Detection of Burr Type XII Reliable Software using Sequential Process Ratio Test

K. V. Murali Mohan¹, R. Satya Prasad² and G. Sridevi³*

¹ECE, H. M. Institute of Technology and Science, Hyderabad - 501301, Andhra Pradesh, India; kvmuralimohan22@gmail.com
²CSE, Acharya Nagarjuna University, Guntur - 522 510, Andhra Pradesh, India; profrsp@gmail.com
³CSE, KL University, Guntur – 522 502, Andhra Pradesh, India; sridevi.gutta2012@gmail.com

Keywords: Burr Type XII Model, ML Estimation, NHPP, Sequential Probability Ratio Test, Time Domain Data

1. Introduction

Wald's procedure is particularly relevant if the data is collected sequentially. Sequential Analysis is different from Classical Hypothesis. With classical hypothesis, first the entire data need to be collected and then the analysis is done to attain the conclusions. But where as in Sequential Analysis, each and every test case is analyzed soon after the data has been collected, and also the results are compared with some threshold value incorporating the new information obtained with the current test case. This permits one to come up with the conclusions during the data collection itself, so that the final decision may be made at much earlier stage. Wald's procedure is well suited if the data is collected sequentially. The main advantage of Sequential Analysis is that the decisions can be taken at an earlier time saving the human time and also in terms of money.

Sequential Probability Ratio Test (SPRT) is usually applied at the circumstances where we need to take decision between two simple hypotheses or a single decision point. The SPRT procedure can be used to distinguish the software under test into one of the two categories like reliable/unreliable, pass/fail and certified/uncertified⁴. The software failure data analysis can be done either by considering the time between failures or failure count in a specific time interval. Also it is assumed that the average number of recorded failures in a given time interval is directly proportional to the length of the interval. The random number of failure occurrences in a given interval can be explained by the Poisson process.

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

(1.1)

As per the observations made by the⁵, the reliability predictions are misleading by applying software reliability growth models in classical hypotheses testing. According to the observations made by him, the statistical methods can be successfully applied to the failure
data. His observations are demonstrated by applying the well-known Sequential Probability Ratio Test (SPRT) of \(^3\) for a software failure data to detect unreliable software components and compare the reliability of different software versions. In this paper, we consider one of the popular software reliability growth model Burr Type XII and adopt the principle of \(^3\) in detecting whether the software is reliable or unreliable in order to accept or reject the developed software. The theory proposed by \(^3\) is described in section 2 Implementation of SPRT for the proposed Burr type XII Software Reliability Growth Model is illustrated in section 3. Maximum Likelihood estimation method is used to estimate the parameters is presented in Section 4. Application of the decision rule to detect the unreliable software with reference to the Software Reliability Growth Model Burr Type XII is depicted in section 5.

2. Wald’s Sequential Test for a Poisson Process

The Sequential Probability Ratio Test (SPRT) was developed by Abraham Wald at Columbia University in 1943\(^4\). The SPRT procedure is used for quality control studies during the manufacturing of software products. The tests can be performed with fewer observations as compared to fixed sample size sets. Testing is performed on large volumes of data which consumes large amount of time in classical hypothesis and it can be reduced to a large extent by implementing Sequential Probability Ratio Tests. The SPRT methodology for Homogeneous Poisson Process given by Stieber is described below\(^5\).

Let \(\{N(t), t \geq 0\}\) be a homogeneous Poisson process with rate \(\lambda\). In this case, \(N(t) = \) number of failures up to time \(t\) and \(\lambda\) is the failure rate (failures per unit time). If the system is put on test and that if we want to estimate its failure rate \(\lambda\). We cannot expect to estimate \(\lambda\) precisely. But we want to reject the system with a high probability if the data suggest that the failure rate is larger than \(\lambda\), and accept it with a high probability, if it is smaller than. Here 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 \(\lambda \leq \lambda_1\). This is the “producer’s” risk, \(\beta\) is the probability of falsely accepting the system. That is accepting the system even if \(\lambda \leq \lambda_1\). 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 as \(t\) > 0 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

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

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

The ratio \(\frac{P_1}{P_0}\) at any time \(t\) is considered as a measure of deciding the truth towards \(\lambda_0\) or \(\lambda_1\), given a sequence of time instants say \(t_1 < t_2 < \ldots < t_k\) and the corresponding realizations \(N(t_1), N(t_2) \ldots N(t_k)\) or \(N(t)\). Simplification of \(\frac{P_1}{P_0}\) gives

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

The decision rule of SPRT is to decide in favour of \(\lambda_1\) or \(\lambda_0\) 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\(^5\) the test process with one more observation in failure data, according to

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

\[
\frac{P_1}{P_0} \leq B \quad (2.4)
\]

\[
B < \frac{P_1}{P_0} < A \quad (2.5)
\]

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

\[
A \equiv 1 - \frac{\beta}{\alpha}, B \equiv \frac{\beta}{1 - \alpha}
\]

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.
\[ N_i(t) = at + b_2 \]  

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

\[ N_i(t) = at - b_1 \]  

To continue the test with one more observation on \([t, N(t)]\) as the random graph \([t, N(t)]\) of is between the two linear boundaries given by equations (2.6) and (2.7) where

\[ a = \frac{\lambda_1 - \lambda_0}{\log \left[ \frac{\lambda_1}{\lambda_0} \right]} \]  

\[ b_1 = \frac{\log \left[ \frac{1-a}{\beta} \right]}{\log \left[ \frac{\lambda_1}{\lambda_0} \right]} \]  

\[ b_2 = \frac{\log \left[ \frac{1-\beta}{a} \right]}{\log \left[ \frac{\lambda_1}{\lambda_0} \right]} \]  

The parameters \( a, \beta, \lambda_0, \) and \( \lambda_1 \) can be chosen in several ways. One way suggested by \(^5\) is

\[ \lambda_0 = \frac{\lambda \log (q)}{q - 1} \]

\[ \lambda_1 = \frac{q \lambda \log q}{q - 1} \]  

Where \( q = \frac{\lambda_1}{\lambda_0} \)

If \( \lambda_0 \) and \( \lambda_1 \) are chosen in this way, the slope of \( N_i(t) \) and \( N(t) \) 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 or components.

### 3. Sequential Probability Ratio Test for Burr Type XII SRGM

In Section 2, for the Poisson process we know that the expected value \( N(t) = \lambda(t) \) called the average number of failures experienced in time 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) as its mean value function the probability equation of such a process is

\[ \Pr[N(t) = Y] = \begin{pmatrix} m(t) \\ y! \end{pmatrix} e^{-m(t)}, y = 0, 1, 2\ldots \]

Depending on the forms of \( m(t) \) we get various Poisson processes called NHPP, for our Burr type XII model. The mean value function is given as

\[ m(t) = a \left[ 1 - (1 + t^b)^{-c} \right], \quad t \geq 0 \]

It can also be written as

\[ P_0 = \frac{e^{-m(t)} \left[ m_1(t) \right]^{N(t)}}{N(t)!} \]

\[ P_1 = \frac{e^{-m(t)} \left[ m_0(t) \right]^{N(t)}}{N(t)!} \]

Where \( m_1(t), m_0(t) \) represents the mean value function at stated parameters indicating reliable software and unreliable software respectively. The mean value function \( m(t) \) comprises the parameters ‘a’, ‘b’ and ‘c’. The two specifications of NHPP for \( b \) are considered as \( b_1, b_2 \) where \( b_1 < b_2 \) and two specifications of \( c \) say \( c_1, c_2 \) where \( c_1 < c_2 \). For our proposed model, \( m(t) \) at \( b_1 \) is said to be greater than \( b_0 \) and \( m(t) \) at \( c_1 \) is said to be greater than \( c_0 \). The same can be denoted symbolically as \( m_1(t) < m_0(t) \). The implementation of SPRT procedure is illustrated below.

System is said to be reliable and can be accepted if

\[ \frac{P_1}{P_0} \leq B \]

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

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

\[ \text{i.e.,} \quad N(t) \leq \frac{\log m_1(t) - \log m_0(t)}{\log m_1(t) - \log m_0(t)} \]

System is said to be unreliable and rejected if

\[ \frac{P_1}{P_0} \geq A \]

\[ \text{i.e.,} \quad \frac{e^{-m(t)} \left[ m_1(t) \right]^{N(t)}}{e^{-m(t)} \left[ m_0(t) \right]^{N(t)}} \geq A \]

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

\[ \text{i.e.,} \quad N(t) \geq \frac{\log m_1(t) - \log m_0(t)}{\log m_1(t) - \log m_0(t)} \]
Detection of Burr Type XII Reliable Software using Sequential Process Ratio Test

Continue the test procedure as long as
\[
\log \left( \frac{\beta}{1 - a} \right) + m_1(t) - m_0(t) \frac{1 - \beta}{a} + m_1(t) - m_0(t) \frac{m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} < N(t)
\]

Substituting the appropriate expressions of the respective mean value function, we get the respective decision rules and are given in followings lines.

Acceptance Region:
\[
N(t) \leq \log \left( \frac{\beta}{1 - a} \right) + a \left( \frac{1 + t^a}{1 + t^c} \right)^{-b} - \left( \frac{1 + t^a}{1 + t^c} \right)^{-b} \]

(3.4)

Rejection Region:
\[
N(t) \geq \log \left( \frac{1 - \beta}{a} \right) + a \left( \frac{1 + t^a}{1 + t^c} \right)^{-b} - \left( \frac{1 + t^a}{1 + t^c} \right)^{-b} \]

(3.5)

Continuation Region:
\[
\log \left( \frac{\beta}{1 - a} \right) + a \left( \frac{1 + t^a}{1 + t^c} \right)^{-b} - \left( \frac{1 + t^a}{1 + t^c} \right)^{-b} \frac{1}{\log m_1(t) - \log m_0(t)} < N(t)
\]

(3.3)

For the specified model, it may be observed that the decision rules are exclusively based on the strength of the sequential procedure \((a, \beta)\) and the value of the mean value functions namely \(m_0(t)\) and \(m_1(t)\). As described by\(^3\), these decision rules become decision rules if the mean value function is linear in passing through origin, that is \(m(t) = \lambda t\). The equations (3.1), (3.2) and (3.3) are considered as generalizations for the decision procedure of\(^3\). SPRT procedure is applied on live software failure data sets and the results that were analyzed are illustrated in Section 5.

4. Parameter Estimation

In this section we develop expressions to estimate the parameters of the Burr type XII model based on time domain data. Parameter estimation is very significant in software reliability prediction. Once the analytical solution form is known for a given model, parameter estimation is achieved by applying a well-known estimation, Maximum Likelihood Estimation (MLE)\(^2\).

The main idea behind Maximum Likelihood parameter assessment is to decide the parameters that maximize the probability (likelihood) of the specimen data. In the other words, MLE methods are versatile and applicable to most models and for different types of data. In this paper parameters are estimated from the time domain data.

The mean value function of Burr type XII model is given by
\[
m(t) = a \left( 1 + t^c \right)^{-b}, \quad t \geq 0
\]

(4.1)

The parameters \(a, b, c\) are estimated with Maximum Likelihood (ML) estimation.

The likelihood function for time domain data is given by
\[
L = e^{-m(t)} \prod_{i=1}^{n} m'(t_i)
\]

(4.2)

Substituting Equation (4.1) in equation (4.2) we get,
\[
L = e^{-a(1 + t^c)^{-b}} \prod_{i=1}^{n} \frac{abct_i^{-1}}{(1 + t^c)^{b+1}}
\]

\[
\log L = -a + a(1 + t^c)^{-b} + \sum_{i=1}^{n} \left[ \log a + \log b + \log c + (c - 1) \log t_i + (b + 1) \log(1 + t_i^c) \right]
\]

(4.3)

Taking the Partial derivative with respect to ‘a’ and equating to ‘0’.

\[
\frac{\partial \log L}{\partial a} = 0
\]

\[
\therefore a = \frac{n(1 + t^c)^b}{(1 + t^c)^b - 1}
\]

(4.4)
The parameter ‘b’ is estimated by iterative Newton Raphson Method using \( b_{n+1} = b_n - \frac{g(b)}{g'(b)} \). Where \( g(b) \) and \( g'(b) \) are expressed as follows.

\[
g(b) = \frac{\partial \log L}{\partial b} = 0
\]

\[
\frac{\partial \log L}{\partial b} = g(b) = \frac{n \log \left( \frac{1}{t+1} \right)}{(t+1)^b - 1} - \frac{n}{b} \sum_{i=1}^{n} \log(t_i + 1) \quad (4.5)
\]

\[
g'(b) = \frac{\partial^2 \log L}{\partial b^2} = 0
\]

\[
\frac{\partial^2 \log L}{\partial b^2} = g'(b) = -n \left[ \log \left( \frac{1}{t+1} \right) \left( \frac{(t+1)^b \log(t+1)}{(t+1)^b - 1} \right)^2 \right] + \frac{1}{b^2} \quad (4.6)
\]

The parameter ‘c’ is estimated by iterative Newton Raphson Method using

\[
c_{n+1} = c_n - \frac{g(c)}{g'(c)}
\]

Where \( g(c) \) and \( g'(c) \) are expressed as follows.

\[
g(c) = \frac{\partial \log L}{\partial c} = 0
\]

\[
\frac{\partial \log L}{\partial c} = g(c) = -n \log(t) + \sum_{i=1}^{n} \frac{t_i}{(1+t_i)} + \sum_{i=1}^{n} \log(t_i)
\]

\[
\frac{\partial^2 \log L}{\partial c^2} = g'(c) = 0
\]

\[
\frac{\partial^2 \log L}{\partial c^2} = g'(c) = n \left( \frac{\log(t)}{(1+t)^2} \right) - \sum_{i=1}^{n} \log(t_i)
\]

### Table 1. Estimates of \( a, b, c \) & specifications of \( b_0, b_1, c_0, c_1 \)

<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>
<th>Estimate of ‘c’</th>
<th>( c_0 )</th>
<th>( c_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xie</td>
<td>30.040800</td>
<td>0.999825</td>
<td>0.499825</td>
<td>1.499825</td>
<td>0.999619</td>
<td>0.499619</td>
<td>1.499619</td>
</tr>
<tr>
<td>AT &amp; T</td>
<td>22.032465</td>
<td>0.999859</td>
<td>0.499859</td>
<td>1.499859</td>
<td>0.999619</td>
<td>0.499619</td>
<td>1.499619</td>
</tr>
<tr>
<td>IBM</td>
<td>15.051045</td>
<td>0.999530</td>
<td>0.499530</td>
<td>1.499530</td>
<td>0.999196</td>
<td>0.499196</td>
<td>1.499196</td>
</tr>
<tr>
<td>NTDS</td>
<td>26.105273</td>
<td>0.998899</td>
<td>0.498899</td>
<td>1.498899</td>
<td>0.998903</td>
<td>0.498903</td>
<td>1.498903</td>
</tr>
<tr>
<td>SONATA</td>
<td>30.016391</td>
<td>0.99958</td>
<td>0.49958</td>
<td>1.49958</td>
<td>0.99920</td>
<td>0.49920</td>
<td>1.49920</td>
</tr>
</tbody>
</table>

### Table 2. SPRT Analysis for 5 data sets

<table>
<thead>
<tr>
<th>Data Set</th>
<th>( T )</th>
<th>( N(t) )</th>
<th>R.H.S. of equation (3.4) Acceptance region ( \leq )</th>
<th>R.H.S. of equation (3.5) Rejection region ( \geq )</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xie</td>
<td>30.02</td>
<td>1</td>
<td>0.003606</td>
<td>0.005130</td>
<td>Rejection</td>
</tr>
<tr>
<td>AT &amp; T</td>
<td>5.50</td>
<td>1</td>
<td>0.115010</td>
<td>0.164486</td>
<td>Rejection</td>
</tr>
<tr>
<td>IBM</td>
<td>10</td>
<td>1</td>
<td>0.023281</td>
<td>0.040821</td>
<td>Rejection</td>
</tr>
<tr>
<td>NTDS</td>
<td>9</td>
<td>1</td>
<td>0.047184</td>
<td>0.065189</td>
<td>Rejection</td>
</tr>
<tr>
<td>SONATA</td>
<td>52.50</td>
<td>1</td>
<td>0.0010008</td>
<td>0.001499</td>
<td>Rejection</td>
</tr>
</tbody>
</table>
6. Conclusion

The SPRT methodology for the proposed software reliability growth model Burr type XII is applied for the software failure data sets. Hence, it is observed that we are able to come to a conclusion in less time regarding the reliability or unreliability of a software product. The results exemplifies that the model has given a decision of rejection for all the data sets at various time instant of the data. Therefore, we may conclude that, applying SPRT on data sets we can come to an early conclusion of reliable/unreliable of software.

7. References