*International Journal of*
*Emerging Research in Management &Technology*
*ISSN: 2278-9359 (Volume-5, Issue-4)*

**Research Article** | **April 2016**

# Software Reliability Estimation: Gompertz

**Dr. R. Satya Prasad**
Associate Professor
Dept. of CS&E, Nagarjuna Nagar
ANU, Andhra Pradesh, India

**V. Surya Narayana**
Associate Professor
Dept. of CS&E. Ramachandra College of Engg.
Eluru, Andhra Pradesh, India

**Dr. G. Krishna Mohan**
Associate Professor
Dept. of CS&E, KL University,
Vaddeswaram, Andhra Pradesh, India

***Abstract—***

***Software Reliability Growth Model is a mathematical model of how the software reliability improves as faults are detected and repaired. The performance of SRGM is judged by its ability to fit the software failure data. How good does a mathematical model fit to the data and reliability of software is presented in the current paper. The model under consideration is the, Gompertz model. MLE method is used to estimate the model parameters. To assess the performance of the considered Software Reliability Growth Model, we have carried out the parameter estimation on the real software failure data sets.***

***Keywords**— Gompertz model, Maximum Likelihood Estimation, SRGM, AIC, Goodness Of Fit, Software Reliability.*

## I. INTRODUCTION

Software reliability is defined as the probability of failure-free software operation for a specified period of time in a specified environment (lyu, 1996). Software Reliability Growth Model (SRGM) is a mathematical model of how the software reliability improves as faults are detected and repaired (quadri, 2010). Among all SRGMs developed so far a large family of stochastic reliability models based on a Non-Homogeneous Poisson Process known as NHPP reliability models, has been widely used. Some of them depict exponential growth while others show S-shaped growth depending on nature of growth phenomenon during testing. The success of mathematical modelling approach to reliability evaluation depends heavily upon quality of failure data collected.

However, a problem is the model validation and selection. If the selected model does not fit the collected software testing data relatively well, we would expect a low prediction ability of this model and the decision makings based on the analysis of this model would be far from what is considered to be optimal decision (Xie, 2001). The present paper presents a method for model validation.

## II. LITERATURE SURVEY

### *A. NHPP Models*

The NHPP group of models provides an analytical framework for describing the software failure phenomenon during testing. They are proved to be quite successful in practical software reliability engineering (Musa, 1987). They have been built upon various assumptions. If 't' is a continuous random variable with probability density function: $f(t,\theta_1,\theta_2,\ldots,\theta_k)$, and cumulative distribution function: $F(t)$.where $\theta_1,\theta_2,\ldots,\theta_k$ are k unknown constant parameters. The mathematical relationship between the pdf and cdf is given as: $f(t)=F'(t)$.

Let $N(t)$ be the cumulative number of software failures by time 't'. A non-negative integer-valued stochastic process $N(t)$ is called a counting process, if $N(t)$ represents the total number of occurrences of an event in the time interval [0, t] and satisfies these two properties:

If $t_1 < t_2$, then $N(t_1) \le N(t_2)$

If $t_1 < t_2$, then $N(t_2) - N(t_1)$ is the number of occurrences of the event in the interval $[t_1, t_2]$.

One of the most important counting processes is the Poisson process. A counting process, $N(t)$, is said to be a Poisson process with intensity $\lambda$ if

1. The initial condition is N(0) = 0
2. The failure process, N(t), has independent increments.
3. The number of failures in any time interval of length s has a Poisson distribution with mean $\lambda s$, that is,

$$P\{N(t+s)-N(t)=n\}=\frac{e^{-\lambda s}(\lambda s)^n}{n!}$$

Describing uncertainty about an infinite collection of random variables one for each value of 't' is called a stochastic counting process denoted by $[N(t), t \ge 0]$. The process $\{N(t), t \ge 0\}$ is assumed to follow a Poisson distribution with characteristic Mean Value Function $m(t)$, representing the expected number of software failures by time 't'. Different models can be obtained by using different non decreasing $m(t)$. The derivative of $m(t)$ is called the failure intensity function $\lambda(t)$.

A Poisson process model for describing about the number of software failures in a given time (0, t) is given by the probability equation.

$$P\left[N(t)=y\right]=\frac{e^{-m(t)}[m(t)]^{y}}{y!}, \quad y=0,1,2,...$$

Where, $m(t)$ is a finite valued non negative and non decreasing function of $'t'$ called the mean value function. Such a probability model for $N(t)$ is said to be an NHPP model. The mean value function $m(t)$ is the characteristic of the NHPP model.

The NHPP models are further classified into Finite and Infinite failure models. Let 'a' denote the expected number of faults that would be detected given infinite testing time in case of finite failure NHPP models. Then, the mean value function of the finite failure NHPP models can be written as: $m(t) = aF(t)$. The failure intensity function $\lambda(t)$ is given by: $\lambda(t) = aF'(t)$ (Pham, 2006).

### *B. SRGM*

SRGMs are a statistical interpolation of defect detection data by mathematical functions (Wood, 1996). They have been grouped into two classes of models-Concave and S-shaped. The only way to verify and validate the software is by testing. This involves running the software and checking for unexpected behaviour of the software output (kapur, 2009). SRGMs are used to estimate the reliability of a software product. In literature, we have several SRGMs developed to monitor the reliability growth during the testing phase of the software development. Software reliability is defined as the probability of failure-free software operation for specified period of time 't' in a specified environment,

$$R(t) = e^{-m(t)}.$$

### *C. Gompertz Software Reliability Model*

The simplest form of a software reliability growth model is an exponential one. However, S-shaped software reliability is more often observed than the exponential one. Some models use a non-homogeneous Poisson process (NHPP) to model the failure process. The NHPP is characterized by its expected value function, m(t). This is the cumulative number of failures expected to occur after the software has executed for time t. Gompertz SRGM is based on an NHPP. In fact, many Japanese computer manufacturers and software houses have applied the Gompertz curve model, which is one of the simplest S-shaped software reliability growth models (Kececioglu, 1991). The Gompertz curve model gave good approximation to cumulative number of software faults observed (Satoh, 2000). It takes the number of faults per unit of time as independent Poisson random variables. The Gompertz model equation for software reliability is,

$$m(t) = ab^{c^{t}}$$

Where, 'a' is the upper limit approached the reliability, R at time t. $0<b<1$, $0<c<1$ are parameters to be estimated form any one of the parameter estimation methods.

a is the expected total number of failures that would occur if testing was infinite.

b is the rate at which the failures detection rate decreases.

c models the growth pattern (small values model rapid early reliability growth, and large values model slow reliability growth).

The Gompertz distribution plays an important role in modeling survival times, human mortality and actuarial tables. According to the literature, the Gompertz distribution was formulated by Gompertz (1825) to fit mortality tables. Recently, many authors have contributed to the statistical methodology and characterization of this distribution. For example, Read (1983), Gordon (1990), Makany (1991), Franses (1994) and Wu & Lee (1999). Garg *et al*. (1970) studied the properties of the Gompertz distribution and obtained the maximum likelihood estimates for the parameters. There are several forms for the Gompertz distribution given in the literature. Some of these are given in Johnson *et al*. (1994). Gompertz software reliability model is a popular model to estimate remaining failures. It has been widely used to estimate software error content, it is a modified model of Moranda reliability model.

## III. PARAMETER ESTIMATION OF THE MODEL

There are two methods of parameter determination. Parameter prediction and parameter estimation. Parameter prediction tries to establish the parameters of a model from the properties of the software product and the development process. Parameter estimation is used in subsystem or system test or operational phase where failure data are available. It is a statistical method trying to estimate model parameters based on failure times. A number of procedures can be used to estimate the parameters of Gompertz reliability model. Among these methods, The maximum likelihood estimation has been frequently considered to estimate the parameters of the Gompertz model.

 

The likelihood function of the sample is given by

$$L = e^{-m(t_n)} \prod_{i=1}^{n} \lambda(t_i)$$
$$L = e^{-ab^{c^{t_n}}} \prod_{i=1}^{n} \left[ ab^{c^{t_i}} c^{t_i} \log b \log c \right] \quad 1$$

It is usually easier to maximize the natural logarithm of the likelihood function rather than the likelihood function itself. So, the natural logarithm of the likelihood function can be written as:

$$\log L = \sum_{i=1}^{n} \left[ \log a + \log\left(b^{c^{t_i}}\right) + \log\left(c^{t_i}\right) + \log(\log b) + \log(\log c) \right] - ab^{c^{t_n}} \quad 2$$

The first derivatives of the natural logarithm of the total likelihood function in (2) with respect to a, b and c are given by:

$$\frac{\partial \log L}{\partial a} = \frac{n}{a} - ab^{c^{t_n}} \quad 3$$
$$\frac{\partial \log L}{\partial b} = nc^{t_n} - \sum_{i=1}^{n} c^{t_i} - \frac{n}{l\, \text{og}\, b} \quad 4$$
$$\frac{\partial \log L}{\partial c} = \frac{1}{c}\left( \frac{n}{\log c} + \sum_{i=1}^{n} t_i + \sum_{i=1}^{n} t_i c^{t_i} \log b \right) - n \log b t_n c^{t_n - 1} \quad 5$$

By equating equation (3) and (4) to zero, the maximum likelihood estimate of a and b can be given by the following estimation equation:

$$a = \frac{n}{b^{c^{t_n}}} \quad 6$$
$$b = e^{\frac{n}{nc^{t_n} - \sum_{i=1}^{n} c^{t_i}}} \quad 7$$

Substituting 'b' in the equation (5), we get

$$\frac{\partial \log L}{\partial c} = \frac{1}{c}\left( \frac{n}{\log c} + \sum_{i=1}^{n} t_i + \sum_{i=1}^{n} t_i c^{t_i} \frac{n}{nc^{t_n} - \sum_{i=1}^{n} c^{t_i}} \right) - \frac{n^2}{nc^{t_n} - \sum_{i=1}^{n} c^{t_i}} t_n c^{t_n - 1} \quad 8$$

Obviously, it is very difficult to obtain a closed-form solution, iterative procedures must be used to solve these equations, numerically. The Newton-Raphson method is used to obtain the MLE of 'c'. Therefore, take the 2$^{nd}$ derivative with respect to 'c' and equating it to Zero.

$$\frac{\partial^2 \log L}{\partial c^2} = \frac{1}{c}\left( \sum_{i=1}^{n} t_i^2 c^{t_i - 1} \frac{n}{nc^{t_n} - \sum_{i=1}^{n} c^{t_i}} + \sum_{i=1}^{n} t_i c^{t_i} \frac{-n}{\left(nc^{t_n} - \sum_{i=1}^{n} c^{t_i}\right)^2}\left( nt_n c^{t_n - 1} - \sum_{i=1}^{n} t_i c^{t_i - 1} \right) - \frac{n}{c(\log c)^2} \right)$$
$$- \frac{1}{c^2}\left( \sum_{i=1}^{n} t_i c^{t_i} \frac{n}{nc^{t_n} - \sum_{i=1}^{n} c^{t_i}} + \sum_{i=1}^{n} t_i + \frac{n}{\log c} \right)$$
$$- n^2 t_n \frac{\left[ \left( nc^{t_n} - \sum_{i=1}^{n} c^{t_i} \right)(t_n - 1)c^{t_n - 2} - c^{t_n - 1}\left( nt_n c^{t_n - 1} - \sum_{i=1}^{n} t_i c^{t_i - 1} \right) \right]}{\left( nc^{t_n} - \sum_{i=1}^{n} c^{t_i} \right)^2}$$

Thus, once the value of c^ is determined, an estimate of b^ is easily obtained from (7). They are then substituted in equation (6) to get an estimate of a^.

## IV. GOODNESS-OF-FIT

Model comparison and selection are the most common problems of statistical practice, with numerous procedures for choosing among a set of models proposed in the literature. Goodness-of-fit tests for this process have been proposed by (Rigdon, 1989). The AIC is a measure of the relative quality of a statistical model, for a given set of data. As such, AIC provides a means for model selection. AIC deals with the tradeoff between the goodness of fit of the model and the complexity of the model.

$$AIC = -2 * L + 2 * k \quad (2)$$

Where 'k' is the number of parameters in the statistical model, and 'L' is the maximized value of the likelihood function for the estimated model.

Given a set of candidate models for the data, the preferred model is the one with the minimum AIC value. Hence AIC not only rewards goodness of fit, but also includes a penalty that is an increasing function of the number of estimated parameters. This penalty discourages over-fitting. The AIC and the Reliability for 6 real life failure data sets for the power law process model is given in table 4.1 as follows.

Table 4.1: AIC and Reliability for Gompertz model

| **Data Set** | **No. of samples** | **AIC** | **$R(t_n)$** |
|---|---|---|---|
| **XIE** | **30** | 67.147946 | 0.999316 |
| **NTDS** | **26** | -134.186978 | 0.000982 |
| **IBM** | **15** | -36.256032 | 0.357614 |
| **ATT** | **22** | 57.347974 | 0.998679 |
| **SONATA** | **30** | 1472.473043 | 1.000000 |
| **LYU** | **24** | -163.747207 | 0.000000 |

## V. ANALYSIS

The performance of SRGM is judged by its ability to fit the software failure data. The term goodness of fit denotes the question of "How good does a mathematical model fit to the data?" Inorder to validate the model under study and to assess its performance, experiments on a set of actual software failure data have been performed. The Data set with the highest AIC value is having the high Reliability and the Data set with lowest AIC value is having the low Reliability at time '$t_n$' i.e $R(t_n)$.

## VI. CONCLUSION

To validate the proposed approach, the parameter estimation is carried out on the data sets collected from different sources. Parameters of the model are estimated by MLE method using cumulative failure data against time. It is observed that with the considered model the data set having AIC value high is exhibiting high Reliability and the data set having AIC value low is exhibiting low Reliability at $n^{th}$ time failure.

**REFERENCES**

[1] Lyu, M.R., Handbook of Software Reliability Engineering, McGraw-Hill, New York. 1996.

[2] Quadri, S.M.K and Ahmad, N. Software Reliability Growth modelling with new modified Weibull testing-effort and optimal release policy, International Journal of Computer Applications, 2010; Vol.6, No.12.

[3] Xie, M., Yang, B. Gaudoin, O. Regression goodness-of-fit Test for Software Reliability Model Validation, ISSRE and Chillarege Corp, 2001.

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

[5] Pham. H., System software reliability, Springer, 2006.

[6] Wood, A., Software Reliability Growth Models, Tandem Computers, Technical report 96.1, 1996.

[7] Kapur, P.K., Sunil kumar, K., Prashant, J. Ompal, S. Incorporating concept of two types of imperfect debugging for developing flexible software reliability growth model in distributed development environment, Journal of Technology and Engineering sciences, 2009; Vol.1, No.1; Jan-Jun.

[8] Xie, M., Goh. T.N., Ranjan.P., "Some effective control chart procedures for reliability monitoring" -Reliability engineering and System Safety, 2002; 77; 143 -150.

[9] Ashoka. M. Sonata Software Limited Data Set, Bangalore, 2010.

[10] Satoh, D. (2000). "A discrete Gompertz equation and a software reliability growth model". IEICE TRANSACTIONS on Information and Systems, 83(7), 1508-1513.

[11] Kececioglu, D. (1991). "Reliability Engineering Handbook, Vol. 2, Prentice-Hall, Englewood Cliffs, N.J., 1991.

[12] Gompertz, B. (1825). "On the Nature of the Function Expressive of the Law of Human Mortality and on the New Mode of Determining the Value of Life Contingencies", Phil. Trans. R. Soc.A., 115, 513-580.

[13] Read, C.B. (1983). "Gompertz Distribution", Encyclopedia of Statistical Sciences, Wiley, New York.

[14] Gordon, N.H. (1990). "Maximum Likelihood Estimation for Mixtures of two Gompertz Distributions when Censoring Occurs", Communication in Statistics – Simulation & Computation, 19, 733-747.

[15] Makany, R. (1991), "A Theoretical Basis of Gompertz's Curve", Biometrical Journal, 33, 121-128.

[16] Franses, P.H. (1994). "Fitting a Gompertz Curve", Journal of the Operational Research Society, 45, 109-113.

[17] Wu, J.W. and Lee, W.C. (1999). "Characterization of the Mixtures of Gompertz Distributions by Conditional Expectation of Order Statistics", Biometrical Journal, 41, 371-381.

[18] Garg, M.L., Rao, B.R. and Redmond, C.K. (1970). "Maximum Likelihood Estimation of the Parameters of the Gompertz Survival Function", Journal of the Royal Statistical Society C, 19, 152-159.

[19] Johnson, N.L., Kotz, S. and Balakrishnan, N. (1994). "Continuous Univariate Distributions", Vol. 1, 2nd ed., John Wiley and Sons, (pp. 25-26, 81-85).

[20] Rigdon, S. E. and Basu, A. P. The Power Law Process: a Model for the Reliability of Repairable Systems. Journal of Quality Technology, 1989; 21(4); pp.251- 259.

 

## ABOUT AUTHOR

**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 Associative Professor in the Department of Computer Science & Engineering, Acharya Nagarjuna University. His current research is focused on Software Engineering. He published 80 research papers in National & International Journals.

**V. Suryanarayana** received M.Tech. degree from JNTUK, Kakinada and working as Assoc. Prof. in CSE Department in Ramachandra College of Engineering, Eluru. He is pursuing Ph.D. in Computer Science & Engineering in Acharya Nagarjuna University, Nagarjuna Nagar. His area of Research is Software Reliability Engineering.

**Dr. G. Krishna Mohan,** working as Associate Professor in the Department of Computer Science & Engineering, KL University. He obtained his M.C.A degree from Acharya Nagarjuna University, M.Tech(CSE) from Jawaharlal Nehru Technological University, Kakinada, M.Phil from Madurai Kamaraj University and Ph.D(CSE) from Acharya Nagarjuna University. He qualified, AP State Level Eligibility Test. His research interests lies in Data Mining and Software Engineering. He published 35 research papers in various National and International journals.

 