Computer Engineering and Intelligent Systems www.iiste.org
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol 2, No.3

Detection of Reliable Software Using SPRT on Interval Domain Data

Krishna Mohan Gonuguntla
Reader, Dept. of Computer Science, P.B.Siddhartha college
Vijayawada, Andhrapradesh, India.
Tel: 91-9440446847 E-mail: km_mm_2000@yahoo.com

Satya Prasad Ravi
Associate Professor, Dept. of Computer Science & Engg., Acharya Nagrjuna University
Nagarjuna Nagar, Guntur, Andhrapradesh, India
Tel: 91-9848487478 E-mail: profrsp@gmail.com

Abstract

In Classical Hypothesis testing volumes of data is to be collected and then the conclusions are drawn which may take more time. But, Sequential Analysis of statistical science could be adopted in order to decide upon the reliable / unreliable of the developed software very quickly. The procedure adopted for this is, Sequential Probability Ratio Test (SPRT). In the present paper we proposed the performance of SPRT on Interval domain data using Weibull model and analyzed the results by applying on 5 data sets. The parameters are estimated using Maximum Likelihood Estimation.

Keywords: Weibull model, Sequential Probability Ratio Test, Maximum Likelihood Estimation, Decision lines, Software Reliability, Interval domain data.

1. Introduction

Wald's procedure is particularly relevant if the data is collected sequentially. Sequential Analysis is different from Classical Hypothesis Testing were the number of cases tested or collected is fixed at the beginning of the experiment. In Classical Hypothesis Testing the data collection is executed without analysis and consideration of the data. After all data is collected the analysis is done and conclusions are drawn. However, in Sequential Analysis every case is analyzed directly after being collected, the data collected upto that moment is then compared with certain threshold values, incorporating the new information obtained from the freshly collected case. This approach allows one to draw conclusions during the data collection, and a final conclusion can possibly be reached at a much earlier stage as is the case in Classical Hypothesis Testing. The advantages of Sequential Analysis are easy to see. As data collection can be terminated after fewer cases and decisions taken earlier, the savings in terms of human life and misery, and financial savings, might be considerable.

In the analysis of software failure data we often deal with either Time Between Failures or failure count in a given time interval. If it is further assumed that the average number of recorded failures in a given time interval is directly proportional to the length of the interval and the random number of failure occurrences in the interval is explained by a Poisson process then we know that the probability equation of the stochastic process representing the failure occurrences is given by a homogeneous poisson process with the expression

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

(Stieber 1997) observes that if classical testing strategies are used, the application of software reliability growth models may be difficult and reliability predictions can be misleading. However, he observes that statistical methods can be successfully applied to the failure data. He demonstrated his observation by applying the well-known sequential probability ratio test of (Wald 1947) for a software failure data to detect unreliable software components and compare the reliability of different software versions. In this paper we consider popular SRGM Weibull model and adopt the principle of Stieber in detecting unreliable software components in order to accept or reject the developed software. The theory proposed by Stieber is presented in Section 2 for a ready reference. Extension of this theory to the SRGM – Weibull is presented in Section 3. Maximum Likelihood parameter estimation method is presented in Section 4. Application of the decision rule to detect unreliable software components with respect to the proposed SRGM is given in Section 5.

2. Wald's Sequential Test for a Poisson Process

The sequential probability ratio test was developed by A.Wald at Columbia University in 1943. Due to its usefulness in development work on military and naval equipment it was classified as 'Restricted' by the Espionage Act (Wald 1947). A big advantage of sequential tests is that they require fewer observations (time) on the average than fixed sample size tests. SPRTs are widely used for statistical quality control in manufacturing processes. An SPRT for homogeneous Poisson processes is described below.

Let $\{N(t),t \geq 0\}$ be a homogeneous Poisson process with rate 'λ'. In our case, N(t) = number of failures up to time 't' and 'λ' is the failure rate (failures per unit time). Suppose that we put a system on test (for example a software system, where testing is done according to a usage profile and no faults are corrected) and that we want to estimate its failure rate 'λ'. We can not expect to estimate 'λ' precisely. But we want to reject the system with a high probability if our data suggest that the failure rate is larger than λ_1 and accept it with a high probability, if it's smaller than λ_0. As always with statistical tests, there is some risk to get the wrong answers. So 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. 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

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

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

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 $t_1 < t_2 < t_3 < \dots\dots < t_K$ and the corresponding realizations $N(t_1), N(t_2), \dots\dots N(t_K)$ 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 (Satyaprasad 2007) the test process with one more observation in failure data, according as

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

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

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

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 $N_U\left(t\right) = a.t + b_2$ (2.6)

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

$$N_L\left(t\right) = a.t - b_1 \qquad (2.7)$$

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

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

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

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

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

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

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 (components).

3. Sequential Test for Software Reliability Growth Models

In Section 2, for the Poisson process we know that the expected value of N(t) = λt called the average number of failures experienced in time 't' .This 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

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

Depending on the forms of m(t) we get various Poisson processes called NHPP. For our Rayleigh model the mean value function is given as $m(t)=a\left(1-e^{-(bt)^2}\right)$ where $a>0, b>0$.

We may write

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

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

Where, $m_1(t)$, $m_0(t)$ are values of the mean value function at specified sets of its parameters indicating reliable software and unreliable software respectively. Let P_0 , P_1 be values of the NHPP at two specifications of b say b_0, b_1 where $(b_0<b_1)$ respectively. It can be shown that for our models $m(t)$ at b_1 is greater than that at b_0 . Symbolically $m_0(t)<m_1(t)$. Then the SPRT procedure is as follows:

Accept the system to be reliable if $\frac{P_1}{P_0}\le B$

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

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

Decide the system to be unreliable and reject if $\frac{P_1}{P_0} \geq A$

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

Continue the test procedure as long as

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

Substituting the appropriate expressions of the respective mean value function – m(t) of Rayleigh we get the respective decision rules and are given in followings lines

Acceptance region:

$$N(t) \leq \frac{\log\left(\frac{\beta}{1-\alpha}\right) + a\left(e^{-(b_0 t)^2} - e^{-(b_1 t)^2}\right)}{\log\left(\frac{1-e^{-(b_1 t)^2}}{1-e^{-(b_0 t)^2}}\right)} \qquad (3.4)$$

Rejection region:

$$N(t) \geq \frac{\log\left(\frac{1-\beta}{\alpha}\right) + a\left(e^{-(b_0 t)^2} - e^{-(b_1 t)^2}\right)}{\log\left(\frac{1-e^{-(b_1 t)^2}}{1-e^{-(b_0 t)^2}}\right)} \qquad (3.5)$$

Continuation region:

$$\frac{\log\left(\frac{\beta}{1-\alpha}\right) + a\left(e^{-(b_0 t)^2} - e^{-(b_1 t)^2}\right)}{\log\left(\frac{1-e^{-(b_1 t)^2}}{1-e^{-(b_0 t)^2}}\right)} < N(t) < \frac{\log\left(\frac{1-\beta}{\alpha}\right) + a\left(e^{-(b_0 t)^2} - e^{-(b_1 t)^2}\right)}{\log\left(\frac{1-e^{-(b_1 t)^2}}{1-e^{-(b_0 t)^2}}\right)} \qquad (3.6)$$

It may be noted that in the above model the decision rules are exclusively based on the strength of the sequential procedure (α,β) and the values of the respective mean value functions namely, $m_0(t), m_1(t)$.

If the mean value function is linear in ‘t’ passing through origin, that is, m(t) = λt the decision rules become decision lines as described by (Stieber 1997). In that sense equations (3.1), (3.2) , (3.3) can be regarded as generalizations to the decision procedure of Stieber. The applications of these results for live software failure data are presented with analysis in Section 5.

4. ML (Maximum Likelihood) Parameter Estimation

Parameter estimation is of primary importance in software reliability prediction. Once the analytical solution for $m(t)$ is known for a given model, parameter estimation is achieved by applying a technique of Maximum Likelihood Estimate (MLE). Depending on the format in which test data are available, two different approaches are frequently used. A set of failure data is usually collected in one of two common ways, time domain data and interval domain data.

The idea behind maximum likelihood parameter estimation is to determine the parameters that maximize the probability (likelihood) of the sample data. The method of maximum likelihood is considered to be more robust (with some exceptions) and yields estimators with good statistical properties. In other words, MLE methods are versatile and apply to many models and to different types of data. Although the methodology for maximum likelihood estimation is simple, the implementation is mathematically intense. Using today's computer power, however, mathematical complexity is not a big obstacle. Assuming that the data are given for the cumulative number of detected errors y_i in a given time-interval $(0,t_i)$ where i = 1,2, …, n. and $0 < t_1 < t_2 < \ldots < t_n$ then the log likelihood function (LLF) takes on the following form.

Likely hood function by using λ(t) is: $L = \prod_{i=1}^{n} \lambda(t_i)$

The logarithmic likelihood function for interval domain data (Pham 2006) is given by:

$$\text{Log L} = \sum_{i=1}^{n} (y_i - y_{i-1}).\log\left[m(t_i) - m(t_{i-1})\right] - m(t_n)$$

The maximum likelihood estimators (MLE) of $\theta_1, \theta_2, \ldots, \theta_k$ are obtained by maximizing L or Λ, where Λ is ln L . By maximizing Λ, which is much easier to work with than L, the maximum likelihood estimators of $\theta_1, \theta_2, \ldots, \theta_k$ are the simultaneous solutions of k equations such that: $\frac{\partial(\Lambda)}{\partial\theta_j} = 0$, j=1,2,…,k

The parameters 'a' and 'b' are estimated using iterative Newton Raphson Method, which is given as

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)}.$$

5. SPRT Analysis of Live Data Sets

We see that the developed SPRT methodology is for a software failure data which is of the form [t, N(t)] where N(t) is the failure number of software system or its sub system in 't' units of time. In this section we evaluate the decision rules based on the considered mean value function for Five different data sets of the above form, borrowed from (Pham 2006), (Pham 2007), (Ohba 1984a), (Misra 1983). Based on the estimates of the parameter 'b' in each mean value function, we have chosen the specifications of $b_0 = b - \delta$, $b_1 = b + \delta$ equidistant on either side of estimate of b obtained through a data set to apply SPRT such that $b_0 < b < b_1$. Assuming the value of $\delta = 0.125$, the estimates are given in the following table.

Table 1: Estimates of a,b & Specifications of b0, b1

Data Set	Estimate of 'a'	Estimate of 'b'	b_0	b_1
DS 1	44.88134	0.070928	-0.054072	0.195928
DS 2	28.129276	0.070566	-0.054434	0.195566
DS 3	81.026164	0.040014	-0.084986	0.165014
DS 4	54.765902	0.062527	-0.062473	0.187527
DS 5	164.246019	0.036012	-0.088988	0.161012

Using the selected b_0, b_1 and subsequently the $m_0(t), m_1(t)$ for the model, we calculated the decision rules given by Equations 3.1, 3.2, sequentially at each 't' of the data sets taking the strength (α, β) as (0.05, 0.2). These are presented for the model in Tables 2.

Table 2: SPRT analysis for 5 data sets

Data Set	T	N(t)	R.H.S of equation (3.4) Acceptance region (≤)	R.H.S of Equation (3.5) Rejection Region(≥)	Decision
DS 1	1	3	0.000412292	1.67079809	Reject
DS 2	1	1	-0.2274684	1.45403022	Accept
	2	1	0.798963147	2.44603498	
	3	2	2.1655486	3.75685608	
DS 3	1	6	0.026389907	3.26524210	Reject
DS 4	1	2	0.054639651	2.01065464	Accept
	2	3	2.113395933	4.02819412	
	3	4	4.869322753	6.71755071	
DS 5	1	9	1.129459108	4.75355492	Reject

From the above table we see that a decision either to accept or reject the system is reached much in advance of the last time instant of the data.

6. Conclusion

The table 2 shows that Weibull model as exemplified for 5 Data Sets indicate that the model is performing well in arriving at a decision. Out of 5 Data Sets the procedure applied on the model has given a decision of rejection for 3, acceptance for 2 and continue for none at various time instants of the data as follows. DS1, DS3 and DS5 are rejected at 1st instant of time. DS2 and DS4 are accepted at 3rd instant of time. Therefore, we may conclude that, applying SPRT on data sets we can come to an early conclusion of reliable / unreliable of software.

References

Misra, P. (1983). "Software Reliability Analysis," *IBM Systems Journal*, vol. 22, pp. 262–270.

Ohba, M. (1984a), “Inflection S-shaped software reliability growth models”, Stochastic Models in Reliability Theory (Osaki,S.and Hatoyama, Y. Editors), pp. 144-162, Springer Verlag Merlin.

Pham, H. (2007). “An Imperfect-debugging Fault-detection Dependent-parameter Software”, International Journal of Automation and Computing, 04(4), October, 325-328, DOI: 10.1007/s11633-007-0325-8.

Pham. H., (2006). “*System software reliability*”, Springer.

Satya Prasad, R., (2007). ”*Half logistic Software reliability growth model* “, Ph.D Thesis of ANU, India.

Stieber, H.A. (1997). “*Statistical Quality Control: How To Detect Unreliable Software Components”*, Proceedings the 8th International Symposium on Software Reliability Engineering, 8-12.

Wald. A., (1947). “Sequential Analysis”, John Wiley and Son, Inc, New York.

Wood, A. (1996). “*Predicting Software Reliability”*, IEEE Computer, 2253-2264.

Author Profile:

First Author:

Mr. G. Krishna Mohan is working as a Reader in the Department of Computer Science, P.B.Siddhartha College, Vijayawada. He obtained his M.C.A degree from Acharya Nagarjuna University in 2000, M.Tech from JNTU, Kakinada, M.Phil from Madurai Kamaraj University and pursuing Ph.D at Acharya Nagarjuna University. His research interests lies in Data Mining and Software Engineering.

Second Author:

Dr. R. Satya Prasad received Ph.D. degree in Computer Science in the faculty of Engineering in 2007 from Acharya Nagarjuna University, Andhra Pradesh. He received gold medal from Acharya Nagarjuna University for his outstanding performance in Masters Degree. He is currently working as Associate Professor and H.O.D, in the Department of Computer Science & Engineering, Acharya Nagarjuna University. His current research is focused on Software Engineering. He has published several papers in National & International Journals.