

## Exponential software reliability using SPRT: MLE

S. Murali Mohan<sup>1</sup>, Dr. R. Satya Prasad<sup>2</sup>, G. Krishna Mohan<sup>3</sup>

<sup>2</sup>(Controller of Examinations, Vikrama Simhapuri University, Nellore)

<sup>1</sup>(Associate Professor Dept. of CS&Engg. Acharya Nagarjuna University, Nagarjuna Nagar)

<sup>3</sup>(Reader, Dept. of Computer Science, P.B.Siddhartha College, Vijayawada)

**Abstract:** In Classical Hypothesis testing volumes of data is to be collected and then the conclusions are drawn, which may need more time. But, Sequential Analysis of Statistical science could be adopted in order to decide upon the reliability or unreliability of the developed software very quickly. The procedure adopted for this is, Sequential Probability Ratio Test (SPRT). It is designed for continuous monitoring. The likelihood based SPRT proposed by Wald is very general and it can be used for many different probability distributions. In the present paper we propose the performance of SPRT on 4 data sets of Time domain data using exponential model and analyzed the results. The parameters are estimated using Maximum Likelihood Estimation method.

**Keywords:** SPRT, SRGM, HPP, NHPP, MLE, Goel-Okumoto model.

### I. INTRODUCTION

Sequential analysis is a method of statistical inference whose main feature is that the number of observations required by the procedure is not determined in advance. The decision to end the observations depends, at each stage, on the results of the samples already taken. (SPRT), which is usually applied in situations, requires a decision between two simple hypothesis or a single decision point. Wald's (1947) SPRT procedure has been used to classify the software under test into one of two categories (e.g., reliable/unreliable, pass/fail, certified/noncertified) (Reckase, 1983). Wald's procedure is particularly relevant if the data is collected sequentially. Classical Hypothesis Testing is different from Sequential Analysis. In Classical Hypothesis testing, the number of cases tested or collected is fixed at the beginning of the experiment. In this method, the analysis is made and conclusions are drawn after collecting the complete data.

In the analysis of software failure data, either TBFs or failure count in a given time interval is dealt with. 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 it is known that the probability equation of the stochastic process representing the failure occurrences is given by a Homogeneous Poisson Process with the expression

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

Stieber (1997) observes that, the application of SRGMs may be difficult and reliability predictions can be misleading, if classical testing strategies are used. 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 for a software failure data to detect unreliable software components and compare the reliability of different software versions. In this chapter the popular SRGM – Exponential model is considered and the principle of Stieber is adopted in detecting unreliable software 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 considered SRGM is presented in Section 3. Maximum Likelihood parameter estimation method is presented in Section 4. Application of the decision rule to detect unreliable software with reference to the SRGM is given in Section 5.

### II. SEQUENTIAL TEST FOR A POISSON PROCESS

A.Wald, developed the SPRT at Columbia University in 1943. 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. The SPRT for Homogeneous Poisson Processes is described below.

Let  $\{N(t), t \geq 0\}$  be a homogeneous Poisson process with rate ' $\lambda$ '. In this case,  $N(t)$  = number of failures up to time ' $t$ ' and ' $\lambda$ ' is the failure rate (failures per unit time). If the system is put on test (for example a software system, where testing is done according to a usage profile and no faults are corrected) and that if we want to estimate its failure rate ' $\lambda$ '. We can not expect to estimate ' $\lambda$ ' precisely. But we want to

reject the system with a high probability if the data suggest that the failure rate is larger than  $\lambda_1$  and accept it with a high probability, if it is smaller than  $\lambda_0$ . As always with statistical tests, there is some risk to get the wrong answers. So we have to specify two (small) numbers ‘ $\alpha$ ’ and ‘ $\beta$ ’, where ‘ $\alpha$ ’ 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. '  $\beta$ ' 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  $t > 0$  as additional data are collected. With specified choices of  $\lambda_0$  and  $\lambda_1$  such that  $0 < \lambda_0 < \lambda_1$  , the probability of finding  $N(t)$  failures in the time span  $(0, t)$  with  $\lambda_1, \lambda_0$  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  $\lambda_0$  or  $\lambda_1$  , 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 favour of  $\lambda_1$ , in favour of  $\lambda_0$  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 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  $A \cong \frac{1-\beta}{\alpha}$  ,  $B \cong \frac{\beta}{1-\alpha}$

Where ‘ $\alpha$ ’ and ‘ $\beta$ ’ are the risk probabilities as defined earlier. A good test is one that makes the  $\alpha$  and  $\beta$  errors as small as possible. The common procedure is to fix the  $\beta$  error and then choose a critical region to minimize the error or maximize the power i.e  $1 - \beta$  of the test. 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

$$N_U(t) = a.t + b_2 \tag{2.6}$$

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

$$N_L(t) = a.t - b_1 \tag{2.7}$$

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

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

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

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

The parameters  $\alpha, \beta, \lambda_0$  and  $\lambda_1$  can be chosen in several ways. One way suggested by Stieber is  $\lambda_0 = \frac{\lambda \cdot \log(q)}{q-1}, \lambda_1 = q \frac{\lambda \cdot \log(q)}{q-1}$  where  $q = \frac{\lambda_1}{\lambda_0}$

If  $\lambda_0$  and  $\lambda_1$  are chosen in this way, the slope of  $N_U(t)$  and  $N_L(t)$  equals  $\lambda$ . The other two ways of choosing  $\lambda_0$  and  $\lambda_1$  are from past projects (for a comparison of the projects) and from part of the data to compare the reliability of different functional areas.

### III. SEQUENTIAL TEST FOR SOFTWARE RELIABILITY GROWTH MODELS

In Section 2, for the Poisson process it is known that the expected value of  $N(t) = \lambda t$  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

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

Depending on the forms of  $m(t)$  various Poisson processes called NHPP are obtained. For our two parameter Exponential model, the mean value function is given as  $m(t) = a(1 - e^{-bt})$  where  $a > 0, b > 0$

It may be written as

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

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

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  $(b_0 < b_1)$ . It can be shown that for our model  $m(t)$  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,  $\frac{P_1}{P_0} \leq B$

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

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

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

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

Continue the test procedure as long as

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

Substituting the appropriate expressions of the respective mean value function  $-m(t)$  of Exponential we get the respective decision rules and are given in following lines  
Acceptance region:

$$N(t) \leq \frac{\log\left(\frac{\beta}{1-\alpha}\right) + a\left(e^{-(b_0t)} - e^{-(b_1t)}\right)}{\log\left(\frac{1-e^{-(b_1t)}}{1-e^{-(b_0t)}}\right)} \quad (3.4)$$

Rejection region:

$$N(t) \geq \frac{\log\left(\frac{1-\beta}{\alpha}\right) + a\left(e^{-(b_0t)} - e^{-(b_1t)}\right)}{\log\left(\frac{1-e^{-(b_1t)}}{1-e^{-(b_0t)}}\right)} \quad (3.5)$$

Continuation region:

$$\frac{\log\left(\frac{\beta}{1-\alpha}\right) + a\left(e^{-(b_0t)} - e^{-(b_1t)}\right)}{\log\left(\frac{1-e^{-(b_1t)}}{1-e^{-(b_0t)}}\right)} < N(t) < \frac{\log\left(\frac{1-\beta}{\alpha}\right) + a\left(e^{-(b_0t)} - e^{-(b_1t)}\right)}{\log\left(\frac{1-e^{-(b_1t)}}{1-e^{-(b_0t)}}\right)} \quad (3.6)$$

It may be noted that in the above mentioned model the decision rules are exclusively based on the strength of the sequential procedure  $(\alpha, \beta)$  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,  $m(t) = \lambda t$  the decision rules become decision lines as described by Stieber. 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.

#### IV. MAXIMUM LIKELIHOOD PARAMETER ESTIMATION: G-O MODEL

The likelihood function of G-O model is given as,  $L = \prod_{i=1}^N abe^{-bt}$

Taking the natural logarithm on both sides, The Log Likelihood function is given as:

$$\log L = \sum_{i=1}^n \log(abe^{-bt_i}) - a[1 - e^{-bt_n}] \quad (4.1)$$

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

$$a = \frac{n}{\left[1 - e^{-(bt_n)}\right]} \tag{4.2}$$

Taking the Partial derivative of log L with respect to ‘b’ and equating to ‘0’.

$$g(b) = \sum_{i=1}^n t_i - \frac{n}{b} + nt_n \frac{e^{-(bt_n)}}{\left(1 - e^{-(bt_n)}\right)} = 0 \tag{4.3}$$

Taking the partial derivative of log L again with respect to ‘b’ and equating to ‘0’.

$$g'(b) = \frac{n}{b^2} - nt_n^2 \left\{ \frac{1}{\left(1 - e^{-(bt_n)}\right)} + \frac{e^{-(bt_n)}}{\left(1 - e^{-(bt_n)}\right)^2} \right\} e^{-bt_n} \tag{4.4}$$

The parameters ‘a’ and ‘b’ are estimated as follows. The parameter ‘b’ is estimated by iterative Newton Raphson Method using  $b_{n+1} = b_n - \frac{g(b_n)}{g'(b_n)}$ , which is substituted in finding ‘a’.

**V. SPRT ANALYSIS OF DATA SETS : TIME DOMAIN**

In this section, the developed SPRT methodology is shown for a software failure data which is of time domain. The decision rules based on the considered mean value function for Four different data sets, borrowed from Pham (2006), Xie *et al.*, (2002) are evaluated. 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  $\delta = 0.0025$ , the choices are given in the following table.

**Table 5.1: Estimates of a, b & Specifications of b<sub>0</sub>, b<sub>1</sub> for Time domain**

Data Set	Estimate of ‘a’	Estimate of ‘b’	b <sub>0</sub>	b <sub>1</sub>
1 (Xie)	31.899246	0.003819	0.001319	0.006319
2 (AT&T)	23.582254	0.003973	0.001473	0.006473
3 (NTDS)	30.168926	0.007917	0.005417	0.010417
4 (IBM)	17.608791	0.006451	0.003951	0.008951

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 3.4 and 3.5, sequentially at each ‘t’ of the data sets taking the strength  $(\alpha, \beta)$  as (0.05, 0.2). These are presented for the model in Table 5.2.

**Table 5.2: SPRT analysis for 5 data sets of Time domain data**

Data Set	T	N(t)	Acceptance region ( $\leq$ )	Rejection Region ( $\geq$ )	Decision	
1	1	30.02	1.818603	4.719158	Accept	
	2	1	5.5	-0.629760		2.323070
		3	10.08	-0.285944		2.689878
3	1	9	-0.465126	6.390944	Accept	
	2	21	1.867158	9.050913		
	3	32	3.798211	11.300292		
4	1	10	-0.923786	4.536966	Continue	
	2	19	-0.100773	5.514319		
	3	32	0.994143	6.842165		
	4	43	1.840126	7.894901		
	5	58	2.884670	9.236447		
	6	70	3.636896	10.239468		
	7	88	4.638963	11.641133		

8	103	5.368646	12.726648
9	125	6.283826	14.204225
10	150	7.125308	15.748486
11	169	7.639797	16.846683
12	199	8.258289	18.483307
13	231	8.686670	20.143135
14	256	8.873289	21.409309
15	296	8.927876	23.435562

From the above table it is observed that a decision of either to accept or reject the system is reached well in advance of the last time instant of the data.

## VI. CONCLUSION

The table 5.2 of Time domain data as exemplified for 4 Data Sets shows that Exponential model is performing well in arriving at a decision. Out of 4 Data Sets of Time domain the procedure applied on the model has given a decision of rejection for 1, acceptance for 2 and continue for 1 at various time instant of the data as follows. Data Set #1 and #3 are accepted at 1<sup>st</sup> and 3<sup>rd</sup> instant of time respectively. Data Set #2 is rejected at 3<sup>rd</sup> instant of time. Data Set #4 is continued. Therefore, by applying SPRT on data sets it can be concluded that we can come to an early conclusion of reliable or unreliable software.

## REFERENCES

- [1] Goel, A.L and Okumoto, K. (1979). "A Time Dependent Error Detection Rate Model For Software Reliability And Other Performance Measures", IEEE Transactions on Reliability, vol.R-28, pp.206-211, 1979.
- [2] Pham. H., (2006). "System software reliability", Springer.
- [3] Reckase, M. (1983). A procedure for Decision Making Using Tailored Testing, 238-257, New Horizons In Testing – Latent Trait Test Theory and Computerized Adaptive Testing, Ed. David J. Weiss, Academic Press, New York.
- [4] Satya Prasad (2007). "Half logistic Software reliability growth model "Ph.D Thesis of ANU, India.
- [5] Stieber, H.A. (1997). "Statistical Quality Control: How To Detect Unreliable Software Components", Proceedings the 8th International Symposium on Software Reliability Engineering, 8-12.
- [6] Wald. A., 1947. "Sequential Analysis", John Wiley and Son, Inc, New York.
- [7] Xie, M., Goh. T.N., Ranjan.P., "Some effective control chart procedures for reliability monitoring" -Reliability engineering and System Safety 77 143 -150, 2002.

## Authors



**MURALI MOHAN. S.**, working as controller of examinations in Vikrama Simhapuri University. He worked as a controller of examinations in Dravidian University for three and half years. He worked as a Principal of an affiliated college of Andhra University in Visakhapatnam (A.P). He is presently a research scholar in the Department of Computer Science of Acharya Nagarjuna University.



**Dr. R Satya Prasad** received Ph.D. degree in Computer Science in the faculty of Engineering in 2007 from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. He have a satisfactory consistent academic track of record and received gold medal from Acharya Nagarjuna University for his outstanding performance in a first rank in Masters Degree. He is currently working as Associate Professor in the Department of Computer Science & Engineering, Acharya Nagarjuna University. He has occupied various academic responsibilities like practical examiner, project adjudicator, external member of board of examiners for various Universities and Colleges in and around in Andhra Pradesh. His current research is focused on Software Engineering, Image Processing & Database Management System. He has published several papers in National & International Journals.



**Mr. G. Krishna Mohan**, 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, M.Tech from JNTU, Kakinada, M.Phil from Madurai Kamaraj University and pursuing Ph.D from Acharya Nagarjuna University. He qualified AP State Level Eligibility Test. His research interests lies in Data Mining and Software Engineering. He published 16 research papers in various National and International journals.