \hat{\alpha}\ and\ \hat{\beta}$ are the MLEs of $a, \alpha\ and\ \beta$, Then we can obtain MLEs of mean value function $m(t)$ and failure intensity function $\lambda(t)$ by substituting $\hat{a}, \hat{\alpha}\ and\ \hat{\beta}$.

4.2.2. Estimation of parameters using Time domain data.

Let $s_i, i=1,2,\dots,n; 0=s_0<s_1<\cdots<s_n)$ be the observed time of occurrence of i^{th} failure of the software.. Then the likelihood function for the time domain data is

$$L=\prod_{i=1}^{n} f(s_i)=\prod_{i=1}^{n}\lambda(s_i)\,e^{-\int_{s_{i-1}}^{s_i}\lambda(t)dx}=e^{-m(s_n)}\prod_{i=1}^{n}\lambda(s_i) \tag{24}$$

Where $s_0=0$ taking natural logarithm for (24) we get log likelihood function

$$lnL=-m(s_n)+\sum_{i=1}^{n} ln\lambda(s_i) \tag{25}$$

Substituting the (12) and (13) in (25)

$$lnL=-\frac{a\left(\frac{s_n}{\alpha}\right)^{\beta}}{1+\left(\frac{s_n}{\alpha}\right)^{\beta}}+\sum_{i=1}^{n} ln\left[\frac{a\left(\frac{\beta}{\alpha}\right)\left(\frac{s_i}{\alpha}\right)^{\beta-1}}{\left[1+\left(\frac{s_i}{\alpha}\right)^{\beta}\right]^2}\right] \tag{26}$$

This can also be presented as,

$$lnL=-\frac{as_n{}^{\beta}}{\alpha^{\beta}+s_n{}^{\beta}}+nlna+nln\beta-nln\alpha-n(\beta-1)ln\alpha+2n\beta ln\alpha-2\sum_{i=1}^{n} ln\left(\alpha^{\beta}+s_i{}^{\beta}\right)+(\beta-1)\sum_{i=1}^{n} lns_i \tag{27}$$

Taking partial derivative for (27) with respect to $a, \alpha\ and\ \beta$ and equating them to zero we get the system of equations (28)-(30)

$$a = \frac{n\left(\alpha^{\beta} + s_n^{\beta}\right)}{s_n^{\beta}} \tag{28}$$

$$\frac{a\alpha^{\beta} s_n^{\beta}}{[\alpha^{\beta} + s_n^{\beta}]^2} + n = 2\sum_{i=1}^{n} \frac{\alpha^{\beta}}{\left(\alpha^{\beta}+s_i^{\beta}\right)} \tag{29}$$

$$\frac{a\alpha^{\beta} s_n^{\beta}(lns_n - ln\alpha)}{[\alpha^{\beta} + s_n^{\beta}]^2} + 2\sum_{i=1}^{n} \frac{\left(\alpha^{\beta} lns_i - s_i^{\beta} ln\alpha\right)}{\left(\alpha^{\beta}+s_i^{\beta}\right)} = nln\alpha + \frac{n}{\beta} + \sum_{i=1}^{n} ln\, s_i \tag{30}$$

Equations (28)-(30) does not gives closed form expression for the maximum likelihood estimation $a, \alpha\ and\ \beta$. However we can use the numerical method to obtain MLEs of $a, \alpha\ and\ \beta$. If $\hat{a}, \hat{\alpha}\ and\ \hat{\beta}$ are the MLEs of $a, \alpha\ and\ \beta$, Then we can obtain MLEs of mean value function $m(t)$ and failure intensity function $\lambda(t)$ by substituting $\hat{a}, \hat{\alpha}\ and\ \hat{\beta}$.

5. Numerical Analysis

We now present a numerical example for finite failure Software Reliability Growth models based on the actual testing-data. The data set was extracted from information about failures in the development of software for real-time multi computer complex of the US Naval Fleet Computer Center of the US Naval Tactical Data System (NTDS) (Goel 1979a). The data consists of 26 software failure occurrence time $\{ s_i(days); i = 1,2, \ldots ,26\}$ out of 250 days. Here we obtain the estimation results and AIC value for existing and our proposed models based on log logistic distribution. The output are summarized in the table 2 given below:

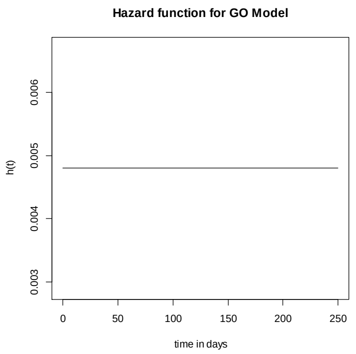
Table 2.

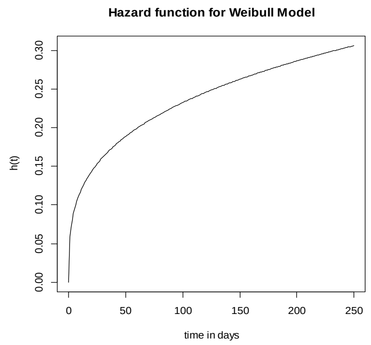
Model	Parameters			AIC
GO	$\hat{a} = 34.9999$	$\hat{b} = 0.0048$		170.0127
Weibull	$\hat{a} = 29.9999$	$\hat{b} = 0.0015$	$\hat{\gamma} = 1.3000$	171.0003
S-Shaped	$\hat{a} = 29.9999$	$\hat{b} = 0.0165$		169.0966
GIED	$\hat{a} = 320.4638$	$\hat{\alpha} = 42.2599$	$\hat{b} = 0.0415$	169.4721
Log logistic	$\hat{a} = 29.6577$	$\hat{\alpha} = 104.852$	$\hat{\beta} = 1.7297$	169.0286

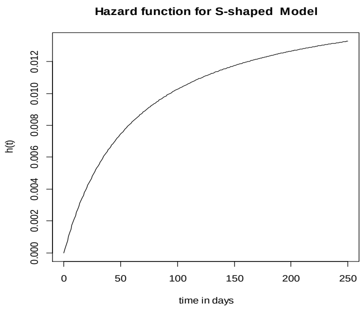
From the above table, we observe that our proposed model with log logistic performs better compared to other models.

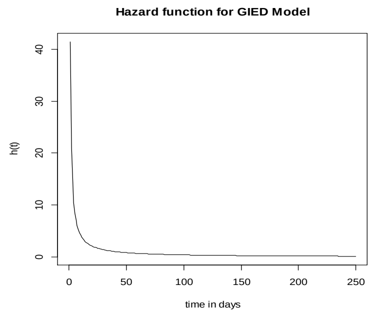
Using the above estimates of the parameters, the graph of hazard function of the existing and our proposed model based on log logistic distribution are plotted against the time (in days). The graphs are shown Figure 1

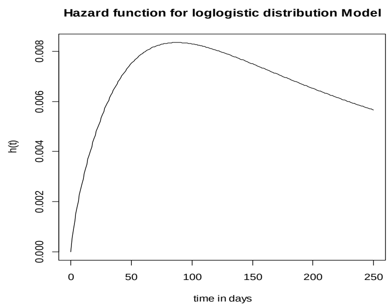
Figure 1.: Hazard function for existing and proposed model.











6. Conclusions

In this paper, we have proposed finite failure SRGM based on log logistic distribution because of the fact that the existing finite failure NHPP models were inadequate to describe failure pattern. We use the mean value function $m(t)$ and failure intensity function $\lambda(t)$ to determine the likelihood function for both interval domain data and time domain data. The estimates of the parameters $a, \alpha \; and \; \beta$ are obtained by the method of maximum likelihood. A numerical example is illustrated for model comparison using AIC. The results shows that the proposed model of SRGM based on log logistic distribution performs better compared to previous finite category models. We also presented the graph of hazard function for various finite failure software reliability growth models. Among various graphs plotted the graph of the hazard rate corresponding to the log logistic distribution is shows increasing/decreasing behavior of the failure occurrence rate per fault.

DOI: 10.18535/ijecs/v5i12.53

References

1. Goel, A. L., "*Software Reliability Models: Assumptions, Limitations and Applicability", *IEEE Trans. Software Engineering; SE-11, pp. 1411-1423, 1985.
2. Lyu. M. R., '*Handbook of Software Reliability Engineering*'. McGraw Hill, 1996.
3. Goel, A. L., and Okumoto. K,. '*Time-dependent error detection rate model for software reliability and other performance, Measure',* IEEE Trans. Reliability R-28(3), pp 206-211, 1979
4. Harishchandra K, Manjunatha K M. *Statistical inference on software reliability assuming exponential fault correction time*. *Economic Quality Control* , **25**; 269-279. 2010a.
5. Harishchandra, K. and Manjunatha, K. M.: *Maximum Likelihood and Bayesian Estimation in a Software Reliability Model with Exponential Fault Correction Time*, Advances and Applications in Statistics, Volume 18, Number 2, pp. 175-188. 2010b.
6. Harishchandra. K, Manjunatha' K.M. and Balakrishna ., *"Software Reliability Model when multiple errors occur at a time including a fault correction process*", Int. Journal of Engineering and Computer Sciences, Vol. 5, No. 12, 2016.
7. Musa J. D, Iannino A, and Okumoto K., '*Software Reliability: Measurement, Prediction, Application'*, McGraw-Hill, Inc., USA, 1987.
8. Zhang X, Pham H., *'Comparisons of non homogeneous Poisson process software reliability models and its applications',* International Journal of Systems Science, Vol. 31, No. 9, 1115- 1123, 2000.
9. Yamada. S, Ohba. M and Osoki. S., '*S shaped reliability growth modeling for software error detection',* IEEE Trans. Reliability R-32(5), pp. 475-484, 1983.