DETECTION OF RELIABLE SOFTWARE USING SPRT ON TIME DOMAIN DATA

G.Krishna Mohan[1] and Dr. Satya Prasad Ravi[2]

[1]Reader, Dept. of Computer Science, P.B.Siddhartha college
Vijayawada, Andhrapradesh, India.
km_mm_2000@yahoo.com
[2]Associate Professor, Dept. of Computer Science & Engg., Acharya Nagrjuna University,
Nagarjuna Nagar, Guntur, Andhrapradesh, India
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 Time 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, Time 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 analysed 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

DOI : 10.5121/ijcsea.2011.1407

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

Stieber[5] 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 [4] 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 Exponential imperfect debugging 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 $t_1 < t_2 < t_3 < .... < t_K$ 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 [3] the test process with one more observation in failure data, according as

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

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

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

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 ‘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 $N_U(t) = a.t + b_2$ (2.6)

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

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

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

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

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

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

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

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

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 Weibull 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

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

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

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

Decide the system to be unreliable and reject if

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

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

Continue the test procedure as long as

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

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

Acceptance region:

$$N(t) \le \frac{\log\left(\dfrac{\beta}{1-\alpha}\right)+a\left(e^{-(b_0t)^2}-e^{-(b_1t)^2}\right)}{\log\left(\dfrac{1-e^{-(b_1t)^2}}{1-e^{-(b_0t)^2}}\right)} \quad (3.4)$$

Rejection region:

$$N(t) \ge \frac{\log\left(\dfrac{1-\beta}{\alpha}\right)+a\left(e^{-(b_0t)^2}-e^{-(b_1t)^2}\right)}{\log\left(\dfrac{1-e^{-(b_1t)^2}}{1-e^{-(b_0t)^2}}\right)} \quad (3.5)$$

Continuation region:

$$\frac{\log\left(\dfrac{\beta}{1-\alpha}\right)+a\left(e^{-(b_0t)^2}-e^{-(b_1t)^2}\right)}{\log\left(\dfrac{1-e^{-(b_1t)^2}}{1-e^{-(b_0t)^2}}\right)} < N\left(t\right) < \frac{\log\left(\dfrac{1-\beta}{\alpha}\right)+a\left(e^{-(b_0t)^2}-e^{-(b_1t)^2}\right)}{\log\left(\dfrac{1-e^{-(b_1t)^2}}{1-e^{-(b_0t)^2}}\right)} \quad (3.6)$$

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 (1997). The applications of these results for live software failure data are presented with analysis in Section 5.

4. ML (MAXIMUM LIKELIHOOD) PARAMETER ESTIMATION

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. If we conduct an experiment and obtain N independent observations, $t_1, t_2, \ldots, t_N$. Then the likelihood function is given by[2] the following product:

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

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

The logarithmic likelihood function is given by: $\Lambda = \ln L = \sum_{i=1}^{N} \ln f(t_i; \theta_1, \theta_2, \ldots, \theta_k)$

Log L = log ($\prod_{i=1}^{n} \lambda(t_i)$)

which can be written as $\sum_{i=1}^{n} \log[\lambda(t_i)] - m(t_n)$

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

For the present model of Weibull, the parameters are estimated from [9].

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 [2][7][8] and SONATA 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 $\delta = 0.0025$, the choices 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**
Xie	30.05159	0.003416	0.000916	0.005916
AT&T	23.71966	0.004824	0.002324	0.007324

IBM	19.16436	0.00711	0.00461	0.00961
Lyu	24.086392	0.033562	0.031062	0.036062
NTDS	28.85193	0.011827	0.009327	0.014327
SONATA	31.961497	0.000912	-0.001588	0.003412
TROPICO-R	33.966286	0.004934	0.002434	0.007434

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 Table 2.

Table 2: SPRT analysis for 7 data sets

Data Set	**T**	**N(t)**	**R.H.S of equation (5.3.10) Acceptance region (≤)**	**R.H.S of Equation (5.3.11) Rejection Region(≥)**	**Decision**
Xie	30.02	1	-0.17291773	0.98310581	***Rejection***
AT & T	5.5	1	-0.66343858	1.22238877	***Rejection***
	7.33	2	-0.6516082	1.23375403	
IBM	10	1	-0.96601265	1.97464106	***Rejection***
	19	2	-0.72618077	2.19598203	
	32	3	-0.16030739	2.71551971	
Lyu	0.5	1	-5.2122577	9.29358823	***Rejection***
	1.7	2	-5.13340617	9.35093377	
	4.5	3	-4.63088682	9.71308540	
	7.2	4	-3.78455807	10.308599	
	10	5	-2.64978817	11.0721445	
	13	6	-1.31134889	11.9012488	
	14.8	7	-0.52315658	12.335913	
	15.7	8	-0.14781485	12.5227584	
	17.1	9	0.39937518	12.7629428	
	20.6	10	1.50499336	13.041212	
	24	11	2.14640728	12.8236973	
	25.2	12	2.26736443	12.6344422	
	26.1	13	2.32374036	12.4570439	
NTDS	9	1	-1.48853384	3.52816085	***Rejection***
	21	2	-0.16621278	4.72824732	
	32	3	1.57766914	6.2822846	
	36	4	2.2450846	6.86442913	
	43	5	3.35860135	7.81077962	
	45	6	3.65218168	8.05284578	
	50	7	4.31762368	8.58317713	
	58	8	5.13176726	9.16554443	
	63	9	5.46268271	9.34426677	
	70	10	5.6895861	9.35204023	
SONATA	52.5	1	-0.49942268	2.30862835	***Acceptance***
	105	2	0.89550118	3.63547513	
	131.25	3	1.7959382	4.48617512	
	183.75	4	3.68180303	6.24476432	
	201.25	5	4.26792637	6.78119865	
	306.25	6	6.45927614	8.62012162	

TROPICO-R 1500	7	1	-0.6606864	1.27763909	***Rejection***
	8	2	-0.64936629	1.28863802	

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(the testing time).

6. CONCLUSION

The table 2 shows that Weibull model as exemplified for 7 Data Sets indicate that the model is performing well in arriving at a decision. Out of 7 Data Sets the procedure applied on the model has given a decision of rejection for 6, acceptance for 1 and continue for none at various time instant of the data as follows. DS1, DS2, DS3, DS4, DS5, DS7 are rejected at 1st , 2nd , 3rd ,13th ,10th and 2nd instant of time respectively. DS6 is accepted at 2nd 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.

7. REFERENCES

[1] GOEL, A.L and OKUMOTO, K. "*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., "System software reliability", Springer. 2006.

[3] Satya Prasad, R., "Half logistic Software reliability growth model ", 2007, Ph.D Thesis of ANU, India.

[4] Wald. A., "Sequential Analysis", John Wiley and Son, Inc, New York. 1947.

[5] STIEBER, H.A. "*Statistical Quality Control: How To Detect Unreliable Software Components"*, Proceedings the 8th International Symposium on Software Reliability Engineering, 8-12. 1997.

[6] WOOD, A. "*Predicting Software Reliability"*, IEEE Computer, 2253-2264. 1996.

[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.

[8] Michael. R. Lyu, "The hand book of software reliability engineering", McGrawHill & IEEE Computer Society press.

[9] Dr R.Satya Prasad, G.Krishna Mohan and Prof R R L Kantham. Article: Time Domain based Software Process Control using Weibull Mean Value Function. International Journal of Computer Applications 18(3):18-21, March 2011. .

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.