SOFTWARE RELIABILITY ESTIMATION

[1] Kavita, [2]Dr. Pardeep Saini
[1]Research Scholar, [2]Supervisor
[1-2] Department of Computer Science & IT, Sunrise University, Alwar, Rajasthan

ABSTRACT

Software reliability growth models have underlying assumptions that are often violated in practice, but empirical evidence has shown that many of them are quite robust despite these violations. The problem is that the evaluation of models often makes it difficult to know which models to use in practice. We present an empirical SRGM selection method that provides guidance on how to choose between SRGMs to decide the best model to use as failures are reported during a testing phase. They are fitted to cumulative failure data and used to estimate software reliability.

INTRODUCTION

A number of software reliability growth models (SRGMs) exist to estimate the expected number of remaining failures. The problem with using SRGMs to estimate failure content is that they have underlying assumptions that are often violated in practice. Despite the fact that empirical evidence has shown that many of the models are quite robust in practice, it is often difficult to decide which model to use in light of these violations of assumptions. The “best model” can vary between systems and even versions. We cannot expect to pick one model and use it successfully for later versions or other systems. The underlying problem is the complexity of factors that interact and affect software reliability (Cai, 1991) considers four factors once the nature of the test, the characteristics of the software, human intervention and the debugging process.

A possible strategy is to use a number of models once testing has progressed in the test plan to the point where it makes sense to worry about whether to stop testing. This will work if there is a systematic strategy to identify the best models during testing and determine when to stop. This work proposes a selection method which is able to determine the best model(s) for estimating the total number of defects in software. From this estimate, reliability can be calculated and used as a decision aid.

SOFTWARE RELIABILITY MODELS

There are both static and dynamic software reliability models to assess the software quality aspect. A static model uses software metrics such as complexity metrics, inspection results, etc. to estimate the number of defects (or bugs) in the software. The dynamic model uses the past failure rate during startup or the cumulative failure profile over time to estimate the number of failures. It includes a time component, usually the time between failures.

Software reliability growth models predict the number of failures at time t. Many SRGMs have two or three parameters. A popular model for estimating residual faults is the Gompertz model. The Gompertz model is a three-parameter model. The failure process is a non-homogeneous process; that is, the characteristics of the probability distribution change over time as it is widely used to estimate software error content. The Gompertz model is an S-shaped model. It is given by the following equation:

$$m(t) = a(b^t)^c a \geq 0,\ 0 \leq b \leq 1,\ 0 \leq c$$

Where

a = expected total number of failures that would occur if testing were infinite

b = rate at which the failure detection rate decreases

c = models the growth pattern (small values model fast early confidence growth and large values model slow confidence growth)

Some models use a non-homogeneous Poisson process (NHPP) to model the failure process. The NHPP is characterized by the expected value function m(t). This is the cumulative number of expected failures after running the software for time t. m(t) is non-decreasing as t increases. The Gompertz model is based on the NHPP.

MAXIMUM LIKELIHOOD ESTIMATION

In much of the literature, the preferred method of obtaining parameter estimates is to use maximum likelihood. Probability equations are derived from model equations and the assumptions

underlying those equations. Parameters are then considered to be the values that maximize these probability functions.

Using the partial derivative of the likelihood function with respect to the model parameters, the maximum likelihood equation is obtained and set to zero. Iterative routines are then used to solve these equations.

If we perform an experiment and obtain N independent observations $t_1, t_2, t_3 \ldots t_n$. The probability density function is given as $f(t, \theta_1, \theta_2 \ldots \theta_k)$.

The likelihood function is:

$$L = \prod_{i=1}^{n} f(t_i)$$

Taking the natural logarithm

$$\log L = \log \left(\prod f(t_i)\right)$$

MLE estimates of unknown parameters are obtained by maximizing log L with respect to parameters.

$$\frac{\partial(\log L)}{\partial \theta_j} = 0 j = 1,2,\ldots k$$

PARAMETER ESTIMATION (GOMPERTZ MODEL)

The likelihood function of the Gompertz model is

$$L = e^{-ab^c} \prod_{i=1}^{n} a\, b^c e^t \log b \log c$$

Taking natural logarithm

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

The first derivatives of the log likelihood function with respect to parameters are

$$\frac{\partial \log L}{\partial a} = \frac{n}{a} - b^c$$

$$\frac{\partial \log L}{\partial b} = nc^t - \sum c^t - \frac{n}{\log b}$$

$$\frac{\partial \log L}{\partial c} = \frac{n}{\log c} - \sum t_i$$

Equating these equations to zero gives estimates of parameters.

$$a = \frac{n}{b^c}$$

$$b = e^{-\sum t_i}$$

Because it is difficult to obtain closed-form solutions, iterative procedures such as the Newton–Raphson method are used to obtain parameter estimates.

STABILITY AND RELIABILITY

Comparing and choosing models are common statistical problems with selection procedures. Goodness-of-fit tests have been proposed for this purpose. AIC (Akaike Information Criterion) provides a measure of model selection.

$$AIC = -2\log L + 2k$$

Where

k = number of parameters in the statistical model

L = maximum likelihood value

The model with the **minimum AIC value** is preferred.

TABLE: PARAMETER ESTIMATES OF THE MODEL

Data Set	Samples	a	b	c	R(t)	AIC
NTE	30	30.526286	0.055202	0.500320	0.999118	67.147946
NTDS	26	26.613869	0.013832	0.125836	0.000092	-114.118978
IBM	15	16.419633	0.045773	0.303497	0.357614	-36.258032
ATT	22	22.734515	0.063920	0.521470	0.986769	57.147941
SONATA	30	37.225335	0.033200	0.860121	1.000000	147.147343
LIVE	24	25.201579	0.034256	0.002983	0.000000	-163.747287

ANALYSIS

The performance of SRGM is judged by its ability to adapt to software failure data. The term goodness-of-fit refers to the question: how well does the mathematical model fit the data? In order to validate the model and assess its performance, experiments were conducted on a set of real software failure data. The data set with the highest AIC value has high confidence and the data set with the lowest AIC value has low confidence.

CONCLUSION

To validate the proposed approach, parameter estimation is performed on different software failure data sets. Model parameters are estimated by MLE using cumulative time-dependent failure data. It is observed that with the considered model, the data set with high AIC value shows high confidence and the data set with low AIC value shows low confidence at the nth failure.

REFERENCES

1. Ashoka M. Sonata Software Limited Data Set, Bangalore, 2010.
2. Frances P.H. (1994). Fitting a Gompertz Curve. Journal of the Operational Research Society.
3. Garg, M.L., Rao, B.R., and Redmond, C.K. (1970). Maximum Likelihood Estimation of the Parameters of the Gompertz Survival Function.
4. Gompertz, B. (1825). On the Nature of the Function Expressive of the Law of Human Mortality.
5. Gordon, N. (1990). Maximum Likelihood Estimation for Mixtures of Gompertz Distributions.
6. Johnson, N.L., Kotz, S., and Balakrishnan, N. (1994). Continuous Univariate Distributions.
7. Kapur, P.K., Kumar, S., and Prajapati, U. (1994). Flexible Software Reliability Growth Model.
8. Kececioglu, D. (1991). Reliability Engineering Handbook.
9. Lyu, M.R. (1996). Handbook of Software Reliability Engineering.
10. Makany, R. (1991). Theoretical Basis of Gompertz Curve.
11. Musa, J.D. (1987). Software Reliability Measurement Prediction Application.
12. Pham, H. (2006). System Software Reliability.
13. Quadri, S.M.K., and Ahmad, N. (2010). Software Reliability Growth Modelling.
14. Read, C.R. (1983). Gompertz Distribution.
15. Rigdon, S.E., and Basu, A.P. (1994). Reliability of Repairable Systems.
16. Satoh, D. (2000). Discrete Gompertz Model.
17. Wood, A. (1996). Software Reliability Growth Models.
18. Wu, J.W., and Lee, W.C. (1999). Characterization of the Structure of Gompertz Distributions.
19. Xie, M., Goh, T.N., and Ranjan, P. (2002). Reliability Engineering and System Safety.
20. Xie, M., Yang, B., and Goh, T.N. (2001). Regression Techniques for Software Reliability Model Validation.