

Detection of Burr Type XII Reliable Software using Sequential Process Ratio Test

K. V. Murali Mohan¹, R. Satya Prasad² and G. Sridevi^{3*}

¹ECE, H. M. Institute of Technology and Science, Hyderabad - 501301, Andhra Pradesh, India; kvmuralimohan22@gmail.com

²CSE, Acharya Nagarjuna University, Guntur - 522 510, Andhra Pradesh, India; profrsp@gmail.com

³CSE, KL University, Guntur - 522 502, Andhra Pradesh, India; sridevi.gutta2012@gmail.com

Abstract

Background/Objectives: There is a need to have tools/mechanisms to detect whether a software is reliable or not. Much time is spent while using Classical Hypothesis testing because the conclusions can be drawn only after collecting large amounts of data. **Methods/Statistical Analysis:** By adopting Sequential Analysis of Statistical science, such decision can be made quickly. In this paper, we proposed a new type of statistical science procedure, Sequential Probability Ratio Test (SPRT) applied for Burr Type XII model based on Time domain data. **Findings:** For the proposed Burr type XII model, we applied the SPRT methodology on five real time software failure datasets that were borrowed from different software services. The result exemplifies that the model has given a decision of rejection for all the datasets. Therefore our findings state that all used datasets are unreliable. **Application/Improvements:** Applying SPRT procedure on datasets, we can come to an early conclusion of reliable/unreliable software.

Keywords: Burr Type XII Model, ML Estimation, NHPP, Sequential Probability Ratio Test, Time Domain Data

1. Introduction

Wald's procedure is particularly relevant if the data is collected sequentially. Sequential Analysis is different from Classical Hypothesis. With classical hypothesis, first the entire data need to be collected and then the analysis is done to attain the conclusions. But where as in Sequential Analysis, each and every test case is analyzed soon after the data has been collected, and also the results are compared with some threshold value incorporating the new information obtained with the current test case. This permits one to come up with the conclusions during the data collection itself, so that the final decision may be made at much earlier stage. Wald's procedure is well suited if the data is collected sequentially. The main advantage of Sequential Analysis is that the decisions can be taken at an earlier time saving the human time and also in terms of money.

Sequential Probability Ratio Test (SPRT) is usually applied at the circumstances where we need to take

decision between two simple hypotheses or a single decision point. The⁴ SPRT procedure can be used to distinguish the software under test into one of the two categories like reliable/unreliable, pass/fail and certified/uncertified¹¹. The software failure data analysis can be done either by considering the time between failures or failure count in a specific time interval. Also it is assumed that the average number of recorded failures in a given time interval is directly proportional to the length of the interval. The random number of failure occurrences in a given interval can be explained by the Poisson process.

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

As per the observations made by the⁵, the reliability predictions are misleading by applying software reliability growth models in classical hypotheses testing. According to the observations made by him, the statistical methods can be successfully applied to the failure

*Author for correspondence

data. His observations are demonstrated by applying the well-known Sequential Probability Ratio Test (SPRT) of⁴ for a software failure data to detect unreliable software components and compare the reliability of different software versions. In this paper, we consider one of the popular software reliability growth model Burr Type XII and adopt the principle of⁵ in detecting whether the software is reliable or unreliable in order to accept or reject the developed software. The theory proposed by⁵ is described in section 2 Implementation of SPRT for the proposed Burr type XII Software Reliability Growth Model is illustrated in section 3. Maximum Likelihood estimation method is used to estimate the parameters is presented in Section 4. Application of the decision rule to detect the unreliable software with reference to the Software Reliability Growth Model Burr Type XII is depicted in section 5.

2. Wald's Sequential Test for a Poisson Process

The Sequential Probability Ratio Test (SPRT) was developed by Abraham Wald at Columbia University in 1943⁴. The SPRT procedure is used for quality control studies during the manufacturing of software products. The tests can be performed with fewer observations as compared to fixed sample size sets. Testing is performed on large volumes of data which consumes large amount of time in classical hypothesis and it can be reduced to a large extent by implementing Sequential Probability Ratio Tests. The SPRT methodology for Homogeneous Poisson Process given by Stieber is described below⁵.

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. 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 $\lambda \leq \lambda_1$. 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

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

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

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 $t_1 < t_2 < \dots < t_k$ and the corresponding realizations $N(t_1), N(t_2) \dots N(t_k)$ or $N(t)$. 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 favour of λ_1 or λ_0 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 continue³ the test process with one more observation in failure data, according to

$$\frac{P_1}{P_0} \geq A \quad (2.3)$$

$$\frac{P_1}{P_0} \leq B \quad (2.4)$$

$$B < \frac{P_1}{P_0} < A \quad (2.5)$$

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

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

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 $[t, N(t)]$ of is between the two linear boundaries given by equations (2.6) and (2.7) where

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

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

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

The parameters a, β, λ_0 and λ_1 can be chosen in several ways. One way suggested by ⁵ is

$$\lambda_0 = \frac{\lambda \log(q)}{q-1}$$

$$\lambda_1 = q \frac{\lambda \log q}{q-1} \text{ Where } q = \frac{\lambda_1}{\lambda_0}$$

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 or components.

3. Sequential Probability Ratio Test for Burr Type XII SRGM

In Section 2, for the Poisson process we know that the expected value $N(t) = \lambda(t)$ called the average number of failures experienced in time 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) as its mean value function the probability equation of such a process is

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

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

$$m(t) = a \left[1 - (1+t^c)^{-b} \right], t \geq 0$$

It can also be written as

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

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

Where $m_1(t), m_0(t)$ represents the mean value function at stated parameters indicating reliable software and unreliable software respectively. The mean value function $m(t)$ comprises the parameters 'a', 'b' and 'c'. The 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 ($c_0 < c_1$). For 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.

System is said to be reliable and can be accepted if

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

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

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

System is said to be unreliable and rejected if

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

$$\text{i.e., } \frac{e^{-m_1(t)} [m_1(t)]^{N(t)}}{e^{-m_0(t)} [m_0(t)]^{N(t)}} \geq A$$

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

Continue the test procedure as long as

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

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

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

Acceptance Region:

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

Rejection Region:

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

Continuation Region:

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

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

For the specified model, it may be observed that the decision rules are exclusively based on the strength of the sequential procedure (a, β) and the value of the mean value functions namely $m_0(t)$ $m_1(t)$. As described by⁵, 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), (3.2) and (3.3) are considered as generalizations for the decision procedure of⁵. SPRT procedure is applied on live software failure data sets and the results that were analyzed are illustrated in Section 5.

4. Parameter Estimation

In this section we develop expressions to estimate the parameters of the Burr type XII model based on time domain data. Parameter estimation is very significant in software reliability prediction. Once the analytical solution form is known for a given model, parameter estimation is achieved by applying a well-known estimation, Maximum Likelihood Estimation (MLE)².

The main idea behind Maximum Likelihood parameter assessment is to decide the parameters that maximize the probability (likelihood) of the specimen data. In the other words, MLE methods are versatile and applicable to most models and for different types of data. In this paper parameters are estimated from the time domain data.

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

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

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

The likelihood function for time domain data is given by

$$L = e^{-m(t)} \prod_{i=1}^n m'(t_i) \quad (4.2)$$

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

$$L = e^{-a[1-(1+t^c)^{-b}]} \prod_{i=1}^n \frac{abct_i^{c-1}}{(1+t_i^c)^{b+1}}$$

$$\text{Log} L = -a + a(1+t^c)^{-b} + \sum_{i=1}^n \left[\text{Log} a + \text{Log} b + \text{Log} c \right. \\ \left. + (c-1) \log t_i - (b+1) \log(1+t_i^c) \right] \quad (4.3)$$

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

$$\text{i.e., } \frac{\partial \text{Log} L}{\partial a} = 0$$

$$\therefore a = \frac{n(1+t^c)^b}{(1+t^c)^b - 1} \quad (4.4)$$

The parameter 'b' is estimated by iterative Newton Raphson Method using $b_{n+1} = b_n - \frac{g(b)}{g'(b)}$, Where $g(b)$ and $g'(b)$ are expressed as follows.

$$g(b) = \frac{\partial \text{Log} L}{\partial b} = 0$$

$$\frac{\partial \text{Log} L}{\partial b} = g(b) = \frac{n \log \left(\frac{1}{t+1} \right)}{(t+1)^b - 1} + \frac{n}{b} - \sum_{i=1}^n \log(t_i + 1) \quad (4.5)$$

$$g'(b) = \frac{\partial^2 \text{Log} L}{\partial b^2} = 0$$

$$\frac{\partial^2 \text{Log} L}{\partial b^2} = g'(b) = -n \left[\log \left(\frac{1}{t+1} \right) \left\{ \frac{(t+1)^b \log(t+1)}{\left[(t+1)^b - 1 \right]^2} \right\} + \frac{1}{b^2} \right] \quad (4.6)$$

The parameter 'c' is estimated by iterative Newton Raphson Method using

$$c_{n+1} = c_n - \frac{g(c)}{g'(c)}$$

Where $g(c)$ and $g'(c)$ are expressed as follows.

$$g(c) = \frac{\partial \text{Log} L}{\partial c} = 0$$

$$\frac{\partial \text{Log} L}{\partial c} = g(c) = \frac{-n}{(1+t^c)} \log(t) + \frac{n}{c} - \sum_{i=1}^n 2 \log(t) \frac{t_i^c}{(1+t_i^c)} + \sum_{i=1}^n \log t_i \quad (4.7)$$

$$g'(c) = \frac{\partial^2 \text{Log} L}{\partial c^2} = 0$$

$$\frac{\partial^2 \text{Log} L}{\partial c^2} = g'(c) = \frac{nt^c \log t}{(1+t^c)^2} \log t - \frac{n}{c^2} - 2 \log t \cdot \sum_{i=1}^n t_i^c \log t_i \left\{ \frac{1}{(1+t_i^c)^2} \right\} \quad (4.8)$$

5. SPRT Analysis of Live Datasets

In this section, the SPRT methodology is applied on five different data sets that are borrowed from^{2,7} and SONATA software services. The decisions are evaluated based on the considered mean value function. Based on the estimates of the parameter 'b' in each mean value function, we have chosen the specifications of $b_0 = b - \delta$, $b_1 = b + \delta$ and $c_0 = c - \delta$, $c_1 = c + \delta$, and apply SPRT such that $b_0 < b < b_1$ and $c_0 < c < c_1$. Assuming the δ value is 0.5, the choices are given in the following Table 1.

Using the specification b_0 , b_1 , and c_0 , c_1 the mean value functions $m_0(t)$ and $m_1(t)$ are computed for each 't'. Later the decisions are made based on the decision rules specified by the equations (3.4), (3.5) for the data sets. At each 't' of the data set, the strengths (α , β) are considered as (0.05, 0.2). SPRT procedure is applied on five different data sets and the necessary calculations are given in the Table 2.

Table 1. Estimates of a , b , c & specifications of b_0 , b_1 , c_0 , c_1

Data Set	Estimate of 'a'	Estimate of 'b'	b_0	b_1	Estimate of 'c'	c_0	c_1
Xie	30.040800	0.999825	0.499825	1.499825	0.999619	0.499619	1.499619
AT & T	22.032465	0.999859	0.499859	1.499859	0.999611	0.499611	1.499611
IBM	15.051045	0.999530	0.499530	1.499530	0.999196	0.499196	1.499196
NTDS	26.105273	0.998899	0.498899	1.498899	0.998903	0.498903	1.498903
SONATA	30.016391	0.999958	0.499958	1.499958	0.999920	0.499920	1.499920

Table 2. SPRT Analysis for 5 data sets

Data Set	T	N(t)	R.H.S. of equation (3.4) Acceptance region (\leq)	R.H.S. of equation (3.5) Rejection region (\geq)	Decision
Xie	30.02	1	0.003606	0.005130	Rejection
AT & T	5.50	1	0.115010	0.164486	Rejection
IBM	10	1	0.023281	0.040821	Rejection
NTDS	9	1	0.047184	0.065189	Rejection
SONATA	52.50	1	0.0010008	0.001499	Rejection

6. Conclusion

The SPRT methodology for the proposed software reliability growth model Burr type XII is applied for the software failure data sets. Hence, it is observed that we are able to come to a conclusion in less time regarding the reliability or unreliability of a software product. The results exemplifies that the model has given a decision of rejection for all the data sets at various time instant of the data. Therefore, we may conclude that, applying SPRT on data sets we can come to an early conclusion of reliable/unreliable of software.

7. References

1. Goel AL, Okumoto K. A Time Dependent Error Detection Rate Model for software Reliability and other performance measures. IEEE Transactions on Reliability. 1979; R-28: 206–11.
2. Pham H. System Software Reliability. Springer; 2006.
3. Satya Prasad R. Half Logistic Software Reliability Growth Model [PhD Thesis]. India: ANU; 2007.
4. Wald A. Sequential Analysis. New Impression edition. New York: John Wiley and Son, Inc; 1947 Sep 30.
5. Stieber HA. Statistical Quality Control: How to detect unreliable software components. Proceedings of the 8th International symposium on Software Reliability Engineering; 1997. p. 8–12.
6. Wood A. Predicting Software Reliability. IEEE Computer; 1996. p. 2253–64.
7. Lyu MR. The Hand book of software Reliability engineering. McGrawHill and IEEE Computer Society Press; 1995. ISBN: 9-07-039400-8.
8. Satya Prasad R, Rama Devi B, Sridevi G. Detection of Burr type XII Reliable software using SPRT on Interval domain data. International Journal of Computer Science and Information Technologies. 2015; 6(2):1806–11.
9. Musa JD, Iannino A, Okumoto K. Software Reliability: measurement prediction Application. New York: McGraw Hill; 1987. ISBN: 0-07-044093-X.
10. Kim H-C. Assessing Software Reliability based on NHPP using SPC. International Journal of Software Engineering and its Applications. 2013; 7(6):61–70.
11. 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.
12. Ashoka M. Data set. Bangalore: Sonata Software Limited; 2010.
13. Musa JD. A theory of software reliability and its applications. IEEE Transactions on Software Engineering. 1975 Sep; SE-1(33):312–7.