International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 2, Issue 3, March 2015, PP 30-32
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
www.arcjournals.org

SPC to Burr Type III Software Reliability

Dr. R.Satya Prasad

Associate Professor, Dept. of Computer Science & Engineering
Acharya Nagarjuna University, Nagarjuna Nagar
Guntur, Andhra Pradesh, INDIA

Ch.Smitha Chowdary

Research Scholar, Dept. of Computer Science
Krishna University, Machilipatnam
Andhra Pradesh INDIA

K.Sobhana

Research Scholar, Dept. of Computer Science
Krishna University, Machilipatnam
Andhra Pradesh INDIA

Abstract: *Software reliability is assessed quantitatively by using Software Reliability Growth Model (SRGM) for tracking and measuring the growth of reliability. Reliability of software is the probability of failure-free operation during specified period in specified environment. To improve reliability and quality the execution of software process must be controlled and the accepted choice for monitoring software process is Statistical Process Control (SPC). This helps the professionals to identify anomalies while monitoring the process and take the necessary action. In this paper we present Non Homogeneous Poisson Process (NHPP) based Burr type III software reliability growth model with time domain data. The Maximum Likelihood (ML) estimation method is used for finding unknown parameters in the model on ungrouped data. Applicability of SPC for monitoring software reliability process is given.*

Keywords: *Burr type III, NHPP, ML estimation, Software Reliability, SPC, Time domain data.*

1. INTRODUCTION

Software reliability is the probability that the software system will operate without failure for a specified time under specified operating conditions [1,2]. Over the past few decades, statistical models of different types have been discussed for assessment of the software reliability. Software reliability is assessed quantitatively by using Software Reliability Growth Model (SRGM) for tracking and measuring the growth of reliability and is used to compute the reliability growth of products during software development phase. These models can be of two types i.e. static and dynamic. The static model uses software metrics in order to estimate the number of defects in the software and the dynamic model uses the past failure discovery rate to estimate the number of failures during software execution over time.

In software engineering one always wants to produce high quality software at low cost. As no one is perfect there is a possibility of errors in the software developed by humans. To improve software reliability these errors need to be identified while the software process is in development and a widely accepted choice for this is the Statistical Process Control.

In this paper we probe applicability of SPC to Burr type III software reliability growth model to analyze the reliability of a software system using Time domain data. The layout of the paper is as follows: Section 2 describes the formulation and interpretation of the Burr type III model for the underlying NHPP. Section 3 discusses Maximum Likelihood (ML) estimation of Burr type III model based on time domain data. Section 4 describes SPC and its applicability. Section 5 presents the conclusion.

2. BACKGROUND FORMULATION OF NHPP

Here we present the theory which underlies NHPP models, the SRGMs under consideration and maximum likelihood estimation for ungrouped data. Let 't' be a continuous random variable in pdf: $f(t;\theta_1, \theta_2,\ldots. \theta_k)$ where, $\theta_1, \theta_2,\ldots. \theta_k$ are k unknown constant parameters which need to be estimated, and cdf: $F(t)$ where the mathematical relationship between the pdf and cdf is given by: $f(t) = F'(t)$. If 'a' denotes the expected number of faults that would be detected given infinite testing time then, the mean value function of the NHPP models can be written as: $m(t) = aF(t)$, where F(t) is a cumulative distribution function then the failure intensity function $\lambda(t)$ in case of NHPP models is given as: $\lambda(t) = aF'(t)$ [3].

2.1. NHPP Model

The Non-Homogenous Poisson Process (NHPP) based software reliability growth models (SRGMs) are proved to be quite successful in practical software reliability engineering [4]. The main issue in the NHPP model is to determine an appropriate mean

value function to denote the expected number of failures experienced up to a certain point in time. Model parameters can be estimated by using Maximum Likelihood Estimate (MLE). Various NHPP SRGMs have been built upon various assumptions. Many of the SRGMs assume that each time a failure occurs, the fault that caused it can be immediately removed and no new faults are introduced. Which is usually called perfect debugging.

2.2. Model Under Consideration: Burr Type III

Burr [5] introduced twelve different forms of cumulative distribution functions for modeling data. The probability density function of a three-parameter Burr type III distribution has the form: f (t,b,c) $= \frac{\text{bct}^{\text{bc-1}}}{[1+\text{t}^{\text{c}}]^{\text{b+1}}}$ where b,c are shape parameters. The corresponding cumulative distribution function is: F (t) = $[1+t^{-c}]^{-b}$. The mean value function $m(t) = a$ $[1+t^{-c}]^{-b}$. The failure intensity function is given as: λ $(t) = \left[\frac{abc}{t^{c+1}(1+t^{-c})^{b+1}}\right]$

3. MAXIMUM LIKELIHOOD ESTIMATION

The preferred method of obtaining parameter estimates is to use the maximum likelihood equations. These equations are derived from the model equations and the assumptions which underlay them. The parameters are taken to be those values which maximize these likelihood functions. These values are found by taking the partial derivate of the likelihood function with respect to the model parameters, the maximum likelihood equations, and setting them to zero. Iterative routines are then used to solve these equations.

Log Likelihood function for ungrouped data [3] is given as,

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

The maximum likelihood estimators of θ_1, θ_2,….. θ_n obtained by maximizing L or $\wedge$, where $\wedge$ is in L . By maximizing $\wedge$, which is much easier to work with than L, the maximum likelihood estimators (MLE) of θ_1, θ_2,….. θ_n are the simultaneous solutions of n equations such

as: $\frac{\partial(\wedge)}{\partial\theta} = 0$, i=1,2,…,n.

3.1. Illustration of Parameter Estimation

Cumulative time between failures data for software reliability monitoring is used. Using the estimators of ‘a’, ‘b’ and ‘c’ we computed m (*t*) [6] .

The Log Likelihood function is given as:

$$LogL = \sum_{i=1}^{n} \log\left[\frac{abc}{t_i^{c+1}(1+t_i^{-c})^{b+1}}\right] - \frac{a}{\left[1+t_n^{-c}\right]^b} \qquad (2)$$

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

$$a = n\left(1+t_n^{-c}\right)^b \qquad (3)$$

Taking the Partial derivative of log L with respect to ‘b’ and equating to‘0’.

$$b = \frac{n}{\sum_{i=1}^{n} \log\left(1+t_i^{-c}\right) - n\log\left(1+t_n^{-c}\right)} \qquad (4)$$

The parameter ‘c’ is estimated by iterative Newton-Raphson Method using $c_{i+1} = c_i - \frac{g(c_i)}{g'(c_i)}$ where g(c) and g’(c) is expressed as follows.

$$g(c) = \frac{-n\log(t_n)}{1+t_n^c} + \frac{n}{c} + \sum_{i=1}^{n} \log t_i \left[-1 + \frac{2}{1+t_i^c}\right] = 0 \qquad (5)$$

$$g'(c) = \frac{n(\log t_n)^2 t_n^c}{(t_n^c+1)^2} - \frac{n}{c} - \sum_{i=1}^{n} \frac{2t_i^c(\log t_i)^2}{(t_i^c+1)^2} \qquad (6)$$

4. APPLICABILITY OF SPC

Software reliability growth models (SRGM’s) are useful to assess the reliability for quality management and testing progress control of software development. To improve reliability and quality the execution of software process must be controlled and the choice for monitoring software process is Statistical Process Control.

The parameters estimated can be used to monitor the process through SPC concepts and methods over time, in order to verify that the process remains in the state of statistical control. SPC may help in finding assignable causes, long term improvements in the software process. Software quality and reliability can be achieved by eliminating the causes or improving the software process or its operating procedures [7].

Control Charts

The most popular technique of SPC for maintaining process control is control charting. The control chart is one of the seven tools for quality control. SPC is used to secure, that the quality of the final product will conform to predefined standards. In any process, regardless of how carefully it is maintained, a certain

amount of natural variability will always exist. A process is said to be statistically "in-control" when it operates with only chance causes of variation. On the other hand, when assignable causes are present, then we say that the process is statistically "out-of-control". Control charts are capable to create an alarm when a shift in the level of one or more parameters of a distribution occurs. Normally, such a situation will be reflected in the control chart by points plotted outside the control limits or by the presence of specific patterns. The most common non-random patterns are cycles, trends, mixtures and stratification [8]. For a process to be in control the control chart should not have any trend or nonrandom pattern. The selection of proper SPC charts is essential to effective statistical process control implementation and use. The SPC chart selection is based on data, situation and need [9]. Chan et al.,[10] proposed a procedure based on the monitoring of cumulative quantity. This approach has been shown to have a number of advantages: it does not involve the choice of a sample size; it raises fewer false alarms; it can be used in any environment; and it can detect further process improvement. Xie et al.,[11] proposed t-chart for reliability monitoring where the control limits are defined in such a manner that the process is considered to be out of control when one failure is less than LCL or greater than UCL. The traditional false alarm probability is to set to be 0.27% although any other false alarm probability can be used. The actual acceptable false alarm probability should in fact depend on the actual product or process [12].

$$T_u = a(1+t^{-c})^{-b} = 0.99865$$

$$T_c = a(1+t^{-c})^{-b} = 0.05$$

$$T_l = a(1+t^{-c})^{-b} = 0.00135$$

These limits when converted to $m(t_U)$, $m(t_C)$ and $m(t_L)$ form will be used to find whether the software process is in control or not by placing the points in Mean value chart. A point below the control limit $m(t_L)$ indicates an alarming signal. A point above the control limit $m(t_U)$ indicates better quality. If the points are falling within the control limits, it indicates the software process is in stable condition [13].

5. CONCLUSION

In this paper we presented Burr type III software reliability growth model which is primarily useful in estimating and monitoring software reliability, viewed as a measure of software quality. To improve quality of a process the execution of software process must be monitored and controlled. SPC is one such process for monitoring. Control charts are a tool of SPC that help in monitoring through which quality can be improved. The early detection of software failure will improve the software reliability. When the control signals are below the control limit, it is likely that there are assignable causes leading to significant process deterioration and it should be investigated. Hence, we conclude that our control mechanism will give a positive recommendation for its use to estimate whether the process is in control or out of control.

REFERENCES

[1] *J. D. MUSA. Software Reliability Engineering.Wiley.1998.*

[2] *J. D. MUSA, A. IANNINO, AND K. OKUMOTO. Software Reliability Measurement Prediction Application. McGraw-Hill, 1987. ISBN 0-07-044093-X.*

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

[4] *Musa, J. D.; Iannino, A.; Okumoto, K. (1987). "Software Reliability - Measurement, Prediction,Application", New York.*

[5] *Burr (1942), "Cumulative Frequency Functions", Annals of Mathematical Statistics, 13, pp. 215-232.*

[6] *Ch.Smitha Chowdary, Dr.R.Satya Prasad, K.Sobhana (2015)," Burr Type III Software Reliability Growth Model", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 1, Ver. I (Jan – Feb. 2015),PP 49-54.*

[7] *Kimura, M., Yamada, S., Osaki, S., (1995). "Statistical Software reliability prediction and itsapplicability based on mean time betweenfailures". Mathematical and Computer Modelling Volume 22, Issues 10-12, Pages 149-155.*

[8] *Koutras, M.V., Bersimis, S., Maravelakis,P.E., 2007. "Statistical process control using shewart control charts with supplementary Runs rules" Springer Science + Business media 9:207-224.*

[9] *MacGregor, J.F., Kourti, T., 1995. "Statistical process control of multivariate processes". Control Engineering Practice Volume 3, Issue 3, March 1995, Pages 403-414.*

[10] *Chan, L.Y, Xie, M., and Goh. T.N., (2000), "Cumulative quality control charts for monitoring production processes. Int J Prod Res; 38(2):397-408.*

[11] *Xie. M, T.N Goh and P.Ranjan. (2002). "Some effective control chart procedures for reliability monitoring", Reliability Engineering and System Safety. 77, 143-150.*

[12] *Swapna S. Gokhale and Kishore S.Trivedi, 1998. "Log-Logistic Software Reliability Growth Model". The 3rd IEEE International Symposium on High-Assurance Systems.*

[13] *MacGregor, J.F., Kourti, T., 1995. "Statistical process control of multivariate processes". Control Engineering Practice Volume 3, Issue 3, March 1995, Pages 403-414.*