The 7th International Symposium on Management, Engineering and Informatics: MEI 2011 in the context of The 15th Multi-conference on Systemics, Cybernetics and Informatics: WMSCI 2011. July 19 - 22, 2011. Orlando, FL.

Revisiting the performance of mixtures of software reliability growth models

Peter A. Keiller[1], Charles J. Kim[1], John Trimble[1], and Marlon Mejias[2]

[1] **Department of Systems and Computer Science**
Howard University
Washington D.C. 20059

[2] **Department of Engineering Management and Systems Engineering**
George Washington University
Washington D.C. 20052

Abstract

In this paper, we evaluate the performance of a mixture of software reliability growth models (Super models) against the original base models. The base models chosen are four well known nonhomogeneous Poisson process (NHPP) growth models proposed in literature. We use the method of maximum likelihood for fitting these models to data.

The Super models are based on the above base models plus a selection criterion based on past prediction measures.

Four Poisson type reliability models and four Super models are evaluated based on their performance on three data sets. The performance is judged by the relative error of the predicted number of failures over future time intervals relative to the number of failures eventually observed during the interval. Conclusions are drawn based on the analysis of the results and the use of the selection of models.

Key Words: Software reliability, Maximum Likelihood Method, Nonhomogeneous Poisson process

1. Introduction

Software has now become an integral part in industrial, financial, commercial, health, defense and real time safety critical systems. In a 2002 press release the National Institute of Science and Technology (NIST) estimated that 80% of development costs involve identifying and correcting bugs. It is estimated that costs related to software failure amounted to over 70 million US dollars. With the the increase of complexity of software systems. managing the quality of software in projects has never been more apparent The use of Software Relaibility Growth Models (SRGMs) can address this management problem by allowing the user to as accurately as possible predict the current and future reliability of the software on test.

Research using NHPP based software models have shown that these models do perform creditably when compared to expert opinion [1]. The problem that arise is that the estimates of these models however tend to either consistently overestimate or underestimate the quality of the software [2].

In this research we propose the use of four commonly referenced NHPP models and four mixture of models or Super models. This paper is a continuation of the work done by Keiller and Miller [3].

Our reliability growth models on test include several of the well known models in literature together with additional models that we call 'Super models'. These Super models are based on a set of the usual reliability growth models plus a selection criterion, which identifies one of the set to use for predictions at each point of time; the selection criterion is based on the 'quality-of-past prediction' measures. The object is to identify the best models or approaches to be use during the test phase.

2. Nonhomogeneous Poisson Process

"A Poisson process is a counting process characterized by its mean value function; and the cumulative number of software failures up to time t, N(t), can be described by the nonhomogeneous Poisson process (NHPP). We also know that the 'reliability growth process' can be represented as a counting process $\{N(t), 0 \le t\}$ where

$\{N(t) = \max\{i: T_i \le t\}, 0 \le t\}$, and T_i's are random variables and t_i's are real scalars. The process is observed for t: $0 \le t \le t_c$ where t_c is the "current time" and $(x_1, x_2, x_3, \ldots x_c)$ is a sequence of interfailure times- $x_i = t_{i+1} - t_i$.

If we let N(t) be the number of failures occurring in an arbitrary interval of time t, then for this counting process $\{N(t), 0 \le t\}$ modeled by the NHPP, N(t) follows a Poisson distribution with parameter M(t). M(t) is called the mean value function and describes the expected cumulative number of failures in (0,t)."[3]

Xie [4] and Lyu[5] outline the following information about the NHPP.

- The probability that N(t) is a given integer n is expressed by

$$P(N(t)= n) = \frac{(M(t))^n}{n!} e^{-M(t)}, \; n = 0,1,2,3,\ldots. \qquad (1)$$

- The function $\lambda(t)$ which is called the instantaneous

failure intensity is defined as

$$\lambda(t) = \lim_{\Delta t \to 0+} \frac{P(N(t+\Delta t) - N(t) > 0)}{\Delta t} \quad (2)$$

- If you are given $\lambda(t)$, the mean function $M(t) = E\{N(t)\}$ satisfies

$$M(t) = \int_0^t \lambda(s)ds . \quad (3)$$

- Reversing the process, if we know M(t), the failure intensity at time t can be obtained as

$$\lambda(t) = \frac{dM(t)}{dt} \quad (4)$$

- Also, the nonhomogeneous Poisson processes have an independent Poisson number of failures in disjoint intervals given by:

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

$$0 \le t,\ 0 \le s;\ n=0,1,2,3,\ldots \quad (5)$$

- By using different nondecreasing functions M(t), we get different NHPP models.

- The reliability function at time t_0 is :

$$R(t \mid t_0) = e^{-(M(t+t_0) - M(t_0))} = P(N(t_0 + t) - N(t_0)=0) \quad (6)$$

- The estimation of the unknown parameters in M(t) is obtained by using either the method of maximum likelihood or the method of least squares.

- The likelihood function for the NHPP model with mean value function M(t) is the following

$$L(n_1, n_2, \ldots, n_k) = \prod_{i=1}^{k} \frac{(M(s_{i-1}) - M(s_i))^{n_i} \exp(M(s_{i-1}) - M(s_i))}{n_i!} \quad (7)$$

where n_i denotes the number of faults detected in time interval (s_{i-1}, s_i), and $0<s_0 < s_1 < s_2< \ldots.< s_i$, $i \ge 0$, are the total monitored test times of the software.

- The parameters in M(t) are estimated by maximizing the likelihood function. Taking the natural log of the likelihood function, we use numerical algorithms to solve for the parameters [6].

3. Software Reliability Models on Test

3.1 Base Models

In this study the following four parametric families of NHPPs, characterized by their mean functions:

M1	Logarithmic	$M_1(t) = \gamma \log(1+\beta t)$	$0<\beta$
M2	Pareto	$M_2(t) = \gamma(1-(1+\beta t)^{-\alpha})$	$0<\alpha,0<\beta$
M3	Exponential	$M_3(t) = \gamma(1- e^{-\eta t})$	$0< \eta$
M4	General Power	$M_4(t) = \gamma((1+\beta t)^{-\alpha}-1)$	$-1<\alpha<0,0<\beta$ $0<\gamma$

are selected to represent one group of models on test (the simple models). Model M1 represents the Musa-Okumoto model [7]; model M2 represents the Littlewood NHPP model [8]; and model M3 represents the Goel-Okumoto model [9]. M4, the 'General Power' curve arises naturally when considering order statistics of independent but non-identically exponentially distributed failure times. Musa and Okumoto [7] have promoted the use of the NHPP models in software reliability growth modeling. Miller [10] also gives strong theoretical justification for using the NHPP.

3.2 Super Models

We consider four Super models. Keiller and Miller [3] defines a Super model as a set of parametric reliability growth models and a selection criterion; for a given software failure data set and for a given time, the selection criterion chooses the parametric model in the set that is to be used for making predictions of future failure behavior. As time passes for a given data set, a given super model may change its choice of parametric family to use for predictions.

Our procedure for a Super model is as follows: using maximum likelihood estimation, we fit all four of the parametric models (M1-M4). Next, the selection criterion picks one parametric class based on the fitted models. The current fitted model of the chosen class is used for making predictions at the current time.

We consider four pure quality-of-prediction measures (8): the u-plot, the y-plot, the prequential likelihood, and the maximum likelihood values. Using these criteria requires fitting each of the four NHPP models (M1-M4) after each failure and calculating the predictive distribution and density of the time until next failure.

To summarize, our four super models are all based on four NHPP models (M1-M4). The selection criteria for the four super models are:

M5	U-plot
M6	Y-plot
M7	Prequential likelihood
M8	Maximum likelihood

The four additional “mixture of models” were developed using dynamic selection among models based on goodness-of-fit and quality-of-prediction criteria such as

the u-plot, y-plot, prequential likelihood and the maximum likelihood values.

3. Failure Data Collection

The failure data used in this research is taken from the research by Almering et. al [11]. The data is taken from three software development projects for high-end TV sets containing several million lines of code. The data sets related to the projects are referred to as TV2003, TV2004, and TV2005. To give the reader a rough idea of the data, it is presented in an aggregate form in Table 1. The total cumulative time for each data set is split into 10 equal intervals and the cumulative number of failures occurring up to each of the 10 elapsed points is shown. From this table, it is possible to construct very rough plots of the reliability growth. Figures 3.1a -3.1c show the relationship between the cumulative interfailure times against the % test times (100% of the total test time of the software) of the tested data sets.

Figure 3.1a (Data set TV2003)

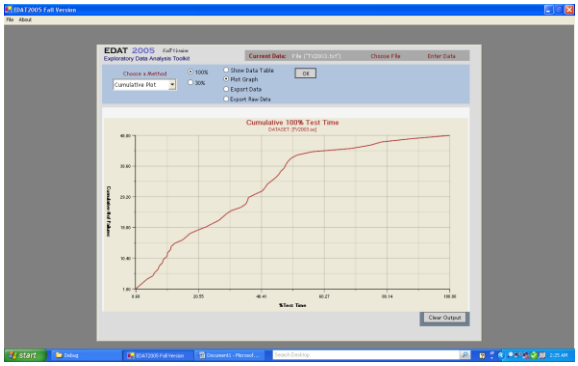

Figure 3.1b (Data set TV2004)

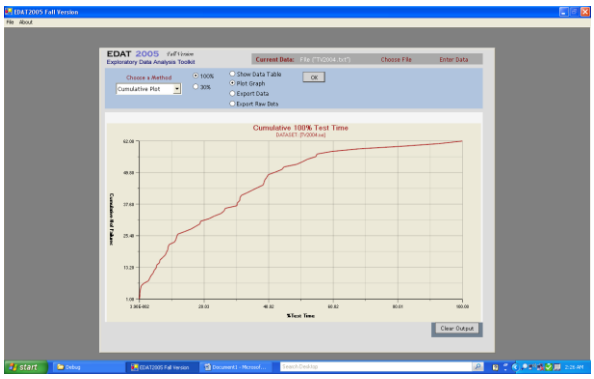

Figure 3.1c (Data set TV2005)

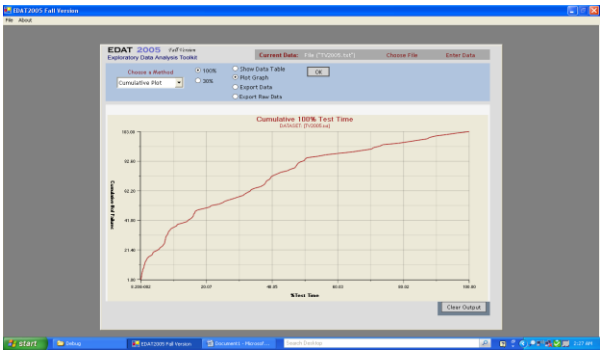

Table 1 Summary of Failure Data

SYSTEM	PERCENT OF TIME ELAPSED (# DEFECTS)									
	10	20	30	40	50	60	70	80	90	100
TV2003	10	18	24	30	41	43	44	46	47	48
TV2004	22	31	36	49	53	58	59	59	60	62
TV2005	36	50	58	72	83	87	90	95	100	103

4 Performance Measures

The performance measures are recapped from previous work [3].

4.1 Quality of Prediction

Let x_{i+1} be given as the interfailure time at time s_{i+1} (i.e. $x_{i+1} = s_{i+1} - s_i$). The first quality-of-prediction measure is the 'u- plot': it is well known that $U=F_x(X)$ has a uniform distribution on the interval [0,1]. Using this fact, if $\hat{F}_{i+1}(\cdot)$ is the true distribution of X_{i+1} then $u_{i+1} = \hat{F}_{i+1}(x_{i+1})$ will be an observation from U[0,1] distribution. Thus the empirical distribution formed from the u's should be closed to U[0,1]. If we observe n_o+n failures, starting to make predictions after the n_oth failure, the plot of

$$\{(u_{no+i}, i/(n+1)), i = 1,2,3,\ldots,n\}$$

is the u-plot. The maximum deviation of the u-plot from the identity function is a measure of quality-of-prediction.

The second measure of quality-of prediction is the 'y-plot': if the predictive distributions are good the u's should look like a random sequence of independent U[0,1] variables, and –log (1-u)'s like exponential random variates. In this case, let

$$y_i = \sum_{j=1}^{i} \log(1-u_{no+j}) \Big/ \sum_{j=1}^{n} \log(1-u_{no+j})$$

i=1,2,3,….,n

and plot the pairs $\{y_{i,}, i/(n+1)), i= 1,2,3,\ldots..,n\}$.

If the predictive distributions are good, this plot should be close to the identity function. A quantitative measure

of the quality-of-prediction is the maximum deviation between the y-plot and the identity function.

The third measures of quality-of-prediction is the prequential likelihood: based on Dawid's generalization of likelihood to a sequential situation , we have

$$PL_n = \prod_{i=1}^{n} \hat{f}_{no+i}; \text{ where } \hat{f}_{i+1}(s) = \frac{d}{ds}\hat{F}_{i+1}(s) \quad i \geq 0$$

For comparison purposes, the best predictive system should have the largest prequential likelihood. For a detailed discussion of these measures of quality-of-prediction, see Brocklehurst [12]. These three measures give a dynamic real-time evaluation of how well a given parametric model has done predicting interfailure times up to the present. Brocklehurst [12] argues that it would seem logical to calculate the next prediction using the model which has performed best up to the present on the particular software failure data set under consideration. These three measures give a basis for making this choice.

4.2 Goodness-of-Fit Measures

The performance of the model is investigated for each data set to see how well it can predict the number of new failures manifesting during finite future time intervals. The experiment is designed as a four step process. For each data set D_k , where k =1,..., 3:

(1) The MLE's of the model parameters are determined based on the first d data points of data set D_k.

(2) The fitted model is then used to predict the number of failures in $[s_{d},_k, s_{n,k}]$ where $s_{l,k} = \sum_{m=1}^{l} x_{m,k}$ and $x_{m,k}$ is the mth data point of data set D_k.

(3) The model is evaluated using

(i) predicted errors $e_{i,k} = \hat{n}_{i,k} - n_{i,k}$

(ii) relative errors $re_{i,k} = \dfrac{\hat{n}_{i,k} - n_{i,k}}{n_{i,k}}$

(iii) absolute relative errors $|re_{i,k}|$

(iv) relative error squared $(re_{i,k})^2$

where $\hat{n}_{i,k}$ $(n_{i,k})$ is the predicted (actual) number of failures in $[s_{10,k}, s_{n,k}]$.

(4) For each data set, the analysis (1)-(3) is performed for d = 10, ..., n_{k-1}. The value ten was selected arbitrarily but is large enough to enable meaningful estimation and small enough to allow for meaningful evaluation of the procedure. The sum of the absolute relative errors (ARE), the sum of the relative errors (RE), and the sum of the relative errors squared (RESQ) are normalized by averaging over the number of predictions for the model.

5. Findings

Table 2 shows the comparison ranking of the Base models (M1-M4) for each of the datasets on test using the u-plot, y-plot, PL, and ML performance measures.

Tables 3a-3c show the Base models chosen by Super Models 5-8 for each of the datasets on test.

Table 2 Ranking of Base Models

PERFORMANCE CRITERIA	DATASETS AND BASE MODELS ON TEST											
	TV2003				TV2004				TV2005			
	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4
U-PLOT	3	1	2	4	2	3	1	4	1	2	4	3
Y-PLOT	3	2	1	4	3	2	1	4	3	1	4	2
PL	3	2	1	4	2	3	1	4	2	3	4	1
ML	2	4	3	1	1	4	3	2	2	3	1	4

Table 3a Base Models Chosen by Super-Models M5-M8 **(SYSTEM ID: TV2003)**

SYSTEM ID: TV2003									
PERFORMANCE CRITERIA	PERCENT OF TIME ELAPSED								
	10	20	30	40	50	60	70	80	90
U-PLOT:M5	M2	M2	M2	M2	M2	M2	M2	M2	M2
Y-PLOT:M6	M1	M1	M2	M2	M2	M1	M3	M3	M3
PL:M7	M4	M3	M1	M4	M4	M1	M3	M3	M3
ML:M8	M2	M4	M4	M3	M1	M3	M4	M4	M4

Table 3b Base Models Chosen by Super-Models M5-M8 **(SYSTEM ID: TV2004)**

SYSTEM ID: TV2004									
PERFORMANCE CRITERIA	PERCENT OF TIME ELAPSED								
	10	20	30	40	50	60	70	80	90
U-PLOT	M3	M3	M3	M4	M3	M3	M3	M3	M3
Y-PLOT	M4	M3	M3	M4	M4	M3	M3	M3	M3
PL	M4	M4	M4	M4	M4	M4	M3	M3	M3
ML	M3	M3	M3	M3	M3	M3	M3	M3	M3

Table 3c Base Models Chosen by Super-Model M5-M8 **(SYSTEM ID: TV2005)**

SYSTEM ID: TV2005									
PERFORMANCE CRITERIA	PERCENT OF TIME ELAPSED								
	10	20	30	40	50	60	70	80	90
U-PLOT	M2	M3	M3	M1	M1	M1	M1	M1	M1
Y-PLOT	M4	M4	M4	M4	M4	M4	M4	M2	M2
PL	M4	M4	M4	M4	M4	M4	M4	M4	M4
ML	M3	M3	M3	M2	M3	M3	M3	M3	M3

Tables 4a-4c show the ranking of models (RE, ARE, and RESQ performance measures) for the datsets on test

Table 4a Ranking of models M1-M8
(SYSTEM ID: TV2003)

SYSTEM ID: TV2003								
PERFORMANCE MESURES	**MODELS ON TEST**							
	M1	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
RE	**7**	**1**	**4**	**8**	**2**	**6**	**5**	**3**
ARE	**7**	**2**	**4**	**8**	**3**	**5**	**6**	**1**
RESQ	**7**	**1**	**4**	**8**	**2**	**5**	**6**	**3**

Table 4b Ranking of models M1-M8
(SYSTEM ID: TV2004)

SYSTEM ID: TV2004								
PERFORMANCE MESURES	**MODELS ON TEST**							
	M1	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
RE	**4**	**3**	**1**	**8**	**6**	**5**	**7**	**2**
ARE	**2**	**1**	**3**	**6**	**7**	**8**	**5**	**4**
RESQ	**2**	**1**	**5**	**4**	**7**	**8**	**3**	**6**

Table 4c Ranking of models M1-M8
(SYSTEM ID: TV2005)

SYSTEM ID: TV2005								
PERFORMANCE MESURES	**MODELS ON TEST**							
	M1	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
RE	**5**	**4**	**3**	**8**	**2**	**7**	**6**	**1**
ARE	**4**	**1**	**8**	**6**	**3**	**7**	**2**	**5**
RESQ	**5**	**1**	**8**	**6**	**3**	**7**	**2**	**5**

6. Conclusion

In this experiment four NHPP models (M1-M4) and four Super models (M5-M8) are investigated. The experiment uses 3 data sets. We noticed that there were no significant performance improvements when using the Super models (M5-M8) compared to the Base models. We had expected one of the simple models (M1-M4) to be truly a superior fitting model and expected the model to be consistently chosen by the Super models (M5-M8). This did not happen. We expect to continue this study in a more controlled experiment based on Monte Carlo data and using model recalibration methods[12]..

7. Acknowledgement

The research was supported by Grant # NRC-27-10-1123.

REFERENCES

[1] Keiller, P.A., and Miller, D.R., "On the use and performance of software reliability growth models", in *Reliability Engineering and System Safety, Special Issue on Software*, Vol 32 Nos 1&2, 1991, 95-117.

[2] Keiller, P.A. and Mazzuchi, T. A., "Investigating a Specific Class of Software Reliability Growth Models". *Proceedings of the Annual Reliability and Maintainability Symposium*, 2002, 242-248.

[3] Keiller, P.A. and Mazzuchi, T. A., " Improving the Predictability of the Musa-Okumoto Software Reliability Growth Model". *The First International Software Assurance Certification Conference (ISACC'99)*,1999; B1- 3.

[4] Xie, M., *Software Reliability modelling*, World Scientific, Singapore,1991.

[5] Lyu, M., *Software Reliability Engineering Handbook*, IEEE Computer Press, 1996.

[6] Nelder, J. A. & Mead, R., "A simplex method for function minimization.", *Computer Journal*, 7 (1965) 308-313.

[7] Musa, J. D. and Okumoto, K., "A Logarithmic Poisson execution time model for software reliability measurement", *IEEE Proceedings of the 7th International Conference on Software Engineering*, 1984, 230-238.

[8] Abdala-Ghaly, A.A., Chan, P.Y. and Littlewood, B., "Evaluation of competing reliability predictions", *IEEE Transactions on Software Engineering*, SE-12, 1986, 950- 967.

[9] Goel, A. L. and Okumoto, K., "Time-dependent error-detection rate model for software reliability and other performance measures",*IEEE Transactions on Reliability*, R-28, 1979, 206-211.

[10] Miller, D. R., Exponential order statistic models of software reliability growth, CR-3909, National Aeronautics and Space Administration, July 1985, (Abridged version: IEEE Transaction on Software Engineering, SE-12, 1986, 12-24).

[11] Vincent Almering et al., " Using Software Reliability Growth Models in Practice", IEEE Software November/December 2007 (vol. 24 no.6) pp. 82-88.

[12] Brocklehurst, S., Chan, P.Y., Littlewood, B., and Shell, J., Recalibrating Software Reliability Models, IEEE Transaction of Software Engineering, Vol. 16, No. 4, 1990