Detection of Reliable Software using S-Shaped Model

R Satya Prasad, PhD.
Associate Professor
CSE Department
Acharya Nagarajuna University
Guntur, India

Y. Sangeetha
Assistant Professor
IT Department
V.R. Siddhartha Engg College
Vijayawada, India

ABSTRACT

Software reliability growth models using Non-Homogeneous Poisson Process (NHPP) with mean value function – dependent on fault detection rate is considered. The well known Sequential Probability Ratio Test (SPRT) procedure of statistical science is adopted for the model in order to decide upon the reliability / unreliability of developed software. In the present paper, we have proposed the performance of SPRT on interval domain data using Inflection S- Shaped model and analyzed the results by applying on 11 data sets.

Keywords

Inflection S-Shaped model, Maximum Likelihood Estimation (MLE), Software testing, Software failure data.

1. INTRODUCTION

In the analysis of software failure data we often deal with either inter failure times or number of recorded failures 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

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

Observes that if classical testing strategies are used (no usage testing), the application of software reliability growth models may be difficult and reliability predictions can be misleading [4]. However, he observes that statistical methods can be successfully applied to failure data. He demonstrated his observation by applying the well known sequential probability ratio test (SPRT) of for a software failure data to detect unreliable software components and compare the reliability of different software versions [5]. In this paper we consider a popular SRGM proposed by [2]. And adopt the principle of [4] in detecting unreliable software components in order to accept/reject developed software. For brevity we denote the SRGM as Inflection S-Shaped model. The theory proposed by [4] is presented in Section 2 for a ready reference. The procedure for parameter estimation is presented in section 3. Extension of this theory to the SRGM – Inflection S-Shaped 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 (SPRT) 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 [5]. 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≥ 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 cannot 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 λ1 and accept it with a high probability, if it's smaller than λ_0 ($0 < \lambda_0 < \lambda_1$). As always with statically 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 $\lambda \le \lambda_0$. This is the "producer's" risk. β is the probability of falsely accepting the system. That is accepting the system even if λ ≥ 1. This is the "consumer's" risk. With specified choices of λ_0 and λ1 such that 0 < λ0 < λ1, the probability of finding N (t) failures in the time span (0, t) with λ1, λ_0 as the failure rates are respectively given by

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

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

The ratio $\frac{P_1}{P0}$ at any time't' is considered at a measure of deciding the truth towards λ_0 or λ_1, given a sequence ofTime instants say $t_1 < t_2 < t_3 <\ldots\ldots\ldots <t_k$ and the corresponding realizations

N $(t_1), N(t_2)$, ……..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 the test process with one more observation in failure data, according as

$$\frac{p_1}{p_0} \geq A \quad (2.3)$$

$$\frac{p_1}{p_0} \leq B \quad (2.4)$$

$$B < \frac{p_1}{p_0} < A \quad (2.5)$$

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

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

Where α and $'\beta'$ 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

$$N_u(t) = a.t + b_2 \quad (2.6)$$

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

$$N_L(t) = a.t - b_1 \quad (2.7)$$

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

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

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

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

The parameters $\alpha, \beta, \lambda_0, \lambda_1$ can be chosen in several ways. One way suggested by [5] is

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

If λ_0 and λ_1 are chosen in this way, the slope of NU (t) and NL (t) equals λ. The other two ways of choosing λ_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 (components).

3. ESTIMATION BASED ON FAILURE COUNT DATA

Let 0 t_1 $_2$.......t_k be a partition of the interval [0,T] and suppose we observe the cumulative number of failures N(t_i) , up to and including time t_i , i=1,2,…k, The log likelihood function for this type of data, where the observations N(t_i) are denoted by n_i,i=1,2,…k , is given by

$$\text{Log L} = \sum_{i=1}^{k}\left[(n_i - n_{i-1})\log\left(m(t_i) - m(t_{i-1})\right)\right] - m(t_k)$$

$$m(t_i) = a\left(\frac{1-e^{-bt_i}}{1+\beta e^{-bt_i}}\right), \ a>0,\ b>0, t_i \geq 0 \quad (3.1)$$

where is the mean value function of Inflection S-Shaped with 'a','b' as its parameters. [7] To get MLEs of 'a' and 'b' the estimating equations are

$$\frac{\partial \log L}{\partial a} = 0$$

$$\alpha = n_k \frac{1+e^{-bt_k}}{1-e^{-bt_k}} \quad (3.2)$$

$$g(b) = \sum_{i=1}^{n}(y_i - y_{i-1})$$

$$\left[\left(\frac{t_i e^{-bt_i} - t_{i-1}e^{-bt_{i-1}}}{e^{-bt_{i-1}} - e^{-bt_i}}\right) + \left(\frac{\beta t_i e^{-bt_i}}{1+\beta e^{-bt_i}}\right) + \left(\frac{\beta t_{i-1} e^{-bt_{i-1}}}{1+\beta e^{-bt_{i-1}}}\right)\right] -$$

$$\frac{n_k t_k e^{-bt_k}(1-\beta+2\beta e^{-bt_k})}{(1-e^{-bt_k})(1+\beta e^{-bt_k})} \quad (3.3)$$

Iterative solution of g(b)=0 would give MLE of 'b' by applying Newton Rapson's Method. In order to get the asymptotic variances and covariance of the MLEs $\hat{a}$, $\hat{b}$ for the present model of Inflection S-Shaped model, the parameters are estimated from [8].

4. 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

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

Depending on the forms of m(t) we get various Poisson processes called NHPP for our model the mean value function is Inflection S-Shaped :

$$m(t) = a\left[\frac{1-e^{-bt}}{1+e^{-bt}}\right] \text{ where } a>0, b>0, t>0$$

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. For instance the model we have been considering their m (t) function, contains a pair of parameters a, b with 'a' as a multiplier. Also a, b are positive. Let $p_{0,}$ p_1, be values of the NHPP at two specifications of b say b_0 , b_1 $(b_0 < b_1)$ respectively. It can be shown that for our model m (t) at b1 is greater than that at b0. Symbolically m0 (t) <m1 (t). Then the SPRT procedure is as follows:

Accept the system to be reliable $\frac{p_1}{p_0} \leq B$

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

$$\text{i.e., } N(t) \leq \frac{\log\left(\frac{\beta}{1-\alpha}\right) + m_1(t) - m_0(t)}{log m_1(t) - log m_0(t)} \quad (3.4)$$

Decide the system to be unreliable and reject if $\frac{p_1}{p_0} \geq A$

$$\text{i.e., } N(t) \geq \frac{\log\left(\frac{1-\beta}{\alpha}\right) + m_1(t) - m_0(t)}{log m_1(t) - log m_0(t)} \quad (3.5)$$

Continue the test procedure as long as

$$\frac{\log\left(\frac{\beta}{1-\alpha}\right)+m_1(t)-m_0(t)}{logm_1(t)-logm_0(t)} < N(t) < \frac{\log\left(\frac{1-\beta}{\alpha}\right)+m_1(t)-m_0(t)}{logm_1(t)-logm_0(t)} \quad (3.6)$$

Substituting the appropriate expressions of the respective mean value functions m (t) of Inflection S-Shaped we get the decision rules and is given in followings lines

$$m(t) = \alpha\left[\frac{1-e^{-bt}}{1+e^{-bt}}\right] \quad \text{where a>0, b>0 and t>0}$$

Acceptance region:

$$N(t) \leq \frac{\log\left(\frac{\beta}{1-\alpha}\right)+a\left[\frac{\left[e^{-b_0t}-e^{-b_1t}+\beta\left(e^{-b_0t}-e^{-b_1t}\right)\right]}{\left(1+\beta e^{-b_1t}\right)\left(1+\beta e^{-b_0t}\right)}\right]}{\log\left[\left(\frac{1-e^{-b_1t}}{1+\beta e^{-b_1t}}\right)\left(\frac{1+\beta e^{-b_0t}}{1-e^{-b_0t}}\right)\right]} \quad (3.7)$$

Rejection region:

$$N(t) \geq \frac{\log\left(\frac{1-\beta}{\alpha}\right)+a\left[\frac{e^{-b_0t}-e^{-b_1t}+\beta\left(e^{-b_0t}-e^{-b_1t}\right)}{\left(1+\beta e^{-b_1t}\right)\left(1+\beta e^{-b_0t}\right)}\right]}{\log\left[\left(\frac{1-e^{-b_1t}}{1+\beta e^{-b_1t}}\right)\left(\frac{1+\beta e^{-b_0t}}{1-e^{-b_0t}}\right)\right]} \quad (3.8)$$

Continuation region:

$$\frac{\log\left(\frac{\beta}{1-\alpha}\right)+a\left[\frac{\left[e^{-b_0t}-e^{-b_1t}+\beta\left(e^{-b_0t}-e^{-b_1t}\right)\right]}{\left(1+\beta e^{-b_1t}\right)\left(1+\beta e^{-b_0t}\right)}\right]}{\log\left[\left(\frac{1-e^{-b_1t}}{1+\beta e^{-b_1t}}\right)\left(\frac{1+\beta e^{-b_0t}}{1-e^{-b_0t}}\right)\right]} < N(t) < \frac{\log\left(\frac{1-\beta}{\alpha}\right)+a\left[\frac{e^{-b_0t}-e^{-b_1t}+\beta\left(e^{-b_0t}-e^{-b_1t}\right)}{\left(1+\beta e^{-b_1t}\right)\left(1+\beta e^{-b_0t}\right)}\right]}{\log\left[\left(\frac{1-e^{-b_1t}}{1+\beta e^{-b_1t}}\right)\left(\frac{1+\beta e^{-b_0t}}{1-e^{-b_0t}}\right)\right]} \quad (3.9)$$

It may be noted that in the above two models 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 [4]. In that sense equations (3.7), (3.8), (3.9) can be regarded as generalizations to the decision procedure of [4].The applications of these results for live software failure data are presented with analysis in Section 5.

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 observed number of failures of software system or its sub system isn't' units of time. In this section we evaluate the decision rules based on the considered mean value functions for eleven different data sets of the above form, borrowed from [1] [5] [6]. Based on the estimates of the parameter a, b in each mean value function, we have chosen the specifications of b0=b-δ, b1=b+ δ equidistant on either side of estimate of b obtained through a Data Set to apply SPRT such that b0 < b < b1. Assuming the value of δ=0.000065, the choices are given in the following table

Table-1. Estimates a, b & Specifications of b_0 ,b_1

Data Set	Estimate of a	Estimate of b	b_0	b_1
DS1[7]	924.9966	0.002491	0.002426	0.002556
DS2[7]	2664.9728	0.000874	0.000809	0.000939
DS3[7]	4327.8121	0.000991	0.000926	0.001056
DS4[7]	28892.527	0.002496	0.002431	0.002561
DS5[7]	3680.6387	0.002498	0.002433	0.002563
DS6[7]	5316.8754	0.000906	0.000841	0.000971
DS7[7]	1268.6209	0.001492	0.001427	0.001557
DS8[7]	1875.0356	0.001331	0.001266	0.001396
WOOD DS1[6]	1932.1627	0.000142	0.000077	0.000207
WOOD DS2[6]	716.8684	0.000201	0.000136	0.000266
WOOD DS3[6]	643.2033	0.000129	0.000194	0.000064

Using the selected b_0 , b_1 and subsequently $m_0(t)$, $m_1(t)$ the For the model we calculated the decision rules given by Equations 3.8, 3.9, sequentially at each 't' of the data sets taking the strength (α, β) as (0.05, 0.05). These are presented for the model in Table II.

Table -II. SPRT Analysis for data sets

Dataset	T	N(T)	Accept region (<=)	Reject region (=>)	Decision
DS1	2	1	-54.279306	58.6622	Continuous
DS2	6	1	-17.553640	21.9802	Continuous
DS3	9	1	-18.346562	26.5004	Continuous
DS4	16	1	12.004326	125.172	
	40	2	80.374189	193.670	Accepted
DS5	5	1	-47.885173	65.3742	
	10	2	3.183398	-6.0189	Rejected

DS6	27	1	-15.917661	25.0714	Rejected
DS7	1	1	10.000000	1000.00	Accepted
DS8	3	1	0.100000	10.0000	Continuous
WOOD DS1	13	1	-2.735741	3.21951	Rejected
WOOD DS2	6	1	0.150466	19.0952	Rejected
	15	2	0.248277	19.1904	
	28	3	0.346065	19.2856	
WOOD DS3	1	1	-2.583435	2.72704	Rejected
	4	2	-2.511781	2.72704	

From the above table we see that decision either to accept or reject the system is reached much in advance of the last time instant of data (the testing time). The fallowing consolidated table reveals the iterations required to come to decision about the Software each data set.

6. CONCLUSION

The table II shows that the proposed Inflection S-Shaped model as exemplified for 11 data sets indicate that the model is performing well in arriving at a decision. Out of 11 data sets ,the procedure applied on the model has given a decision of rejection for 5,acceptance for 2 and continuing for 4 at various time instant of data as follows. DS5,DS6,WOOD DS1,WOOD DS2,WOOD DS3 are rejected at 2,1,1,3,2 instant of time respectively, DS1,DS2,DS3,DS8 are continuing at one instant of time respectively , DS4,DS7 are acceptence at 1&2 instant of time respectively.there fore,we may conclude that, applying SPRT on data sets we can come to an early conclusion of reliable / unreliable of software

7. REFERENCE

[1] PHAM, H. (2005). “A Generalized Logistic Software Reliability Growth models”, OPSEARCH, Vol.42 Pp.322- 331.

[2] Satya Prasad, R (2007). “Half logistic software reliability growth models “ Ph.D Thesis (awarded) of ANU INDIA.

[3] Satya Prasad, R (2010) “Half logistic software Reliability growth model S “ International journal of Advanced Research in Computer science, Vol.1, no 2, Pp.31-36.

[4] STIEBER, H.A (1997). “Statistical quality control .How To detect unreliable software components”, proceeding The 8th International Symposium on software reliability engineering, Pp.8-12.

[5] WALD (1947). “Sequential Analysis”, Wiley, Newyork.

[6] WOOD, A. (1996) “Predicting software reliability “, IEEE Computer, Pp.2253-2264.

[7] PHAM. H, (2006) “System software reliability “, Springer.

[8] Satya Prasad .R, Sangeetha. Y (2012) “SPC for software reliability using Inflection S-Shaped model”, International Journal computer application Vol. 60, no 2, Pp. 22-27.