Duane Based SPRT: Regression Approach

Mr.N.V.K.Stanley Raju[1], Dr. R. Satya Prasad[2]
[1]Research Scholar, [2]Associate Professor
Dept. of CS&E, Rayalaseema University[1]
Dept. of Computer Science & Engg., Acharya Nagrjuna University[2]
Nagarjuna Nagar, Guntur, Andhrapradesh, India.

Abstract - Classical Hypothesis testing needs more time to draw conclusions by collecting volumes of data. But, to decide upon the reliability or unreliability of the developed software very quickly Sequential Analysis of Statistical science could be adopted. 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. The Regression method is used to derive the unknown parameters. In this paper a control mechanism based on sequential probability Ratio Test is applied using mean value function of Duane model and analyzed the results.

I. INTRODUCTION

Sequential Probability Ratio Test (SPRT) is an ongoing statistical analysis repeatedly conducted as data is collected. It is used in anomaly detection and decision making for electronics, structures and process controls. The data are repeatedly reassessed and a decision is made to Reject the null hypothesis and stop collecting data, Fail to reject the null hypothesis and stop collecting data or Continue collecting data until a decision regarding the null hypothesis can be reached.

The SPRT sets threshold boundaries, which take the form of parallel lines, one of which represents the expected outcome and the other a significantly different outcome. When the value of the calculated test statistic falls outside of these threshold boundaries, a conclusion can be drawn and data collection stops. The parameters are estimated using Regression approach. In the present paper, the Duane model is applied on six sets of existing software reliability data and analyzed the results.

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 analyzed 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

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

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 (SPRT) of Wald (1947) 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 Duane model and adopt the principle of Stieber (1997) in detecting unreliable software components in order to accept or reject the developed software. The theory proposed by Stieber (1997) is presented in Section 2 for a ready reference. Extension of this theory to the SRGM – Duane model is presented in Section 3. Application of the decision rule to detect unreliable software components with respect to the proposed SRGM is given in Section 4. Analysis of the application of the SPRT on six data sets and conclusions drawn are given in Section 5.

II. 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 (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 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 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

$$Q_1 = \frac{e^{-\lambda_1 t}\left[\lambda_1 t\right]^{N(t)}}{N(t)!} \tag{2.1}$$

$$Q_0 = \frac{e^{-\lambda_0 t}\left[\lambda_0 t\right]^{N(t)}}{N(t)!} \tag{2.2}$$

The ratio $\frac{Q_1}{Q_0}$ 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 $\frac{Q_1}{Q_0}$ gives $\frac{Q_1}{Q_0} = \exp(\lambda_0 - \lambda_1)t + \left(\frac{\lambda_1}{\lambda_0}\right)^{N(t)}$

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 $\frac{Q_1}{Q_0}$ 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

$$\frac{Q_1}{Q_0} \geq A \tag{2.3}$$

$$\frac{Q_1}{Q_0} \leq B \tag{2.4}$$

$$B < \frac{Q_1}{Q_0} < A \tag{2.5}$$

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

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

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

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

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 α, β, λ_0 and λ_1 can be chosen in several ways. One way suggested by Stieber (1997) is

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

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 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 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 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 the Duane model the mean value function is given as $m\left(t\right)=at^{b}$ where $a>0,b>0$

We may write

$$Q_1=\frac{e^{-m_1(t)}.\left[m_1(t)\right]^{N(t)}}{N(t)!}$$

$$Q_0=\frac{e^{-m_0(t)}.\left[m_0(t)\right]^{N(t)}}{N(t)!}$$

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 $\left(b_0<b_1\right)$ respectively. It can be shown that for our model $m\left(t\right)$ at b_1 is greater than that at b_0 . Symbolically $m_0\left(t\right)<m_1\left(t\right)$. Then the SPRT procedure is as follows:

Accept the system to be reliable, if $\frac{Q_1}{Q_0}\leq B$

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

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

Decide the system to be unreliable and reject if

$$\frac{Q_1}{Q_0}\geq A$$

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

Continue the test procedure as long as

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

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

Acceptance region:

$$N(t)\leq\frac{\log\left(\frac{\beta}{1-\alpha}\right)+a\left[t^{b_1}-t^{b_0}\right]}{\log\left(\frac{t^{b_1}}{t^{b_0}}\right)} \qquad (3.4)$$

Rejection region:

$$N(t)\geq\frac{\log\left(\frac{1-\beta}{\alpha}\right)+a\left[t^{b_1}-t^{b_0}\right]}{\log\left(\frac{t^{b_1}}{t^{b_0}}\right)} \qquad (3.5)$$

Continuation region:

$$\frac{\log\left(\frac{\beta}{1-\alpha}\right)+a\left[t^{b_1}-t^{b_0}\right]}{\log\left(\frac{t^{b_1}}{t^{b_0}}\right)}<N\left(t\right)<\frac{\log\left(\frac{1-\beta}{\alpha}\right)+a\left[t^{b_1}-t^{b_0}\right]}{\log\left(\frac{t^{b_1}}{t^{b_0}}\right)} \qquad (3.6)$$

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 4.

IV. SPRT ANALYSIS OF LIVE DATA SETS

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 Six different data sets of the above form, borrowed from (Xie, 2002), (Pham, 2006) and (SONATA, 2010). 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 $\delta=0.025$, the choices are given in the following table.

Table 4.1: Estimates of a, b & Specifications of b_0, b_1 for Time domain

Data Set	No. of samples	Estimated Parameters		b_0	b_1
		a	b		
1 (AT&T)	22	7.837651	0.208889	0.183889	0.233889
2 (IBM)	15	0.648328	0.572320	0.547320	0.597320
3 (LYU)	24	12.246815	0.421371	0.396371	0.446371
4 (NTDS)	26	7.286835	0.289831	0.264831	0.314831
5 (SONATA)	30	0.313930	0.616067	0.591067	0.641067
6 (Xie)	30	10.248983	0.214795	0.189795	0.239795

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 (α, β) as (0.05, 0.2). These are presented for the model in Table 4.2. The following consolidated table reveals the iterations required to come to a decision about the software of each Data Set.

Table 4.2: SPRT analysis for 6 data sets of Time domain data

Data Set	T	N(t)	Acceptance region (≤)	Rejection Region (≥)	Decision
AT&T	5.5	1	-7.086426	43.721472	Accept
	7.33	2	-3.757109	39.724684	
	10.08	3	-0.780309	36.706239	
	80.97	4	12.572664	32.284342	
IBM	10	1	-11.110866	26.505406	Continue
	19	2	-7.083844	22.332513	
	32	3	-4.273628	20.718082	
	43	4	-2.696782	20.331684	
	58	5	-1.040599	20.290736	
	70	6	0.054160	20.441299	
	88	7	1.464815	20.809946	
	103	8	2.496692	21.184876	
	125	9	3.848535	21.787441	
	150	10	5.218662	22.504827	
	169	11	6.172779	23.057063	
	199	12	7.563357	23.926418	
	231	13	8.925378	24.840119	
	256	14	9.920819	25.540638	
	296	15	11.413511	26.634811	
Lyu	0.5	1	54.103885	- 70.854666	Accept
NTDS	9	1	-0.400395	39.019639	Accept
	21	2	7.391272	35.840616	
SONATA	52.5	1	-4.259726	17.608175	Accept
	105	2	-1.162490	17.448469	
	131.25	3	-0.039255	17.720192	
	183.75	4	1.838615	18.451908	
	201.25	5	2.392493	18.720871	
	306.25	6	5.268948	20.399729	
	411.25	7	7.673696	22.063424	
Xie	30.02	1	12.147559	37.608504	Accept

From the above table, a decision of either to accept, reject the system or continue is reached much in advance of the last time instant of the data.

V. CONCLUSION

The above consolidated table of Duane model as exemplified for six Data Sets indicates that the model is performing well in arriving at a decision. The model has given a decision of acceptance for 5 Data Sets i.e AT & T at 4th instance, Lyu at 1st instance, NTDS at 2nd instance, SONATA at 7th instance and Xie at 1st instance and a decision of continue for 1 Data Set i.e IBM. Therefore, we may conclude that, applying SPRT on data sets we can come to an early conclusion of reliability / unreliability of software.

VI. REFERENCES

[1]. Krishna Mohan. G and Satya Prasad. R. (2011). "Detection of Reliable Software using SPRT on Time Domain data", International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol. 1, No. 4, pp[92-99].

[2]. Lyu, M.R. (1996). "The hand book of software reliability engineering", McGrawHill & IEEE Computer Society press.

[3]. Pham. H., (2006). "System software reliability", Springer.

[4]. Satya Prasad, R and Stanley Raju, N. V.K. (2015). "Regression Based Software Reliability Estimation: Duane Model", International Journal of Computer Trends and Technology, Volume 26, Issue 1.

[5]. Satyaprasad (2007).”Half logistic Software reliability growth model “Ph.D Thesis of ANU, India.

[6]. Stieber, H.A. (1997). “Statistical Quality Control: How To Detect Unreliable Software Components”, Proceedings the 8th International Symposium on Software Reliability Engineering, 8-12.

[7]. Wald. A., (1945). “Sequential tests of statistical hypotheses”. Annals of Mathematical Statistics, 16:117–186.

[8]. Wald. A., (1947). “Sequential Analysis”, John Wiley and Son, Inc, New York.

[9]. Xie, M., Goh. T.N., Ranjan.P. (2002). “Some effective control chart procedures for reliability monitoring” -Reliability engineering and System Safety 77 143 -150.