Software Reliability using SPRT: Inflection S-shaped Model

Dr. R. Satya Prasad1, K. Prasada Rao2 and G. Krishna Mohan3

1Associate Professor, Dept. of Computer Science & Enng., Acharya Nagarjuna University
Nagarjuna Nagar, Guntur, Andhra Pradesh, India. 91-9848487478.

2Professor & Director, Dept. of MCA, CMRIT, Kundalahalli, Bangalore.

3Reader, Dept. of Computer Science, P.B. Siddhartha College
Vijayawada, Andhra Pradesh, India. 91-9440446847.

Abstract

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 Time domain data and analyzed the results. The parameters are estimated using Maximum Likelihood Estimation method.

Keywords: Inflection S-shaped model, Maximum Likelihood Estimation, SPRT, Software testing, Software failure data.

1. INTRODUCTION

Wald’s procedure is particularly relevant if the data is collected sequentially. Sequential Analysis is different from 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 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 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 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

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

(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 (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 model Inflection S-shaped 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 – Inflection S-shaped 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 5 data sets and conclusions drawn are given in Section 5 and 6 respectively.
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 (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’s 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. 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

\[
Q_1 = e^{-\lambda_1 t} \left[ \lambda_1 t \right]^{N(t)}
\]

\[
Q_0 = e^{-\lambda_0 t} \left[ \lambda_0 t \right]^{N(t)}
\]

The ratio \( \frac{Q_1}{Q_0} \) at any time ‘\( t ' \)’ is considered as a measure of deciding the truth towards \( \lambda_0 \) or \( \lambda_1 \), given a sequence of time instants say \( t_1 < t_2 < t_3 < \ldots \ldots < t_K \) and the corresponding realizations \( N(t_1), N(t_2), \ldots \ldots N(t_K) \) 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 \( \lambda_1 \), in favor of \( \lambda_0 \) or to continue by observing the number of failures at a later time than ‘\( t ' \)’ according as \( \frac{Q_1}{Q_0} \) 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

\[
\frac{Q_1}{Q_0} \geq A
\]

\[
\frac{Q_1}{Q_0} \leq B
\]

\[
B < \frac{Q_1}{Q_0} < A
\]

The approximate values of the constants A and B are taken as \( A \approx \frac{1 - \beta}{\alpha} \), \( B \approx \frac{\beta}{1 - \alpha} \)

Where ‘\( \alpha ' \)’ 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

\[
N(t) = at + b_2
\]

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

\[
N_L(t) = a t - b_1
\]

The parameters \(\alpha, \beta, \hat{\lambda}_0\) and \(\hat{\lambda}_1\) can be chosen in several ways. One way suggested by Stieber (1997) is

\[
\hat{\lambda}_0 = \lambda_0 \log(q), \quad \hat{\lambda}_1 = q \frac{\lambda_1 \log(q)}{q-1}
\]

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.

### 3. INFLECTION S-SHAPED MODEL

Software reliability growth models (SRGM’s) are useful to assess the reliability for quality management and testing-progress control of software development. They have been grouped into two classes of models concave and S-shaped.

The most important thing about both models is that they have the same asymptotic behavior, i.e., the defect detection rate decreases as the number of defects detected (and repaired) increases, and the total number of defects detected asymptotically approaches a finite value. The inflection S-shaped model was proposed by Ohba in 1984. This model assumes that the fault detection rate increases throughout a test period. The model has a parameter, called the inflection rate, that indicates the ratio of detectable faults to the total number of faults in the target software.

True, sustained exponential growth cannot exist in the real world. Eventually all exponential, amplifying processes will uncover underlying stabilizing processes that act as limits to growth. The shift from exponential to asymptotic growth is known as sigmoidal, or S-shaped, growth.

Ohba models the dependency of faults by postulating the following assumptions:

- Some of the faults are not detectable before some other faults are removed.
- The detection rate is proportional to the number of detectable faults in the program.
- Failure rate of each detectable fault is constant and identical.
- All faults can be removed.

Assuming [Ohba 1984b]:

\[
b(t) = \frac{b}{1 + ce^{-bt}}
\]

This model is characterized by the following mean value function:

\[
m(t) = \frac{a}{1 + ce^{-bt}} \left(1 - e^{-bt}\right)
\]

Where ‘\(b\)’ is the failure detection rate, and ‘\(c\)’ is the inflection factor. The failure intensity function is given as:
\( \lambda(t) = \frac{abe^{-bt}}{(1+ce^{-bt})^2} \).

4. 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 such a process is

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

Depending on the forms of \( m(t) \) we get various Poisson processes called NHPP. For the Inflection S-shaped model the mean value function is given as

\[
m(t) = \frac{a}{1 + ce^{-bt}}
\]

where \( a > 0, b > 0 \)

We may write

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

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

Where, \( m_1(t) \) and \( m_0(t) \) are values of the mean value function at specified sets of its parameters indicating reliable software and unreliable software respectively. Let \( Q_0, Q_1 \) 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 \( m_1(t) < m_0(t) \). Then the SPRT procedure is as follows:

Accept the system to be reliable

\[
\frac{Q_1}{Q_0} \leq B
\]

i.e.,

\[
N(t) \leq \frac{\log \left( \frac{\beta}{1 - \alpha} \right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)}
\]

(4.1)

Decide the system to be unreliable and reject if

\[
\frac{Q_1}{Q_0} \geq A
\]

i.e.,

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

(4.2)

Continue the test procedure as long as

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

(4.3)

Substituting the appropriate expressions of the respective mean value function \( m(t) \) of Inflection S-shaped model we get the respective decision rules and are given in followings lines

Acceptance region:
Assuming the value of \( \delta \) is known, we can derive the decision rules for the SPRT methodology. The decision regions are given by Equations (4.4), (4.5), and (4.6).

### Rejection region:

\[
N(t) \geq \frac{\log \left( \frac{1-\beta}{\alpha} + \frac{a(1+c)(e^{-b \delta} - e^{-b})}{(1+ce^{-b \delta})(1+ce^{-b})} \right)}{\log \left( \frac{1-e^{-b \delta}}{(1+ce^{-b \delta})(1-e^{-b})} \right)}
\]

### Continuation region:

\[
\frac{\log \left( \frac{\beta}{1-\alpha} + \frac{a(1+c)(e^{-b \delta} - e^{-b})}{(1+ce^{-b \delta})(1+ce^{-b})} \right)}{\log \left( \frac{1-e^{-b \delta}}{(1+ce^{-b \delta})(1-e^{-b})} \right)} < N(t) < \frac{\log \left( \frac{1-\beta}{\alpha} + \frac{a(1+c)(e^{-b \delta} - e^{-b})}{(1+ce^{-b \delta})(1+ce^{-b})} \right)}{\log \left( \frac{1-e^{-b \delta}}{(1+ce^{-b \delta})(1-e^{-b})} \right)}
\]

It may be noted that in the above 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 (1997). In that sense equations (4.1), (4.2), (4.3) 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.

### 5. 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 Five different data sets of the above form, borrowed from Pham (2006) and Lyu. 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 \( \delta = 0.0025 \) and \( c = 0.05 \) the choices are given in the following table.

#### Table 5.1: Estimates of \( a, b \) & Specifications of \( b_0, b_1 \) for Time domain

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Estimate of ‘a’</th>
<th>Estimate of ‘b’</th>
<th>( b_0 )</th>
<th>( b_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>33.239615</td>
<td>0.003212</td>
<td>0.000712</td>
<td>0.005712</td>
</tr>
<tr>
<td>DS2</td>
<td>33.498685</td>
<td>0.006139</td>
<td>0.003639</td>
<td>0.008639</td>
</tr>
<tr>
<td>DS3</td>
<td>23.298053</td>
<td>0.003595</td>
<td>0.001095</td>
<td>0.006095</td>
</tr>
<tr>
<td>DS4</td>
<td>25.031345</td>
<td>0.003057</td>
<td>0.000557</td>
<td>0.005557</td>
</tr>
<tr>
<td>DS5</td>
<td>31.404848</td>
<td>0.020966</td>
<td>0.018466</td>
<td>0.023466</td>
</tr>
</tbody>
</table>

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 5.3.4 and 5.3.5, 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. The following consolidated table reveals the iterations required to come to a decision about the software of each Data Set.
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.
6. CONCLUSION.

The above consolidated table shows that Inflection S-shaped model as exemplified for 5 Data Sets indicate that the model is performing well in arriving at a decision. The model has given a decision of acceptance for 2 Data Sets i.e DS1 & DS2, a decision of rejection for 2 Data Sets i.e DS3 & DS4 and Continue for 1 Data set i.e DS5. Therefore, we may conclude that, applying SPRT on data sets we can come to an early conclusion of reliability / unreliability of software.

REFERENCES


Authors:

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 a first rank 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 published 50 research papers in National & International Journals.

Mr. K. Prasad Rao, Working as a Professor and Director, Dept. of M.C.A, CMR Institute of Technology. He is having 22 years of experience as a Head & Lecturer in Computer Science field. He published papers in 3 National and 1 International journals. He is pursuing Ph.D at Acharya Nagarjuna University. His research interests lies in Software Engineering.

Mr. G. Krishna Mohan, 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, M.Tech from JNTU, Kakinada, M.Phil from Madurai Kamaraj University and pursuing Ph.D from Acharya Nagarjuna University. He qualified, AP State Level Eligibility Test. His research interests lies in Data Mining and Software Engineering. He published 14 research papers in various National and International journals.