Varshine.T.A/ Elixir Comp. Sci. & Engg. 63 (2013) 18327-18333

Awakening to reality

Available online at www.elixirpublishers.com (Elixir International Journal)

## **Computer Science and Engineering**



Elixir Comp. Sci. & Engg. 63 (2013) 18327-18333

# Detection of reliablility using SPRT: delayed s-shaped

Varshine.T.A

ABSTRACT

Computer Science Engineering, Amrita Viswa Vidhyapeetham.

### ARTICLE INFO

Article history: Received: 9 August 2013; Received in revised form: 29 September 2013; Accepted: 4 October 2013;

## Keywords

Software, Sequenial, Hypothesis, Monitor. 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 / 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 5 data sets of ungrouped using delayed S-shaped model and analyzed the results. The parameters are estimated using Maximum Likelihood Estimation method.

© 2013 Elixir All rights reserved

#### Introduction

Sequential Probability Ratio Test (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. However, in Sequential Analysis every case is analysed directly. The data collected up to 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 observe. Data collection can be terminated after few cases, decisions can be taken quickly. This leads to saving in terms of human life and finance.

In the analysis of software failure data, we often deal with Time Between Failures. 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

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

(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 for a software failure data to detect unreliable software components and compare the reliability of different software versions. In this chapter the popular SRGM – delayed S-shaped 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 considered SRGM is given in Section 5.

© 2013 Elixir All rights reserved

#### 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 ' $\lambda$ '. In our case, N(t) = number of failures up to time 't' and ' $\lambda$ ' 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 ' $\lambda$ '. We can not expect to estimate ' $\lambda$ ' precisely. But we want to reject the system with a high probability if our 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  $\lambda \geq \lambda_1$ . 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 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

$$P_{1} = \frac{e^{-\lambda_{1}t} \left[\lambda_{1}t\right]^{N(t)}}{N(t)!}$$

$$P_{0} = \frac{e^{-\lambda_{0}t} \left[\lambda_{0}t\right]^{N(t)}}{N(t)!}$$

$$(2.1)$$

$$(2.2)$$

The ratio  $\underline{P_1}$  at any time 't' is considered as a measure of deciding the truth towards  $\lambda_0$  or  $\lambda_1$ , given a sequence of time  $\overline{P_0}$ 

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 P gives

$$\frac{I_1}{P_0}^{\mathcal{B}}$$

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

The decision rule of SPRT is to decide in favor of  $\lambda_1$ , in favor 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 the test process with one more observation in failure data, according to

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

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

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

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

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

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

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

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

$$a = \frac{\lambda_{1} - \lambda_{0}}{\log\left(\frac{\lambda_{1}}{\lambda_{0}}\right)}$$
(2.8)  
$$b_{1} = \frac{\log\left[\frac{1 - \alpha}{\beta}\right]}{\log\left(\frac{\lambda_{1}}{\lambda_{0}}\right)}$$
(2.9)  
$$b_{2} = \frac{\log\left[\frac{1 - \beta}{\alpha}\right]}{\log\left(\frac{\lambda_{1}}{\lambda_{0}}\right)}$$
(2.10)

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

$$\lambda_1 = q \frac{\lambda \log(q)}{q-1}$$
 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. Sequential Test for Software Reliability Growth Models

In Section 2, for the Poisson process we know 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 m(t) as its mean value function the probability equation of a such a process is

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

#### Varshine.T.A/ Elixir Comp. Sci. & Engg. 63 (2013) 18327-18333

Depending on the forms of m(t) we get various Poisson processes called NHPP. For our delayed S-shaped model, the mean value function is given as  $m(t) = a(1-(1+bt)e^{-bt})$  where a > 0, b > 0 and 'b' is the error detection rate per error in the steady state. This model is called delayed S-shaped NHPP model for such an error detection process, in which the observed growth curve of the cumulative number of detected errors is S-shaped (Yamada et. al., 1984). We may write

$$P_{1} = \frac{e^{-m_{1}(t)} \cdot [m_{1}(t)]^{N(t)}}{N(t)!}$$
$$P_{0} = \frac{e^{-m_{0}(t)} \cdot [m_{0}(t)]^{N(t)}}{N(t)!}$$

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,  $\frac{P_1}{P_0} \le B$ 

i.e.,  $\frac{e^{-m_{1}(t)} \cdot [m_{1}(t)]^{N(t)}}{e^{-m_{0}(t)} \cdot [m_{0}(t)]^{N(t)}} \le B$ 

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,  $\frac{P_1}{P_0} \ge A$ 

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

Continue the test procedure as long as

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

Substituting the appropriate expressions of the respective mean value function -m(t) of delayed S-shaped, we get the respective decision rules and are given in following lines

Acceptance region:

i

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

Rejection region:

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

Continuation region:

$$\frac{\log\left(\frac{1-\beta}{\alpha}\right) + a\left[(1+b_{0}t)e^{-b_{0}t} - (1+b_{1}t)e^{-b_{1}t}\right]}{\log\left[\frac{1-(1+b_{1}t)e^{-b_{0}t}}{1-(1+b_{0}t)e^{-b_{0}t}}\right]} < N(t) < \frac{\log\left(\frac{\beta}{1-\alpha}\right) + a\left[(1+b_{0}t)e^{-b_{0}t} - (1+b_{1}t)e^{-b_{1}t}\right]}{\log\left[\frac{1-(1+b_{1}t)e^{-b_{0}t}}{1-(1+b_{0}t)e^{-b_{0}t}}\right]}$$
(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.

#### 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 (i.e ungrouped) and interval domain data (i.e grouped).

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 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. If we conduct an experiment and obtain N independent observations,  $t_1, t_2, \ldots, t_N$ . Then the likelihood function is given by (Pham, 2006) the following product:

$$L(t_1, t_2, \dots, t_N \mid \theta_1, \theta_2, \dots, \theta_k) = L = \prod_{i=1}^N f(t_i; \theta_1, \theta_2, \dots, \theta_k)$$

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

The logarithmic likelihood function is given by:

$$\ln L = \sum_{i=1}^{N} \ln f(t_i; \theta_1, \theta_2, \dots, \theta_k) \qquad LogL = Log\left(\prod_{i=1}^{n} \lambda(t_i)\right)$$

Assuming that the data given is the cumulative number of Time Between Failures (i.e Time domain / Ungrouped), the log likelihood function takes on the following form.  $\sum_{i=1}^{n} \log \left[ \lambda(t_i) \right] - m(t_n)$ (4.1)

The maximum likelihood estimators (MLE) of 
$$\theta_1, \theta_2, \dots, \theta_k$$
 are obtained by maximizing L or  $\Lambda$ , where  $\Lambda$  is  $\ln L$ . By maximizing  $\Lambda$ , which is much easier to work with than L, the maximum likelihood estimators (MLE) of  $\theta_1, \theta_2, \dots, \theta_k$  are the simultaneous solutions of k equations such that:  

$$\frac{\partial(\Lambda)}{\partial \theta_i} = 0^{,j=1,2,\dots,k}$$

$$\frac{\partial \log L}{\partial a} = 0^{\dagger} \qquad g(b) = \frac{\partial \log L}{\partial b} = 0^{\dagger} \qquad g'(b) = \frac{\partial^2 \log L}{\partial b^2} = 0$$

The parameters 'a' and 'b' are estimated using iterative Newton Raphson Method, which is given as,  $x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)}$ 

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

$$g(b) = \frac{2n}{b} - \sum_{i=1}^{n} t_i - \frac{nbt_n^2 e^{-bt_n}}{\left[1 - (1 + bt_n)e^{-bt_n}\right]}$$
(4.3)
$$2m \cdot (1 - e^{-bt_n} - bt_n)nt^2 e^{-bt_n}$$
(4.4)

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

#### SPRT Analysis of ungrouped Data

In this section we evaluate the decision rules based on the considered mean value function for FIVE different data sets, borrowed from Pham (2006), Xie (2002) and SONATA (Ashoka, 2010) software services. 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  $\delta = 0.004$ , the choices are given in the following table.

Table 5.1: Estimates of a, b & Specifications of  $b_0$ ,  $b_1$  for ungrouped data

Data Set	Estimate of 'a'	Estimate of 'b'	b <sub>0</sub>	<b>b</b> <sub>1</sub>
1	30.293717	0.009035	0.005035	0.013035
2	34.926746	0.001884	-0.002116	0.005884
3	26.769244	0.021628	0.017628	0.025628
4	15.484667	0.017924	0.013924	0.021924
5	22.253256	0.009543	0.005543	0.013543

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.

Тε	abl	e 5	5.2	2:	SP	РRТ	ana	lysis	for	5	data	sets	of	ungrou	ped	data

Data Set	Т	N(t)	Acceptance region (≤)	Rejection Region (≥)	Decision	
	30.02	1	0.026797	2.507130		
	31.46	2	1.299420	3.790187		
1	53.93	3	1.477420	4.139399	Rejection	
1	55.29	4	1.709327	4.382196		
	58.72	5	2.609367	5.309976		
	71.92	6	3.070112	5.881298		
2	52.5	1	1.077386	3.527223	Acceptance	
	9	1	-0.37989	5.796440		
	21	2	1.25401	8.012783		
	32	3	2.01087	9.376087		
	36	4	3.16368	10.768744		
3	43	5	3.65445	11.706527	Continue	
	45	6	4.43522	12.621776		
	50	7	5.64211	14.178827	Conunue	
	58	8	6.59510	15.736469		
	63	9	7.52864	17.077987		
	70	10	8.09289	18.255929		
	71	11	8.56826	18.823258		
	77	12	9.10569	19.936479		

18332

	78	13	9.52827	20.459176		
	87	14	10.42669	22.316060		
	91	15	10.83845	23.189186		
	92	16	10.94345	23.413148		
	95	17	11.23429	24.069925		
	98	18	11.41747	24.633026		
	104	19	12.07487	26.094660		
	105	20	11.60198	25.761835		
	116	21	9.61279	25.437009		
	149	22	12.49613	35.008313		
	156	23	-0.36822	23.978441		
	247	24	-1.76182	72.705175		
	249	25	-2.13020	74.328763		
	250	26	-27.87521	49.601618		
	10	1	-1.158060	3.904705		
	19	2	-0.352824	4.996781		
	32	3	0.437638	6.245045		
	43	4	1.528641	7.768524		
	58	5	2.423944	9.330167		
	70	6	3.421917	10.934184		
	88	7	4.252168	12.816797		
4	103	8	4.760696	14.357293	Continue	
	125	9	4.980364	16.403251		
	150	10	5.152370	19.221257		
	169	11	3.984377	20.585023		
	199	12	2.350313	24.165604		
	231	13	0.471426	30.104967		
	256	14	-3.932532	34.081659		
	296	15	-20.697571	36.829605		
5	5.5	1	-0.838349	1.625846	Dejection	
	7.33	2	-0.802263	1.675552	кејесион	

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

#### Conclusion

The table 5.2 of Time domain data as exemplified for 5 Data Sets shows that Delayed S-shaped model is performing well in arriving at a decision. Out of 5 Data Sets of Time domain the procedure applied on the model has given a decision of rejection for 2, acceptance for 1 and continue for 2 at various time instant of the data as follows. Data Set #1 and #5 are rejected at  $6^{th}$  and  $2^{nd}$  instant of time. Data Set #2 is accepted at  $1^{st}$  instant of time. Data Set #3 and #4 are continuing. Therefore, we may conclude that, applying SPRT on data sets we can come to an early conclusion of reliable / unreliable of software.

#### References

- [1] Ashoka. M., (2010). "Sonata software limited Data Set", Bangalore
- [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] Stieber, H.A. (1997). "Statistical Quality Control: How To Detect Unreliable Software Components", Proceedings the 8th International Symposium on Software Reliability Engineering, 8-12.
- [5] Wald. A., 1947. "Sequential Analysis", John Wiley and Son, Inc, New York.
- [6] Xie, M., Goh. T.N., Ranjan.P., "Some effective control chart procedures for reliability monitoring" -Reliability engineering and System Safety 77, 143 -150, 2002.
- [7] Yamada, S., Ohba, M. and Osaki, S., (1984). "S-shaped software reliability growth models and their applications," IEEE Transactions on Reliability, Vol. R-33.