



Burr Type III Process Model with SPRT for Software Reliability

CH.Smitha¹, Dr.R.Satya Prasad², Dr.R.Kiran Kumar³

1. Research Scholar, Dept. of Computer Science, Krishna University, Andhra Pradesh. INDIA-521001

2. Associate Professor, CSE, AcharyaNagarjuna University, Andhra Pradesh, INDIA-522502

3. Asst.Professor & Head of Dept. of Computer Science, Krishna University, Andhra Pradesh. INDIA-521001

Abstract: Software systems have become integral part of everyday life and dependency on these makes the assessment of their reliability, a crucial task in software development. To facilitate the assessment of software reliability, effective tools and mechanisms are required. Classical approaches such as hypothesis testing are significantly time consuming as the conclusion can only be drawn after collecting huge amounts of data. Statistical methods like Sequential Analysis can be applied to arrive at a decision quickly. We propose to implement Sequential Probability Ratio Test (SPRT) for Burr Type III model based on time domain data. For this, parameters are estimated using Maximum Likelihood Estimation to apply SPRT on real time software failure datasets borrowed from different software projects.

Keywords: Burr Type III Model, ML Estimation, Software Reliability, Sequential Probability Ratio Test, Time Domain Data.

INTRODUCTION

Software reliability is probability of fault free operations provided by the software product under consideration over a specified period of time in a specified operational environment [1]. Assessment of software reliability needs effective tools and mechanisms. In classical Hypothesis Testing, the entire data has to be collected first, later the analysis is done and conclusions are drawn based on the data collected. The application of software reliability growth models may be difficult and reliability predictions can be misleading when classical testing strategies are used (no usage testing) whereas certain methods like statistical methods can be successfully applied to the failure data [2]. Sequential analysis is a method of statistical inference and here number of observations required by the procedure is not determined in advance of the experiment. The termination decision of the experiment depends, at each stage, on the results of the observation previously made. A merit of sequential method, as applied to testing statistically a hypothesis, is that a test procedure can be constructed which requires on average a small number of observations that equally test the reliability of the procedure based on a predetermined number of observations[3] [4]. Stieber's observations are demonstrated by applying the well-known Sequential Probability Ratio Test (SPRT) of Wald [5] for a software failure data to detect unreliable software components and compare the reliability of different software versions.

Software failure data is needed for Software reliability analysis. The two types of failure data that exist are time-domain data and interval-domain data. The time-domain data records failures that occur at individual times. The interval-domain data records count of number of failures occurring during a fixed time period. With existing software reliability models, time-domain data provides better accuracy in the estimation of parameters, but involves more data collection efforts [6]. The probability equation of the stochastic process representing the failure occurrences is given by a homogeneous Poisson process with the expression.

$$P[N(t) = n] = \frac{[\lambda t]^n}{n!} e^{-\lambda t} \quad 1.1$$

This paper describes a method for detecting reliable software based on the SPRT, using Maximum Likelihood Estimation (MLE) of parameter estimation. The Wald's SPRT procedure can be used to distinguish the software under test into one of the two categories like reliable/unreliable, pass/fail and certified/uncertified [7]. SPRT is the optimal statistical test that makes the correct decision in the shortest time among all tests that are subject to the same level of decision errors [8]. SPRT is used to detect the fault based on the calculated likelihood of the hypotheses.

We consider one of the popular software reliability growth model Burr Type III and adopted the principle of Stieber [2] in detecting whether the software is reliable or unreliable in order to accept or reject the developed software. The theory proposed by Stieber is described in section 2. Implementation of SPRT for the proposed Burr type III Software Reliability Growth Model is illustrated in section 3. Maximum Likelihood estimation method is used to estimate the parameters and presented in Section 4.

II. WALD'S SEQUENTIAL TEST FOR A POISSON PROCESS

The Sequential Probability Ratio Test (SPRT) was developed by Abraham Wald at Columbia University in 1943[5]. During the manufacturing of software products the SPRT procedure for quality control studies is used. Fixed sample size sets with fewer observations can be considered to perform tests. The SPRT methodology for Homogeneous Poisson Process is described below.

Let $\{N(t), t \geq 0\}$ be a homogeneous Poisson process with rate ' λ '. In this case, $N(t)$ = number of failures up to time 't' and ' λ ' is the failure rate (failures per unit time). If the system is put on test and that if we want to estimate its failure rate ' λ '. We cannot expect to estimate ' λ ' precisely. But we want to reject the system with a high probability if the data suggest that the failure rate is larger than λ_1 and accept it with a high probability, if it is smaller than λ_0 . Here we have to specify two (small) numbers ' α ' and ' β ', where ' α ' is the probability of falsely rejecting the system. That is rejecting the system even if $\lambda \leq \lambda_0$. This is the "producer's" risk. ' β ' is the probability of falsely accepting the system. That is accepting the system even if $\lambda \geq \lambda_1$. This is the "consumer's" risk. Wald's classical SPRT is very sensitive to the choice of relative risk required in the specification of the alternative hypothesis. With the classical SPRT, tests are performed continuously at every time point, as $t > 0$ additional data are collected. With specified choices of λ_0 and λ_1 such that $0 < \lambda_0 < \lambda_1$, the probability of finding $N(t)$ failures in the time span $(0, t)$ with λ_1, λ_0 as the failure rates are respectively given by

$$P_1 = \frac{e^{-\lambda_1 t} [\lambda_1 t]^{N(t)}}{N(t)!} \tag{2.1}$$

$$P_0 = \frac{e^{-\lambda_0 t} [\lambda_0 t]^{N(t)}}{N(t)!} \tag{2.2}$$

The ratio $\frac{P_1}{P_0}$ at any time 't' is considered as a measure of deciding the truth towards λ_0 or λ_1 , given a sequence of time instants say $t_1 < t_2 < t_3 < \dots < t_k$ and the corresponding realizations $N(t_1), N(t_2), \dots, N(t_k)$ of $N(t)$.

Simplification of $\frac{P_1}{P_0}$ gives

$$\frac{P_1}{P_0} = \exp(\lambda_0 - \lambda_1)t + \left(\frac{\lambda_1}{\lambda_0}\right)^{N(t)}$$

The decision rule of SPRT is to decide in favor of λ_1 , in favor of λ_0 or to continue by observing the number of failures at a later time than 't' according as $\frac{P_1}{P_0}$ is greater than or equal to a constant say A, less than or equal to a constant say B or in between the constants A and B. That is, we decide the given software product as unreliable, reliable or continue [9] the test process with one more observation in failure data, according to

$$\frac{P_1}{P_0} \geq A \tag{2.3}$$

$$\frac{P_1}{P_0} \leq B \tag{2.4}$$

$$B < \frac{P_1}{P_0} < A \tag{2.5}$$

The approximate values of the constants A and B are taken as

$$A \cong \frac{1 - \beta}{\alpha}, \quad B \cong \frac{\beta}{1 - \alpha}$$

Where ‘ α ’ and ‘ β ’ are the risk probabilities as defined earlier. A simplified version of the above decision processes is
 To reject the system as unreliable if $N(t)$ falls for the first time above the line

$$N_U(t) = at + b_2 \quad (2.6)$$

To accept the system to be reliable if $N(t)$ falls for the first time below the line

$$N_L(t) = at - b_1 \quad (2.7)$$

To continue the test with one more observation on $(t, N(t))$ as the random graph of $[t, N(t)]$ is between the two linear boundaries given by equations (2.6) and (2.7) where

$$a = \frac{\lambda_1 - \lambda_0}{\log\left(\frac{\lambda_1}{\lambda_0}\right)} \quad (2.8)$$

$$b_1 = \frac{\log\left[\frac{1-\alpha}{\beta}\right]}{\log\left(\frac{\lambda_1}{\lambda_0}\right)} \quad (2.9)$$

$$b_2 = \frac{\log\left[\frac{1-\beta}{\alpha}\right]}{\log\left(\frac{\lambda_1}{\lambda_0}\right)} \quad (2.10)$$

The parameters α, β, λ_0 and λ_1 can be chosen in several ways. One way suggested by Stieber is

$$\lambda_0 = \frac{\lambda \cdot \log(q)}{q-1}, \quad \lambda_1 = q \cdot \frac{\lambda \cdot \log(q)}{q-1} \quad \text{where } q = \frac{\lambda_1}{\lambda_0}$$

If λ_0 and λ_1 are chosen in this way, the slope of $N_U(t)$ and $N_L(t)$ equals λ . The other two ways of choosing λ_0 and λ_1 are from past projects (for a comparison of the projects) and from part of the data to compare the reliability of different functional areas (components).

III. SEQUENTIAL TEST FOR SOFTWARE RELIABILITY GROWTH MODELS

We know that for any Poisson process, the expected value of $N(t) = \lambda(t)$ called the average number of failures experienced in time ‘t’. Which is also called the mean value function of the Poisson process. On the other hand if we consider a Poisson process with a general function (not necessarily linear) $m(t)$ as its mean value function the probability equation of a such a process is

$$P[N(t) = Y] = \frac{[m(t)]^y}{y!} \cdot e^{-m(t)}, \quad y = 0, 1, 2, \dots$$

Depending on the forms of $m(t)$ we get various Poisson processes called NHPP, for the Burr Type III model. The mean value function is given as

$$m(t) = a[1 + t^{-c}]^{-b}$$

It can also be written as

$$P_1 = \frac{e^{-m_1(t)} \cdot [m_1(t)]^{N(t)}}{N(t)!}$$

$$P_0 = \frac{e^{-m_0(t)} \cdot [m_0(t)]^{N(t)}}{N(t)!}$$

Here $m_1(t), m_0(t)$ represents the mean value function for the stated parameters by indicating reliable software and unreliable software respectively. Over here the mean value function $m(t)$ comprises the parameters ‘a’, ‘b’ and ‘c’ and two specifications of NHPP for b are considered as b_0, b_1 where $(b_0 < b_1)$ and two specifications of c say c_0, c_1 where $(c_0 < c_1)$.

In our proposed model, $m(t)$ at b_1 is said to be greater than b_0 and $m(t)$ at c_1 is said to be greater than c_0 . The same can be denoted symbolically as $m_0(t) < m_1(t)$. The implementation of SPRT procedure is illustrated below.

The system is said to be reliable and can be accepted if

$$\frac{P_1}{P_0} \leq B$$

i.e.,
$$\frac{e^{-m_1(t)} \cdot [m_1(t)]^{N(t)}}{e^{-m_0(t)} \cdot [m_0(t)]^{N(t)}} \leq B$$

i.e.,
$$N(t) \leq \frac{\log\left(\frac{\beta}{1-\alpha}\right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} \quad (3.1)$$

The system is said to be unreliable and rejected if

$$\frac{P_1}{P_0} \geq A \text{ i.e., } N(t) \geq \frac{\log\left(\frac{1-\beta}{\alpha}\right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} \quad (3.2)$$

And continue the test procedure as long as

$$\frac{\log\left(\frac{\beta}{1-\alpha}\right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} < N(t) < \frac{\log\left(\frac{1-\beta}{\alpha}\right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} \quad (3.3)$$

By substituting the appropriate expressions of the respective mean value function, we get the respective decision rules and they are given in followings lines.

Acceptance Region

$$N(t) \leq \frac{\log\left(\frac{\beta}{1-\alpha}\right) + a\left((1+t^{-c_1})^{-b_1} - (1+t^{-c_0})^{-b_0}\right)}{\log\left[\frac{(1+t^{-c_1})^{-b_1}}{(1+t^{-c_0})^{-b_0}}\right]} \quad (3.4)$$

Rejection Region:

$$N(t) \geq \frac{\log\left(\frac{1-\beta}{\alpha}\right) + a\left((1+t^{-c_1})^{-b_1} - (1+t^{-c_0})^{-b_0}\right)}{\log\left[\frac{(1+t^{-c_1})^{-b_1}}{(1+t^{-c_0})^{-b_0}}\right]} \quad (3.5)$$

Continuation Region:

$$\frac{\log\left(\frac{\beta}{1-\alpha}\right) + a\left((1+t^{-c_1})^{-b_1} - (1+t^{-c_0})^{-b_0}\right)}{\log\left[\frac{(1+t^{-c_1})^{-b_1}}{(1+t^{-c_0})^{-b_0}}\right]} < N(t) < \frac{\log\left(\frac{1-\beta}{\alpha}\right) + a\left((1+t^{-c_1})^{-b_1} - (1+t^{-c_0})^{-b_0}\right)}{\log\left[\frac{(1+t^{-c_1})^{-b_1}}{(1+t^{-c_0})^{-b_0}}\right]} \quad (3.6)$$

As observed for the specified model, the decision rules are exclusively based on the strength of the sequential procedure (α , β) and the value of the mean value functions namely $m_0(t)$ $m_1(t)$. As described by Stieber, these decision rules become decision lines if the mean value function is linear in passing through origin, that is $m(t) = \lambda t$. The equations (3.1) and (3.2) are considered as generalizations for the decision procedure of Stieber. SPRT procedure can be applied on live software failure data sets and the results can be analyzed.

IV. PARAMETER ESTIMATION

Parameter estimation plays a significant role in software reliability prediction. Once the analytical solution form is known for a given model, parameter estimation can be achieved by applying a well-known estimation called Maximum Likelihood Estimation (MLE). The main idea behind Maximum Likelihood parameter assessment is to decide the parameters that maximize the probability (likelihood) of the specimen data to assess reliability. In the other words, MLE methods are versatile and applicable to most models and for different types of data. Here parameters are estimated from the time domain data [10]. We present expressions for the parameter estimates of the Burr type III model.

The mean value function of Burr type III model is given by

$$m(t) = a[1 + t^{-c}]^{-b} \quad t > 0, a, b, c > 0 \quad (4.1)$$

The parameters a, b, c are estimated with Maximum Likelihood (ML) estimation.

The likelihood function for time domain data is given by

$$LLF = \sum_{i=1}^n \log[\lambda(t_i)] - m(t_n) \quad (4.2)$$

Substituting Equation (4.1) in equation (4.2) we get

$$\begin{aligned} \text{Log}L &= \sum_{i=1}^n \log \left[\frac{abc}{t_i^{c+1} (1 + t_i^{-c})^{b+1}} \right] - \frac{a}{[1 + t_n^{-c}]^b} \\ \text{Log}L &= \frac{-a}{(1 + t_n^{-c})^b} + \sum_{i=1}^n [\log a + \log b + \log c - (c+1)\log t_i - (b+1)\log(1 + t_i^{-c})] \quad (4.3) \end{aligned}$$

Taking the Partial derivative with respect to 'a' and equating to '0'.

$$\begin{aligned} \frac{\partial \text{Log}L}{\partial a} &= 0 \\ \Rightarrow a &= n(1 + t_n^{-c})^b \quad (4.4) \end{aligned}$$

Taking the Partial derivative with respect to 'b' and equating to '0'.

$$\begin{aligned} \frac{\partial \text{Log}L}{\partial b} &= 0 \\ \Rightarrow b &= \frac{n}{\sum_{i=1}^n \log(1 + t_i^{-1}) - n \log(1 + t_n^{-1})} \quad (4.5) \end{aligned}$$

The parameter 'c' is estimated by iterative Newton-Raphson Method using $c_{i+1} = c_i - \frac{g(c_i)}{g'(c_i)}$ where g(c) and g'(c) are expressed as follows.

$$\begin{aligned} \frac{\partial \text{Log}L}{\partial c} &= 0 \\ \Rightarrow g(c) &= \frac{-n \log(t_n)}{1 + t_n^c} + \frac{n}{c} + \sum_{i=1}^n \log t_i \left[-1 + \frac{2}{1 + t_i^c} \right] \quad (4.6) \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 \text{Log}L}{\partial c^2} &= 0 \\ \Rightarrow g'(c) &= \frac{n(\log t_n)^2 t_n^c}{(t_n^c + 1)^2} - \frac{n}{c^2} - \sum_{i=1}^n \frac{2t_i^c (\log t_i)^2}{(t_i^c + 1)^2} \quad (4.7) \end{aligned}$$

V. CONCLUSION

Assessment of software systems reliability is a crucial task in software development due to its increased dependence and effective tools and mechanisms are required to facilitate the assessment of software reliability. This can be achieved through SPRT. The SPRT methodology for the proposed software reliability growth model Burr type III can be applied for the software failure data sets. Through this we can come to a conclusion in less time regarding the reliability or unreliability of a software product.

REFERENCES

- [1] Marinos, Swapna S. Gokhale Peter N., and Kishor S. Trivedi. "Important Milestones in Software Reliability Modeling." In Proceedings of Software Engineering and Knowledge Engineering (SEKE' 96), Lake Tahoe, NV, pp.345-352.1996.
- [2] STIEBER, H.A.(1997). "Statistical Quality Control: How To Detect Unreliable Software Components", Proceedings the 8th International Symposium on Software Reliability Engineering, 8-12.
- [3] Card D., (1994), "Statistical Process Control for Software", IEEE Software, May, 95-97.
- [4] John D.Musa; "Software Quality and Reliability Basics"; AT&T Bell Laboratories. CH 2468-7/87/0000/014,1987 IEEE.
- [5] Wald A. Sequential Analysis. New Impression edition. New York: John Wiley and Son, Inc; 1947 Sep 30.
- [6] K.B.Misra."Handbook of Performability Engineering". Springer. 2008.
- [7] Reckase MD. A Procedure for decision making using tailored testing. In: Weiss DJ, editor. New horizons in testing: Latent trait theory and computerized adaptive testing. New York: Academic Press; 1983. p. 237-54.
- [8] V.Goutham, R.Satya Prasad "An SPRT Procedure for an Ungrouped Data using MMLE Approach", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727, Volume 14, Issue 6 (Sep. - Oct. 2013), PP 37-42.
- [9] Satya Prasad R. Half Logistic Software Reliability Growth Model [PhD Thesis]. India: ANU; 2007.
- [10] Ch.Smitha Chowdary, Dr R.Satya Prasad, K.Sobhana, "Burr Type III Software Reliability Growth Model," IOSR-JCE Volume17, Issue 1, Jan-Feb 2015.

BIBLIOGRAPHY OF AUTHORS



Mrs. Ch. Smitha Chowdary received MCA from Kakatiya University in 2003 and M.Tech., (Computer Science & Engineering) from Acharya Nagarjuna University in 2010. Now she is pursuing Ph.D., in Computer Science & Engineering from Krishna University as Part-Time Research Scholar under the guidance of Dr. R.SatyaPrasad and co guide Dr.R.Kiran Kumar. Her research interest lies in Software Reliability Engineering, Data Warehousing and Data Mining



Dr. R. Satya Prasad received Ph.D. degree in Computer Science in the Faculty of Engineering in 2007 from Acharya Nagarjuna University, Andhra Pradesh, India. He received gold medal from Acharya Nagarjuna University for his outstanding performance in master's degree. He is currently working as Associate Professor in the department of Computer Science & Engineering, Acharya Nagarjuna University. He performed various academic roles like practical examiner, project adjudicator, external member of board of examiners for various universities and colleges in and around Andhra Pradesh. He received Dr. Abdul Kalam Life Time Achievement Award for his remarkable achievements in the field of Teaching, Research and Publications. His current research is focused on Software engineering, Image processing & Database Management system. He has published 70 research papers in National & International Journals



Dr. K. Kirankumar is an Assistant professor in Department of Computer Science, Krishna University. He is having 12+ experiences in Teaching and college Administration. He guided and developed many tools for smooth administration of the college. He is having more than 25 publications in different International journals. His areas of interest are bioinformatics, data mining, software engineering, embedded systems, and network security. He is guiding the PhD scholars, who are doing research from reputed universities.