VOL. 11, NO. 1, JANUARY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

SOFTWARE RELIABILITY ANALYSIS USING LIFETIME DISTRIBUTIONS

V. Vallinayagam[1], S. Parthasarathy[2] and P. Venkatesan[3]
[1]Department of Mathematics, St.Josephes Engineering College, Chennai, India
[2]Department of Mathematics, SRM University-Ramapuram Campus, Ramapuram, Chennai, India
[3]Department of Statistics, National Institute for Research in Tuberculosis, Chennai, India
E-Mail: vngam05@yahoo.co.in

ABSTRACT

In this paper, an empirical comparison made between three parametric models namely Exponential, Log-logistic and Gompertz distributions in the analysis of software reliability data. Processor failure data was used to compare the models in terms of deviance. Gompertz distribution gave the better fit than the other models in terms of deviance.

Keyword: software reliability, deviance, MLE, exponential, log-logistic, gompertz.

1. INTRODUCTION

Software reliability is a key part in software quality. The nature and complexity of software have changed lot in the past few decades. In the recent years, it is very necessary to produce good quality of software with high precession of reliability. In the olden days software errors and bugs were fixed at a later stage in the software development. Today to produce high quality reliable software is a big challenge because nowadays most standard components and better process are introduced every day in this software engineering field. If not considered carefully, software reliability can be the reliability bottleneck of the whole system. Ensuring software reliability is no easy task. As hard as the problem is, promising progresses are still being made toward more reliable software.

Software Reliability measurement is a set of mathematical procedures that can be used to estimate and predict the reliability behavior of software during its development and operation. The primary goal of software reliability is to answer the following question, that is given a system, what is the probability that it will fail in a given time interval, or, what is the expected duration between successive failures? Also it is an important factor affecting system reliability see e.g., Musa [5, 6], Lyu [4] and Pham H [7, 8]. It differs from hardware reliability in that it reflects the design perfection, rather than manufacturing perfection. The high complexity of software is the major contributing factor of Software Reliability problems. Software Reliability is not a function of time - although researchers have come up with models relating the two.

The modeling technique for Software Reliability is reaching its prosperity, but before using the technique, we must carefully select the appropriate model that can best suit our case. Measurement in software is still in its infancy. No good quantitative methods have been developed to represent Software Reliability without excessive limitations. Various approaches can be used to improve the reliability of software, however, it is hard to balance development time and budget with software reliability. Software failures may be due to errors, ambiguities, oversights or misinterpretation of the specification that the software is supposed to satisfy, carelessness or incompetence in writing code, inadequate testing, incorrect or unexpected usage of the software or other unforeseen problems.

In order to estimate software reliability data we have to use some probability models. Various NHPP (Non-homogeneous Poisson Process) software reliability models are available to estimate the software reliability. Therefore, in this paper, in the preliminary section, we have discussed three statistical distribution models namely Exponential, Log logistic and Gompertz to compare software reliability data with the help of deviance.

Many authors are analyzing software reliability data using some fixed model namely exponential power model etc. (see [12, 13]). But in this paper we have not fixed any mathematical model for this analysis whereas if we use deviance we can compare which mathematical model is fit for any software reliability data. Of course one parameter distribution like exponential is not sufficient to analyze the data. The other multi parameter distributions like gamma, Weibull, Log-normal, etc. are commonly used. In this paper we have taken only three distributions Exponential, Log-Logistic and Gompertz. We can use any mathematical model for this analysis but before we have to check whether the model is fit for the data are not using deviance.

Finally in the conclusion section, we have come up with the conclusion that which processor is more reliable for the given software reliability data with the help of mathematical distribution model and its deviance.

2. PRELIMINARIES

Software reliability $R(t)$ is the probability of failure free operation of a computer program for a specified time t under a specified environment. The term failure means departure of program operation from user requirements and a defect in a program that causes failure is called fault.

The expected number of failures in a given time interval is called failure intensity $F(t)$. Expected value of a failure given in an interval is called Mean-Time-To-Failure (MTTF). The number of failures expected in a time period t is denoted by $E(t)$. Let T be a random variable representing the failure time. The probability that the software will fail by time t is $F(t) = Pr[T \leq t] = \int_0^\infty f(x)dx$. The probability of the software executing until time t is $R(t) = Pr[T > t] = 1 - F(t) = \int_t^\infty f(x)dx$. Therefore MTTF or $E(T) = \int_0^\infty R(t)dt$. The Hazard rate is calculated by $\mu(t) = \frac{f(t)}{R(t)}$. The terms hazard rate and failure rate are often used interchangeably.

The simplest case when T is exponentially distributed random variable with hazard rate $\mu(t) = \lambda t$, then MTTF $= \frac{1}{\lambda}, R(t) = e^{-\mu(t)}$. Suppose after the first failure at some time T_1,the software becomes non-executable until it is repaired at some time $R_1 \geq T_1$, then it works until the occurrence of second failure at some time say $T_2 \geq R_1$etc. Then the mean time to failure MTTF $= E(T_1)$ and reliability time $R(t) = \Pr[T > t]$, mean time to repair MTTR $= E(R_1 - T_1)$ and mean time between failures is MTBF = MTTF +MTTR

2.1 Maximum Likelihood Estimation (MLE)

MLE can be used to estimate parameters for the suspended data along with complete failure data i.e., maximizes the probability of observing the data that we have. In Mathematical, if x is a continuous random variable with pdf $f(x, \theta_1, \theta_2, \ldots \theta_n)$ where $\theta_1, \theta_2, \ldots \theta_n$are n unknown parameters which need to be estimated, with k independent observations, $x_1, x_2, \ldots x_k$, which correspond in the case of life data analysis to failure times. The mathematical relationship between pdf and cdf is $f(x) = \frac{d(F(x))}{dx}$.The likelihood function L for the unknown parameters is given by $L(\theta_1, \theta_2, \ldots \theta_n | x_1, x_2, \ldots x_k) = \prod_{i=1}^k f(x_i, \theta_1, \theta_2, \ldots \theta_n)$. The logarithmic likelihood function is given by $L^* = In\ L = \sum_{i=1}^k f(x_i, \theta_1, \theta_2, \ldots \theta_n)$. The maximum likelihood estimators of $\theta_1, \theta_2, \ldots \theta_n$are obtained by maximizing the above equation. For getting the maximum value of each parameter, the partial derivatives with respect to each parameter have set to be zero i.e $\frac{\partial L^*}{\partial \theta_j} = 0, j = 1\ to\ n$. MLE estimation for the complete failure data with the failure times $t_1, t_2 \ldots t_n$ is $\frac{\partial L^*}{\partial \theta_j} = \frac{n}{\lambda} - \sum_{i=1}^n t_i = 0$.

2.2 Deviance

Let M_1be a generalized linear model. Let y be continuous random variable with pdf $f(y, \theta_1, \theta_2, \ldots \theta_n)$ where $\theta_1, \theta_2, \ldots \theta_n$are n unknown parameters which need to be estimated with n independent observations $y_1, y_2, \ldots y_n$. The loglikelihood function for this model is$L^*(M_1, y) = In\ L = \sum_{i=1}^n f(y_i, \theta)$.

Let $M_s$be a saturated model. Let y be continuous random variable with pdf $f(y, \theta_1, \theta_2, \ldots \theta_n)$where $\theta_1, \theta_2, \ldots \theta_n$are n unknown parameters which need to be estimated with p independent observations $y_1, y_2, \ldots y_p$. The loglikelihood function for this model is $L^*(M_s, y) = In\ L = \sum_{i=1}^p f(y_i, \theta)$.

Then the deviance of the model $M_1$is twice the difference between the loglikelihood of that model and the saturated modelM_s. That is $-2\big(L^*(M_1, y) - L^*(M_s, y)\big)$. This deviance has a chi-square distribution with degrees of freedom $n - p$. Where n is the number of observations in the model M_1 and p is the number of observations in the model M_s.

If M_1 and M_2 are two different generalized linear models. Then the fit of the model can be assessed by comparing the deviances $D_1$and D_2 of these models. The difference of the deviance is

$$D = D_2 - D_1$$
$$= -2\big(L^*(M_2, y) - L^*(M_s, y)\big) + 2\big(L^*(M_1, y) - L^*(M_s, y)\big)$$
$$= -2\big(L^*(M_2, y) - L^*(M_1, y)\big)$$

This deviance has a chi-square distribution with degrees of freedom v equal to the number parameters that are estimated in one model but fixed in the other. That is, it is equal to the difference in the number of parameters estimated in M_1 and M_2.

2.3 Distributions

2.3.1 Exponential

The single parameter exponential distribution is a very commonly used distribution in reliability engineering. It is used to describe units that have a constant failure rate. The Probability density function for this distribution is given by

$$f(t) = \frac{1}{ae^{\left(-\frac{1}{a}\right)t}}\ t \geq 0, a > 0,$$

where a is the scale parameter and t is the survival time. Note that, here rate parameter $\lambda = \frac{1}{a}$. Also the mean or MTTF is defined as follows $E(t) = \int_0^\infty t.f(t)dt = \int_0^\infty t.\lambda.e^{-\lambda t}\ dt = \frac{1}{\lambda}$. Failure rate function or hazard rate function can be defined as $\mu(t) = \lambda =$ constant.

2.3.2 Log-Logistic

In probability and statistics, the Log - Logistic distribution is a continuous probability distribution for a non-negative random variable. It is used in survival analysis as a parametric model for events whose rate increases initially and decreases later. The log logistic distribution can be used to model the lifetime of an object, the lifetime of organism, or a service time. The probability density function for this two parameter distribution is given by

$$f(x; a, b) = \frac{\left(\frac{b}{a}\right)\left(\frac{x}{a}\right)^{b-1}}{\left(1+\left(\frac{x}{a}\right)^{b}\right)^{2}}.$$

Where, a is scale parameter, b is the shape parameter and x is the random variable, $x \in (0, \infty)$.Hazard rate function for Log-Logistic distribution is $\mu(t) = \left(\frac{b}{a}\right)\left(\frac{t}{a}\right)^{b-1} /(1 + (t/a)^b)$ and Mean or MTTF is $E(t) = (a\pi/b)/\sin(\pi/b)$.

2.3.3 Gompertz

The probability density function of the Gompertz distribution with shape parameter a and rate parameter b is given by $f(x; a, b) = bae^{bx}\ e^{(-ae^{bx})}, a, b > 0, x \geq 0$. and hazard rate function is $\mu(t) = be^{(at)}$. The hazard rate is increasing if the shape parameter $a > 0$ and decreasing for $a < 0$. For $a = 0$ the Gompertz is equivalent to the exponential distribution with constant hazard and rate parameter b. Mean or MTTF for this distribution is $E(t) = \int_{-t}^{\infty} \frac{e^{-t}}{t} dt$.

3. ANALYSIS

In this article we have analyzed software reliability failure data which is available in STAT: Analysis of life time data JMP and text data files by Meeker 1987. The data which is in the form is given in Table-1.

Table-1. Data format description.

Variable	Description
Processor ID	There are five processor ID's connected by a local network
Failure time	In months
Recovery time	In months
Error type	Possible error types are CPU, I/O (network or disk problems), software, and unknown.

There are 183 processor failure data collected from distributed system which is connected by a local network with all the five processors. We have calculated failure count and mean time between failures for each processor.

Table-2. Comparison of deviance for life time distributions.

Parameters	Exponential			Log-logistic			Gompertz		
	Coeff	Std. Err	P>Z	Coeff	Std. Err	P>Z	Coeff	Std. Err	P>Z
ID 1	.01	.20	.95	-.06	.03	0.04	.36	.87	0.67
ID2	-.25	.27	.34	.16	.04	0.00	-1.30	1.42	0.36
ID3	-.14	.30	.62	.14	.06	0.02	-.05	1.43	0.96
ID 4	-.61	.35	.08	.20	.06	0.00	-1.94	1.93	0.31
Deviance	365.81			-48.89			6.22		

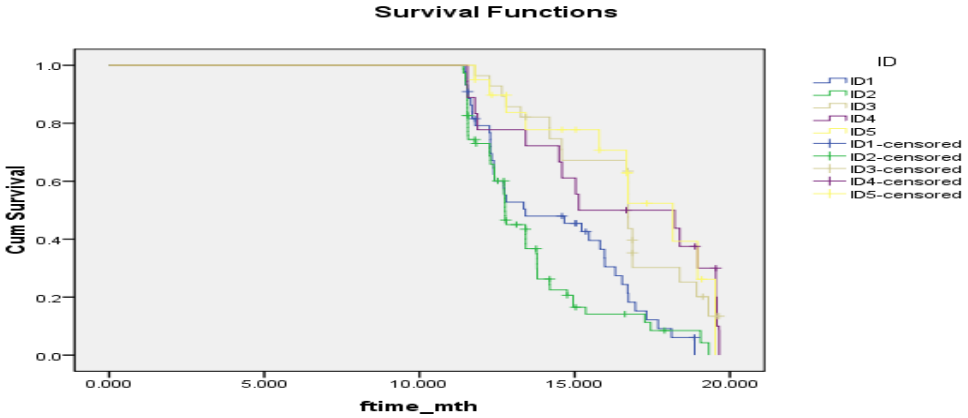
Table-2 shows that the deviance of each lifetime distributions. The deviance of Exponential and Gompertz has positive deviance whereas the deviance of Log-logistic is negative. Among the positive deviances, Exponential distribution has 365.81which is more deviated from the constant value and Gompertz has 6.22 which is less deviated from the constant value. Therefore Gompertz is better fit than Exponential. If the deviance value negative we can conclude that the distribution is not fit for that data.

Table-3. Mean and median of each processor (Failure time).

	Variable	ID1	ID2	ID3	ID4	ID5
Mean	Estimate	14.36	13.58	16.40	16.22	16.95
	St.d Error	.38	.29	.46	.77	.69
Median	Estimate	13.41	12.75	16.72	15.12	18.15
	St.d Error	1.52	.33	.004	3.19	.98

Table-3 shows that the mean and median of the each processors. Among the 5 processors, mean of 2nd processor is less than the other processors, that is, the number of failures in this processor is less than the other

processors. Similarly mean of 5th processor is greater than the rest of the processors that means the number of failures using processor 5 is higher than other processors.



4. CONCLUSIONS

Table-2 gives the deviance differences between the three distributions namely Exponential, Log-logistic and Gompertz. From Table-2 we have concluded that Gompertz is the best fit model for this software reliability data (low deviance). Also Log-Logistic is not suitable for this software reliability data (negative deviance). Without fixing any particular mathematical model, using this type of analysis we can conclude whether the model is fit or not for any software reliability data.

REFERENCES

[1] Florac W. A., Carleton A. D. 1999. Measuring the software process. Addison-Wesley.

[2] G.J. Knafl. 1992. Solving maximum likelihood equations for two-parameter software reliability models using grouped data. Proc. of the 3rd Int. Conf on Software Reliability Engineering, North Carolina, Research Triangle Park, IEEE Computer Press. pp. 205-213.

[3] G.J. Knafl and J. Morgan. 1996. Solving ML equations for 2-parameter Poisson-process models for ungrouped software-failure data. IEEE Transactions on Reliability. R-45(l): 42-53.

[4] M. Lyu. 1996. (Editor), Handbook of Software Reliability Engineering, McGraw-Hill, New York, USA.

[5] J.D. Musa, A. Iannino and K. Okumoto. 1987. Software Reliability Measurement Prediction Application, McGraw-Hill, New York.

[6] Musa J D. 2004. Software Reliability Engineering: More reliable software, faster and cheaper. Tata McGraw-Hill Education.

[7] Pham. H. 2006. System Software Reliability. Springer.

[8] Pham. H. 2003. Handbook of reliability Engineering. Springer.

[9] Read. 1983. Gombertz Distribution: Encyclopedia of Statistical Sciences. Wiley New York.

[10] V. Vallinayagam, S. Parthasarathy, P. Venkatesan. 2014. A Comparative Study of Life Time Models in the Analysis of Survival Data. IJAR. 4(1).

[11] V. Vallinayagam, S. Prathap, P. Venkatesan. 2014. Parametric Regression Models in the Analysis of Breast Cancer Survival Data. International Journal of Science and Technology. 3(3): 163-167.

[12] A K Srivastava and Vijaykumar. Analysis of Software reliability using Exponential power model. International Journal of Advanced Computer Science and Applications. 2(2): 38-45.

[13] Satya Prasad, B Sreenivasa Rao and R R L Kantham. Monitoring Software Reliability using Statistical process control: An MMLE Approach.