

Detection of Reliable Software Using SPRT

Dr. R.Satya Prasad

Dept. of CS & Engg.
Acharya Nagarjuna University
Guntur, Andhrapradesh,
India

N. Supriya

Dept. of Computer Science, Adikavi
Nannaya University Rajahmedndry,
Andhrapradesh,
India

G.Krishna Mohan

Dept. of Computer Science
P.B.Siddhartha College, Vijayawada,
Andhrapradesh,
India

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 have proposed the performance of SPRT on Time domain data using exponential imperfect debugging model and analyzed the results by applying on 5 data sets. The parameters are estimated by using Maximum Likelihood Estimation.

Keywords- Exponential imperfect debugging; Sequential Probability Ratio Test; Maximum Likelihood Estimation; Decision lines; Software Reliability; Software failure data.

I. INTRODUCTION

Wald's procedure is particularly relevant if the data is collected sequentially. Sequential Analysis is different from that of Classical Hypothesis Testing where 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 the data is collected the analysis is done, conclusions are drawn. However, in Sequential Analysis every case is analyzed directly after being collected, 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 is easily seen. 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

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

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. The extension of this theory to the SRGM – Exponential imperfect debugging is presented in Section 3. The Maximum Likelihood parameter estimation method is presented in Section 4. and Application of the decision rule to detect unreliable software components with respect to the proposed SRGM is given in Section 5.

II. 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. Suppose if 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 if we want to reject the system with a high probability, 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

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

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

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 $t_1 < t_2 < t_3 < \dots < t_K$ 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 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

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

$$\frac{P_1}{P_0} \leq B \quad (2.4) \quad B < \frac{P_1}{P_0} < A$$

(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

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

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

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

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

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

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

$$\lambda_0 = \frac{\lambda \cdot \log(q)}{q-1}, \lambda_1 = q \cdot \frac{\lambda \cdot \log(q)}{q-1}, \text{ where } q = \frac{\lambda_1}{\lambda_0}$$

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).

III. 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 (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 $m(t)$ we get various Poisson processes called NHPP. For our model the mean value function

is given as $m(t) = \frac{a}{1-c} (1 - e^{-(1-c)bt})$ where $a > 0, b > 0$

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 $(b_0 < b_1)$ respectively. 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 $\frac{P_1}{P_0} \leq B$

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

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

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 the considered model we get the respective decision rules and are given in followings lines

Acceptance region:

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

Rejection region:

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

Continuation region:

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

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, $m(t) = \lambda t$ the

decision rules become decision lines as described by Stieber (1997). In that case 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.

IV. 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 in their approach and can be applied to many models and also to different types of data. Although the methodology for maximum likelihood estimation is simple, the implementation is mathematically complex. 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, \dots, t_N . Then the likelihood function is given by [9] the following product:

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

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

The logarithmic likelihood function is given by:

$$\Lambda = \ln L = \sum_{i=1}^N \ln f(t_i; \theta_1, \theta_2, \dots, \theta_k)$$

$\log L = \log \left(\prod_{i=1}^n \lambda(t_i) \right)$; which can be written as

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

The maximum likelihood estimators (MLE) of $\theta_1, \theta_2, \dots, \theta_k$ 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 $\theta_1, \theta_2, \dots, \theta_k$ are the simultaneous solutions of k equations such that: $\frac{\partial(\Lambda)}{\partial \theta_j} = 0, \quad j=1, 2, \dots, k$

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 Exponential imperfect debugging at $c=0.05$, the parameters are estimated from [10].

V. SPRT ANALYSIS OF 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] with the assumption of $c=0.05$. 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 I. ESTIMATES OF a,b & SPECIFICATIONS OF b_0, b_1

Data Set	Estimate of 'a'	Estimate of 'b'	b_0	b_1
DS 1 [7]	31.738136	0.003253	0.000753	0.005753
DS 2 [2]	24.182003	0.003091	0.000591	0.005591
DS 3 [2]	22.286839	0.003627	0.001127	0.006127
DS 4 [2]	32.293828	0.006095	0.003595	0.008595
DS 5 [8]	30.276648	0.020823	0.018323	0.023323

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 II.

TABLE II. SPRT ANALYSIS FOR 5 DATA SETS

Data Set	T	N(t)	R.H.S of equation (5.3.10) Acceptance region (\leq)	R.H.S of Equation (5.3.11) Rejection Region (\geq)	Decision
DS 1	30.02	1	1.419789011	3.625727114	Accept
DS 2	5.5	1	-0.404541835	1.53395301	Reject
	7.33	2	-0.309766769	1.632482881	
DS 3	10	1	-0.288370452	2.305609383	Reject
	19	2	0.258176419	2.885230322	
	32	3	1.012360023	3.688059978	
	43	4	1.619570979	4.337249402	
	58	5	2.404532289	5.180679581	
	70	6	2.998756638	5.822712830	
DS 4	9	1	-0.210017229	4.882283497	Accept
	21	2	1.757136003	7.020723877	
	32	3	3.432506995	8.859613125	
DS 5	0.5	1	-6.177485603	11.85991989	Continue
	1.7	2	-5.518244758	12.73424974	
	4.5	3	-4.051626961	14.71612777	
	7.2	4	-2.728688612	16.55426597	
	10	5	-1.446988791	18.39005947	
	13	6	-0.17086726	20.28319498	
	14.8	7	0.548597293	21.38490073	
	15.7	8	0.89573932	21.926644	
	17.1	9	1.419425808	22.75775024	
	20.6	10	2.64443639	24.77711192	
	24	11	3.724217517	26.66533205	
	25.2	12	4.08045555	27.31591129	
	26.1	13	4.339358286	27.79874025	
	27.8	14	4.809408109	28.69931804	
	29.2	15	5.178228348	29.43026	
	31.9	16	5.844193721	30.81447884	
	35.1	17	6.558795727	32.4155083	
	37.6	18	7.06283825	33.63975553	
	39.6	19	7.432965441	34.60421986	
	44.1	20	8.162162302	36.73248646	
47.6	21	8.634040324	38.35477082		
52.8	22	9.188544561	40.72597288		
60	23	9.681798744	43.96479429		
70.7	24	9.861407038	48.78058702		

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).The following consolidated table reveals the iterations required to come to a decision about the software of each data set.

VI. CONCLUSION

The table II shows that The Exponential imperfect debugging 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 2, acceptance for 2 and continue for 1 at various time instant of the data as follows. DS1, DS4 are accepted at 1st and 3rd instant of time respectively, DS2, DS3 are rejected at 2nd and 6th instant of time respectively. DS5 is continuing. Therefore, we may conclude that, applying SPRT on data sets we can come to an early conclusion of reliable / unreliable of software.

REFERENCES

- 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.
- Pham. H., "System software reliability", Springer. 2006.
- Satya Prasad, R., "Half logistic Software reliability growth model ", 2007, Ph.D Thesis of ANU, India.
- Wald. A., "Sequential Analysis", John Wiley and Son, Inc, New York. 1947.
- Stieber, H.A. "Statistical Quality Control: How To Detect Unreliable Software Components", Proceedings the 8th International Symposium on Software Reliability Engineering, 8-12. 1997.
- Wood, A. "Predicting Software Reliability", IEEE Computer, 2253-2264. 1996.
- Xie, M., Goh. T.N., Ranjan.P., "Some effective control chart procedures for reliability monitoring" -Reliability engineering and System Safety 77 143 -150, 2002.
- Michael. R. Lyu, "The hand book of software reliability engineering", McGrawHill & IEEE Computer Society press.
- Pham. H., 2006. "System software reliability", Springer.
- Satya Prasad. R, Supriya. N and Krishna Mohan. G, "spc for software reliability: imperfect software debugging model". International Journal of Computer Science Issues. Vol.8, Issue.3, No.2.

AUTHORS PROFILE



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.



Mrs. N. Supriya Working as Academic Consultant, in the department of Computer Science, Adikavi Nannaya university, Rajahmundry. She is at present pursuing Ph.D at Acharya Nagarjuna University. Her research interests lies in Software Engineering and Data Mining.



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.