Journal of Theoretical and Applied Information Technology
20th April 2016. Vol.86. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

GOMPERTZ BASED SPRT: MLE

[1]Dr. R. Satya Prasad, [2]V. Suryanarayana, [3]Dr. G. Krishna Mohan
1Associate Professor, Dept. of CSE, Acharya Nagrjuna University, Guntur.
2Associate Professor, CSE, Ramachandra College of Engineering.
3Associate Professor, Dept. of CSE, KL University, Vaddeswaram, India.
E-mail: [1]profrsp@gmail.com, [2]s_vadhri@yahoo.co.in, [3]gvlkm@kluniversity.in

ABSTRACT

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. The parameters are estimated using Maximum Likelihood Estimation (MLE). In the present paper, the Gompertz model is used on five sets of existing software reliability data and analyzed the results.

Keywords: *Gompertz, Sequential Probability Ratio Test, MLE, Decision lines, 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 were 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 upto 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

$$P\left[N\left(t\right)=n\right]=\frac{e^{-\lambda t}\left(\lambda t\right)^{n}}{n!} \qquad (1.1)$$

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 Gompertz 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 – Gompertz is presented in Section 3. Application of the decision rule to detect unreliable software with respect to the proposed SRGM is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

given in Section 4. Analysis of the application of the SPRT on five 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 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 'λ'. In our case, N(t)= number of failures up to time ' t' and ' λ' 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 ' λ'. We can not expect to estimate ' λ' precisely. But we want to reject the system with a high probability if our data suggest that the failure rate is larger than λ_1 and accept it with a high probability, if it's smaller than λ_0. As always with statistical tests, there is some risk to get the wrong answers. So we have to specify two (small) numbers 'α' and 'β', where 'α' 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. β 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 λ_0 and λ_1 such that $0 < \lambda_0 < \lambda_1$, the probability of finding N(t) failures in the time span (0,t) with λ_1,λ_0 as the failure rates are respectively given by

$$Q_1 = \frac{e^{-\lambda_1 t}\left[\lambda_1 t\right]^{N(t)}}{N(t)!} \qquad (2.1)$$

$$Q_0 = \frac{e^{-\lambda_0 t}\left[\lambda_0 t\right]^{N(t)}}{N(t)!} \qquad (2.2)$$

The ratio $\frac{Q_1}{Q_0}$ at any time 't' is considered as a measure of deciding the truth towards λ_0 or λ_1, given a sequence of time instants say $t_1 < t_2 < t_3 < .... < t_K$ and the corresponding realizations $N(t_1), N(t_2), ...N(t_K)$ 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 λ_1, in favor of λ_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

$$\frac{Q_1}{Q_0} \geq A \qquad (2.3)$$

$$\frac{Q_1}{Q_0} \leq B \qquad (2.4)$$

$$B < \frac{Q_1}{Q_0} < A \qquad (2.5)$$

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

Where 'α' and 'β' 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 $N_U(t) = a.t + b_2$ (2.6)

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

$$N_L(t) = a.t - b_1 \qquad (2.7)$$

To continue the test with one more observation on (t, N(t)) as the random graph of [t, N(t)] is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

between the two linear boundaries given by equations (2.6) and (2.7) where

$$a = \frac{\lambda_1 - \lambda_0}{\log\left(\frac{\lambda_1}{\lambda_0}\right)} \quad (2.8)$$

$$b_1 = \frac{\log\left[\frac{1-\alpha}{\beta}\right]}{\log\left(\frac{\lambda_1}{\lambda_0}\right)} \quad (2.9)$$

$$b_2 = \frac{\log\left[\frac{1-\beta}{\alpha}\right]}{\log\left(\frac{\lambda_1}{\lambda_0}\right)} \quad (2.10)$$

The parameters α, β, λ_0 and λ_1 can be chosen in several ways. One way suggested by Stieber (1997) is $\lambda_0 = \frac{\lambda.\log(q)}{q-1}$, $\lambda_1 = q\frac{\lambda.\log(q)}{q-1}$

$$where\ q = \frac{\lambda_1}{\lambda_0}$$

If λ_0 and λ_1 are chosen in this way, the slope of N_U (t) and N_L (t) equals λ. The other two ways of choosing λ0 and λ1 are from past projects and from part of the data to compare the reliability of different functional areas.

3. GOMPERTZ

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.

where

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.

4. SEQUENTIAL TEST FOR SRGMS

In Section II, for the Poisson process we know that the expected value of N(t) = λt 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 m(t) as its mean value function the probability equation of a such a process is

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

Depending on the forms of m(t) we get various Poisson processes called NHPP.

We may write

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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

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

Where, $m_1(t)$, $m_0(t)$ are values of the mean value function at specified sets of its parameters indicating reliable software and unreliable software respectively. Let P_0, P_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 $m_0(t) < m_1(t)$. Then the SPRT procedure is as follows:

Accept the system to be reliable if $\frac{Q_1}{Q_0} \le B$

i.e., $\frac{e^{-m_1(t)}.\left[m_1(t)\right]^{N(t)}}{e^{-m_0(t)}.\left[m_0(t)\right]^{N(t)}} \le B$

i.e., $N(t) \le \frac{\log\left(\frac{\beta}{1-\alpha}\right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)}$ (4.1)

Decide the system to be unreliable and reject if $\frac{Q_1}{Q_0} \ge A$

i.e., $N(t) \ge \frac{\log\left(\frac{1-\beta}{\alpha}\right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)}$ (4.2)

Continue the test procedure as long as

$$\frac{\log\left(\frac{\beta}{1-\alpha}\right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} < N(t) < \frac{\log\left(\frac{1-\beta}{\alpha}\right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} \quad (4.3)$$

Substituting the appropriate expressions of the respective mean value function – m(t) of Gompertz, we get the respective decision rules and are given in followings lines

Acceptance region:

$$N(t) \le \frac{\log\left(\frac{\beta}{1-\alpha}\right) + a\left[b_1^{c^t} - b_o^{c^t}\right]}{\log\left[\frac{b_1^{c^t}}{b_o^{c^t}}\right]} \quad (4.4)$$

Rejection region:

$$N(t) \ge \frac{\log\left(\frac{1-\beta}{\alpha}\right) + a\left[b_1^{c^t} - b_o^{c^t}\right]}{\log\left[\frac{b_1^{c^t}}{b_o^{c^t}}\right]} \quad (4.5)$$

Continuation region:

$$\frac{\log\left(\frac{\beta}{1-\alpha}\right) + a\left[b_1^{c^t} - b_o^{c^t}\right]}{\log\left[\frac{b_1^{c^t}}{b_o^{c^t}}\right]} < N(t) < \frac{\log\left(\frac{1-\beta}{\alpha}\right) + a\left[b_1^{c^t} - b_o^{c^t}\right]}{\log\left[\frac{b_1^{c^t}}{b_o^{c^t}}\right]}$$

(4.6)

It may be noted that in the above model the decision rules are exclusively based on the strength of the sequential procedure (α,β) 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) = λt 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 (Xie, 2002), (Pham, 2006) and (LYU,1996). The procedure adopted in estimating the parameters is a MLE. 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 $b_0 < b < b_1$.

Journal of Theoretical and Applied Information Technology
20th April 2016. Vol.86. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Assuming the value of $\delta = 0.0125$, the choices are given in the following table.

Table 5.1: Estimates Of A, B, C & Specifications Of B_0, B_1 For Time Domain

Data Set	Estimated Parameters			b_0	b_1
	A	B	C		
XIE	30.526286	0.055202	0.500320	0.042702	0.067702
NTDS	26.632869	0.013822	0.125836	0.001322	0.026322
IBM	16.419633	0.045773	0.303497	0.033273	0.058273
ATT	22.734515	0.063920	0.521470	0.051420	0.076420
SONATA	37.225335	0.033200	0.860221	0.020700	0.045700
LYU	25.201579	0.034256	0.002503	0.021756	0.046756

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 4.4 and 4.5, sequentially at each 't' of the data sets taking the strength (α, β) 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.

Table 5.2: SPRT Analysis For 5 Data Sets Of Time Domain Data

Data Set	T	N(t)	Acceptance region (≤)	Rejection Region (≥)	Decision
Xie	0.3002	1	-2.826793	59.737204	Reject
	0.3146	2	-2.206270	57.494022	
	0.5393	3	1.350998	36.177091	
	0.5529	4	1.386293	35.355749	
	0.5872	5	1.435542	33.420746	
	0.7192	6	1.279444	27.394171	
	0.7707	7	1.134477	25.504157	
	0.809	8	1.011414	24.227375	
	1.019	9	0.262405	18.693918	
	1.1487	10	-0.176396	16.174011	
	1.1534	11	-0.191377	16.092403	
	1.2157	12	-0.382987	15.066311	
	1.2496	13	-0.481615	14.548564	
	1.3407	14	-0.726378	13.282507	
NTDS	0.0900	1	-20.864556	106.973442	Accept
	0.2100	2	3.548739	58.336452	
AT&T	0.0550	1	-116.115607	264.981855	Reject
	0.0733	2	-82.431467	203.521608	
	0.1008	3	-55.159526	152.780558	
	0.8097	4	-2.245358	23.641218	
	0.8491	5	-2.203396	22.481989	
	0.9989	6	-2.168049	18.815393	
	1.0336	7	-2.177009	18.101978	
	1.1332	8	-2.219902	16.276710	
	1.2471	9	-2.283002	14.524279	
	1.4459	10	-2.388322	12.108089	
	1.5240	11	-2.421583	11.331935	
	1.6700	12	-2.467655	10.083459	
IBM	0.10	1	-76.678726	177.954249	Reject
	0.19	2	-34.503401	99.513954	
	0.32	3	-16.503745	63.069060	
	0.43	4	-10.379356	48.837615	
	0.58	5	-6.315527	37.586710	
	0.70	6	-4.625284	31.750855	
	0.88	7	-3.273048	25.662517	
	1.03	8	-2.700683	22.020965	
	1.25	9	-2.297923	18.072715	
	1.50	10	-2.134507	14.841024	
	1.69	11	-2.100168	12.966872	
	1.99	12	-2.101447	10.694180	
LYU	0.005	1	-162711.453713	289601.538220	Continue
	0.017	2	-47838.523941	85194.708983	
	0.045	3	-18056.658666	32200.340438	
	0.072	4	-11275.970606	20134.653834	
	0.100	5	-8111.652855	14503.996742	
	0.130	6	-6233.928743	11162.724793	
	0.148	7	-5472.690505	9808.153818	
	0.157	8	-5157.528513	9247.343842	
	0.171	9	-4733.217259	8492.308821	
	0.206	10	-3924.761943	7053.708736	
	0.240	11	-3365.197489	6057.989843	
	0.252	12	-3203.754998	5770.709128	

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	0.261	13	-3092.415506	5572.584340	
	0.278	14	-2901.777007	5233.348747	
	0.292	15	-2761.448346	4983.637132	
	0.319	16	-2525.598911	4563.946417	
	0.351	17	-2293.060546	4150.144468	
	0.376	18	-2138.931257	3875.869168	
	0.396	19	-2029.640071	3681.382554	
	0.441	20	-1819.980863	3308.284351	
	0.476	21	-1684.320294	3066.866596	
	0.528	22	-1515.979711	2767.287258	
	0.600	23	-1331.067266	2438.207667	
	0.707	24	-1125.847320	2072.971576	
SONATA	0.525	1	3.411063	15.519451	*Accept*

From the Table 5.2, 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 of Sequential Probability Ratio Test with Gompertz as exemplified for five Data Sets indicates that the model is performing well in arriving at a decision. The model has given a decision of rejection for 3 Data Sets i.e. Xie, AT&T and IBM at 14th , 12th and 12th instances respectively, a decision of continue for 1 Data Set i.e. LYU, and a decision of accept for 2 Data sets i.e NTDS and SONATA at 2nd and 1st instance. Therefore, we may conclude that, applying SPRT on data sets we can come to an early conclusion of reliability / unreliability of software.

REFERENCES

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

[2] 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.

[3] 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.

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

[5] 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).

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

[7] Lyu, M.R. (1996). "The hand book of software reliability engineering", *McGrawHill & IEEE Computer Society press*.

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

[9] Pham. H., (2006). "System software reliability", Springer.

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

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

[12] Satyaprasad (2007)."Half logistic Software reliability growth model "*Ph.D Thesis of ANU*, India.

[13] Stieber, H.A. (1997). "Statistical Quality Control: How To Detect Unreliable Software Components", *Proceedings the 8th International Symposium on Software Reliability Engineering*, 8-12.

[14] Wald. A., 1947. "Sequential Analysis", *John Wiley and Son*, Inc, New York.

[15] 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.

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