International Journal of Pure and Applied Mathematics
Volume 26 No. 2 2006, 217-239

REGRESSION APPROACH TO SOFTWARE RELIABILITY MODELS

A.M. Mostafa[1], K.M. Ramachandran[2 §], A.N.V. Rao[3]
[1,2,3]Department of Mathematics/Statistics
University of South Florida
Tampa, FL 33620-5700, USA

Abstract: In this work, linear regression methods are applied to study the software reliability properties. In particular, predictive properties of the mean time to failure for modeling the patterns of software failure times are explained using the regression approach. These methods are applied to three different failure time data sets. The predictive accuracy of the mean time between failure estimates is measured using the mean square error and mean absolute value difference. These error measures are also derived using the maximum likelihood estimates. A comparison of these predictive errors with some of the previous works from the literature is given. The goodness-of-fit-tests for model validation are performed. In order to decide on model preference, evaluative criteria such as the mean magnitude of errors are also computed for the suggested models.

AMS Subject Classification: 62J05
Key Words: linear regression, non-homogenous Poisson process, power law process, software reliability

1. Introduction

Software has become an essential part of industry, medical systems, spacecraft and military systems, and in many other commercial systems. The application of software in many systems has led software reliability to be an important research area Musa et al [11], Rigdon et al [16], Sinpurwalla et al [20], and Xie [23]. Intensive studies were carried out by researchers and engineers to increase the chance that the software systems will perform satisfactory during operation.

Received: November 28, 2005

§Correspondence author

This requires the removal of faults during the testing phase. Researchers used existing technologies in order to improve the software reliability significantly by avoiding the occurrence of faults in the design and development of the software programs. Software reliability is a measure of the quality and performance of a software package. Thus, software reliability can be defined as the probability of failure free software in a specified environment for a specified period of time. A failure is the departure of software behavior from user requirements. This phenomenon has to be distinguished from the fault (bug) in the software code, which causes the occurrence of failure as soon as it is activated during program execution. If each time after a failure has been experienced, the underlying fault is detected and fixed correctly, then, the reliability of software will be improved with time. Many software reliability models (SRMs) are based on the inter-failure times. In this work, we are considering similar modeling principles, and our classification follows that of Horigome et al [6] by analyzing the time between successive failures. As failure occurrences initiate the removal of faults, failure times have been recorded and time between failures (TBF) has been used to find the mean time between failures (MTBF), which has been used for investigating the reliability growth. Models that discuss the behavior of MTBF are called SRMs. Since 1970, scientists and engineers have been coming up with different models to analyze the failure data in order to improve the software reliability. It is assumed that the successive software failure times are statistically independent.

It is now well recognized that reliability growth models are better described by a non-homogeneous Poisson process (NHPP). A power law process (PLP) is a special case of a non-homogeneous Poisson process. Horigome et al [6] proposed a power law process (PLP) model for describing the reliability growth, which relates the lifetimes from one stage to the next. By using a logarithmic transformation of the time between failures (TBF), they reduced the PLP model to an autoregressive process of order 1. An extension of this work, assuming stochastically monotone failure rate, is given by Littlewood et al [8] (model I). Mazzuchi et al [9] made extensions (model II) to Littlewood et al [8] and proposed new computational techniques. In Suresh [21], she used a different approach to model II of Mazzuchi et al [9] and observed an improvement in the following two error measure, the mean square error (MSE) and mean absolute value difference (MAVD). Qiao [13], and Qiao et al [14] proposed a reliability growth model based on PLP. Roberts [18], used the ideas of Qiao et al [14] to study software reliability models, and reported an improvement on these error measurements.

It has been shown in the literature, Mazzuchi et al [9], Qiao [13], Qiao et al [14], Roberts [18], and Suresh [21], that the PLP model is a good modeling assumption to represent TBF. The idea of this work stems from the fact that the logarithm of the PLP equation shows that it is a linear function of logarithm of failure time data. This led us to propose that the model could be taken as a simple linear regression. In this work, based PLP model, we will compute the predictive properties of mean time to failure (MTTF) using maximum likelihood estimation (MLE) method, regression method using logarithm of data with PLP assumption, simple linear regression applied directly to the data, and successive predicted time between failures method. We will show that the proposed models significantly improve the predictive accuracy under two popular measures of error, the mean square error and the mean absolute value difference. We will also simulate software failure times by taking different parameter values of the PLP, and the results were encouraging in terms of reducing MSE and MAVD. The methods introduced in this work not only reduce the prediction errors, they are also intuitively appealing and easy to implement.

In the next section, we will introduce some preliminary notations and definitions. In Section 3, we will present the new models; all are primarily based on linear regression. The goodness of fit tests will be carried out in Section 4. In addition, we will derive an appropriate intensity function when we base our analysis on linear regression. In Section 5, we will compare the predictive errors, MSE and MAVD, for the new models with those of previous models. Some other evaluative criteria to assess the performance of predictive models, such as the mean magnitude of relative error (MMRE), the mean magnitude of error relative to the estimate (MMER), and the median of the absolute residual (MdAR) are given in Section 6. We will conclude the paper in Section 7.

2. Preliminaries

In this section, we will first introduce some important terms and background results that are necessary. Suppose that a repairable system is observed until n failures $t_1, t_2, \ldots, t_n$ occur, where $0 < t_1 < t_2 < \ldots < t_n$. Let $T > 0$ be the random variable representing the time to next failure. The reliability function $R(t)$ is defined as the probability of no failure occurs up to time t and it is expressed by

$$R(t) = P(T > t) = \int_t^\infty f(x)\,dx \quad (t > 0),$$

where $f(t)$ is the probability density function (pdf) of the failure time $T > 0$. The cumulative distribution function (cdf) of the random variable T, can be written in terms of $R(t)$ as follows:

$$F(t) = \int_0^t f(x)\,dx = P(T \le t) = 1 - R(t)\,. \tag{1}$$

An NHPP is described by the failure intensity function, which is denoted by $v(t)$. The intensity function is the probability of failure in a small time interval divided by the length of the interval, i.e., the *intensity* (or *failure rate*) function of T is defined as:

$$v(t) = \lim_{\Delta t \to 0} \frac{P(t < T \le t + \Delta t \,|\, T > t)}{\Delta t}. \tag{2}$$

Since the failure intensity function $v(t)$ depends only on the cumulative failure time t, and not on the previous pattern of failure times, we can assume that a failed system is in exactly the same condition after a repair as it was just before the failure.

If the intensity function has the form, $v(t) = \lambda\beta t^{\beta-1}$, $t > 0$, $\beta > 0$, $\lambda > 0$, then the process is called the power law process, Rigdon et al [15], Rigdon et al [17], and Suresh [21], which is a special case of NHPP. Here, λ is the scale parameter and β is the shape parameter of PLP. This model is capable of modeling software systems that are improving and the systems that are deteriorating by the appropriate choice of the parameters. When $\beta > 1$, the failure intensity is increasing (TBF becomes shorter) at an exponential rate with time, and the PLP can model the reliability of a repairable system with rapid deterioration. When $\beta < 1$, the intensity function is strictly decreasing (TBF becomes larger), which corresponds to modeling the reliability of a repairable system with rapid improvement. For the PLP, when $\beta = 1$, mean time between failures is equal to a constant value.

The PLP has proved to be a useful in reliability modeling for several reasons.

1. It can be used to model deteriorating systems (TBF are getting shorter); as well as to model improving systems (TBF are getting larger).

2. Duane (1964), Rigdon et al [15] showed that many systems used at General Electric failure data fit a model that is closely related to PLP, and Statistical inference procedures can be used easily applied for PLP models.

We assume that the failure times of the test are random. Given the failure rate function, $v(t)$, the reliability function, $R(t)$, and the probability density function, $f(t)$, can be computed by:

$$R(t) = \exp\left[-\int_0^t v(x)\,dx\right],$$

and

$$f(t) = v(t)\exp\left[-\int_0^t v(x)dx\right] = v(t)\,R(t).$$

The mean time between failures of the repairable systems is the expected interval length from the current failure time $T_n = t_n$, to the next failure time $T_{n+1} = t_{n+1}$.

Let $f(t \mid t_1, t_2, \ldots, t_n)$ be the conditional distribution function of failure time T_{n+1} given $T_1 = t_1$, $T_2 = t_2, \ldots, T_n = t_n$. The joint pdf of the observed failure times can be derived using the following expression,

$$f(t_1, t_2, \ldots, t_n) = f_1(t_1)\, f_2(t_2 \mid t_1)\, f_3(t_3 \mid t_1, t_2) \cdots f_n(t_n \mid t_1, t_2, \ldots, t_{n-1}).$$

We will use the following results from Rigdon et al [16].

Theorem 1. *The joint pdf of the failure times $T_1, T_2, \ldots, T_n$ from an NHPP with intensity function $v(t)$ is*

$$f(t_1, t_2, \ldots, t_n) = \left(\prod_{i=1}^{n} v(t_i)\right)\exp\left(\int_0^{t_n} v(x)\,dx\right). \tag{3}$$

Also, from Rigdon et al [16], we have the following Markov property for the reliability function:

$$R_k(t_k \mid t_1, t_2, \ldots, t_{k-1}) = R(t_k \mid t_{k-1}) = \exp\left(-\int_{t_{k-1}}^{t_k} v(x)\,dx\right), \quad t_k > t_{k-1}. \tag{4}$$

As a consequence, we have

$$f(t_k \mid t_{k-1}) = v(t_k)\exp\left(-\int_{t_{k-1}}^{t_k} v(x)\,dx\right),\ t_k > t_{k-1}. \tag{5}$$

Mean squared error (MSE) is the most commonly used error measurement criteria of prediction. The MSE of an estimator T of an unobservable parameter θ is defined by

$$MSE(T) = E\left((T-\theta)^2\right).$$

For simplicity of computation, we will use the following formula

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(TBF_i - M\hat{T}BF_i\right)^2. \tag{6}$$

Mean Absolute value difference (MAVD) is the average of the difference between predicted mean time between failures and actual time between failure values, and computed by

$$MAVD = \frac{1}{n}\sum_{i=1}^{n}\left|TBF - M\hat{T}BF\right|. \quad (7)$$

2.1. The MLE Estimators of λ and β

Now we will describe the MLE approach for reliability estimation. Under the assumptions that the data follows PLP model, the joint pdf of the failure times is

$$\begin{aligned} f\left(t_1, t_2, \ldots, t_n\right) &= \left(\prod_{i=1}^{n}\lambda\beta\left(t_i\right)^{\beta-1}\right)\exp\left(-\int_0^{t_n}\lambda\beta x^{\beta-1}dx\right) \\ &= (\lambda\beta)^n\left(\prod_{i=1}^{n}t_i\right)^{\beta-1}\exp\left[-\lambda t_n^{\beta}\right], \quad 0 < t_1 < t_2 < \ldots < t_n < \infty. \quad (8) \end{aligned}$$

Then the log-likelihood function is

$$\ell(\lambda, \beta \,|\, t) = n\log\lambda + n\log\beta + (\beta - 1)\sum_{i=1}^{n}\log\left(t_i\right) - \lambda\left(t_n\right)^{\beta}. \quad (9)$$

Now, using the standard derivations Qiao et al [14], and Rigdon et al [15], the maximum likelihood estimators (MLEs) of the parameters, λ and β, are

$$\hat{\beta} = \frac{n}{\sum_{i=1}^{n-1}\ln\left(t_n/t_i\right)} = \frac{n}{\sum_{i=1}^{n-1}\left[\ln\left(t_n\right) - \ln\left(t_i\right)\right]}, \qquad \hat{\lambda} = \frac{n}{t_n^{\beta}}. \quad (10)$$

One of the most characteristic measures of reliability is the mean time-to-failure (MTTF) that represents length of the expected (average) interval of time to the next failure.

For the PLP,

$$MTBF_n = \int_{t_n}^{\infty} t\lambda\beta t^{\beta-1}\exp\left[-\lambda t^{\beta} + \lambda t_n^{\beta}\right]dt - t_n. \quad (11)$$

If $\beta = 1$, then

$$MTBF_n = \int_{t_n}^{\infty}\lambda t\exp\left(-\lambda t + \lambda t_n\right)dt - t_n = \frac{1}{\lambda}.$$

Thus, we can use $MTBF_n = \frac{1}{v_n(t)}$ Ascher et al [1], Cox et al [4], Roberts [18], and Suresh [21] as an approximation for the mean time between failures at nth stage.

Using the MLE, an estimator of the MTBF is given by

$$M\hat{T}BF = \frac{1}{\hat{v}(t)} = \left(\hat{\lambda}\hat{\beta}\right)^{-1} t^{1-\hat{\beta}}. \tag{12}$$

Later in Table 5 of Section 5, we will give the predictive errors, MSE and MAVD, resulting from using these estimators.

3. Regression Approach

In the present study, we shall apply regression methods for modeling the software failure times. First we assume the PLP model for the data. For the PLP models, in Subsection 3.1, instead of estimating the parameters λ and β through the standard method of MLE, we will estimate these parameters through regression (least squares) approach. We will see that this will result in substantially smaller values of MSE and MAVD compared to models discussed in Crow et al [5], Horigome et al [6], Mazzuchi et al [9], Roberts [18], and Suresh [21]. In addition, we will also compute these measures in Subsection 3.2 using simple linear regression model directly without assuming the PLP model, and derive what will be the resulting intensity function. In Subsection 3.3, we will calculate these error measures resulting from the successive prediction using the regression model. For our analysis, we will use following data sets from Musa [10]: Project 1, Project 5, and System 40.

3.1. Regression through the Power Law Process

For the power law process model, the MTBF can be approximated by the inverse of the intensity function Ascher et al [1], Cox et al [4], Roberts [18], and Suresh [21]. That is,

$$MTBF = \frac{1}{v(t)} = (\lambda\beta)^{-1} t^{(1-\beta)}, \tag{13}$$

where t is the failure time. Taking the natural logarithm, we get:

$$\ln(MTBF) = -\ln(\lambda\beta) + (1-\beta)\ln(t). \tag{14}$$

By writing, $Y = \ln(MTBF)$, $b = -\ln(\lambda\beta)$, $a = (1-\beta)$, and $Z = \ln(t)$, we can rewrite (14) as a linear equation

$$Y = b + aZ.$$

Data	System 40 ($\log(t)$, t is in seconds)	Project 1 ($\log(t)$, t is in seconds)	Project 5 ($\log(t)$, t is in seconds)
MSE	0.0653	0.1074	0.0483
MAVD	0.1660	0.2312	0.1588

Table 1: MSE and MAVD of different data with the *MTBFa-Regression Model*

Using the method of least squares for the linear regression model, Ryan [19], we can derive least squares estimators of a and b as $\hat{a}$ and $\hat{b}$, where

$$\hat{a} = \frac{\sum Z_i Y_i - \left(\sum Z_i\right)\left(\sum Y_i\right)/n}{\sum Z_i^2 - \left(\sum Z_i\right)^2/n}, \tag{15}$$

and

$$\hat{b} = \overline{Y} - \hat{a}\overline{Z}. \tag{16}$$

Now, using the following derivation

$$a = (1-\beta) \text{ implies } \beta = 1-a,$$
$$b = -\ln(\lambda\beta) \text{ implies } \lambda = \frac{1}{1-a}\,\frac{1}{e^b} = \frac{e^{-b}}{1-a},$$

we get the regression estimators of the PLP parameters λ and β, denoted by: $\hat{\beta}_{reg}$, and $\hat{\lambda}_{reg}$, as

$$\hat{\beta}_{reg} = 1 - \hat{a} \tag{17}$$

and

$$\hat{\lambda}_{reg} = \frac{1}{1-\hat{a}}\,\frac{1}{e^{\hat{b}}}. \tag{18}$$

Using these estimators, we can obtain an estimator of MTBF as:

$$M\hat{T}BFa_{reg} = \left(\hat{\lambda}_{reg}\hat{\beta}_{reg}\right)^{-1} t^{\left(1-\hat{\beta}_{reg}\right)}. \tag{19}$$

We will call this as the MTBFa-regression model.

We now present some numerical results using MTBFa-regression model by using real software failure data Musa [10]. Table 1 gives the MSE and MAVD calculated for three different software failure data of System 40, Project 1, and Project 5.

Comparison of these values with the corresponding values for the models discussed in Section 1 will be given in Section 5. One of the major advantages of this model lies in the simplicity to compute and comprehend, while substantially reducing MSE and MAVD.

Instead of assuming a PLP model, we suppose that for logarithm of data, $\log(t)$, follows a linear relationship, that is, $Y = \ln(MTTF) = a\ln t + b$. Then the question is: what is the corresponding intensity function? To answer this question, we will introduce the following. Let T be a continuous random variable (r.v) on $(0, \infty)$ denoting failure time. The mean residual time (MRT) is the average time to the next failure, given that no failure occurs up to time t, and is defined by

$$m(t) = E(T - t \,|\, T > t).$$

The following result from Bartoszynski [2] gives a relationship between MRT and the reliability function.

Theorem 2. *Let T be a random variable of continuous type, with density $f(x)$ and CDF $(F(x))$, assume that T is nonnegative, so that $f(t) = F(t) = 0$ for $t < 0$. Then*

$$E(T) = \int_0^\infty [1 - F(t)]\,dt = \int_0^\infty R(t)\,dt \tag{20}$$

and the MRT is

$$m(t) = \frac{\int\limits_t^\infty R(u)\,du}{R(t)}. \tag{21}$$

From the assumed linear relationship,

$$Y = \ln(MTTF) = a\ln t + b, \tag{22}$$

we get,

$$MTTF = e^{a\ln t + b} = e^{a\ln t}e^b = e^b t^a.$$

Now, equating the MRT with MTTF in order to find the intensity failure function, $v(t)$, $MTTF = m(t)$, we have

$$\begin{aligned} \frac{\int\limits_t^\infty R(u)\,du}{R(t)} &= e^b t^a, \\ e^b t^a R(t) &= \int_t^\infty R(u)\,du. \end{aligned} \tag{23}$$

Call $k = e^b$ for simplicity.

By differentiating (23) and using the following formula

$$\frac{d}{dt}\left\{\int_t^\infty R\,(u)\,du\right\} = -\frac{d}{dt}\left\{\int_\infty^t R\,(u)\,du\right\} = -R(t),$$

we obtain the following:

$$akt^{a-1}R(t) + kt^a \frac{d}{dt}\,R(t) = -R(t)\,,$$

or

$$\begin{aligned} \frac{d}{dt}\,R(t) + \left[\frac{1+akt^{a-1}}{kt^a}\right] R(t) = 0, \\ \frac{d}{dt}\,R(t) + \left[k^{-1}t^{-a} + at^{-1}\right] R(t) = 0. \end{aligned} \tag{24}$$

Since,

$$v(t) = \frac{f(t)}{R(t)},$$

where

$$f(t) = \frac{dF(t)}{dt} = \frac{d(1-R(t))}{dt} = -\frac{dR(t)}{dt}.$$

Now from (24),

$$-\frac{dR(t)}{dt} = \left[k^{-1}t^{-a} + at^{-1}\right] R(t).$$

Thus,

$$v(t) = \frac{f(t)}{R(t)} = \frac{-\frac{dR(t)}{dt}}{R(t)} = \frac{\left[k^{-1}t^{-a} + at^{-1}\right] R(t)}{R(t)} = \left[k^{-1}t^{-a} + at^{-1}\right], \quad t > 0.$$

Hence, if we assume the model, $Y = \ln(MTTF)$, then the resulting intensity function will be

$$v(t) = \frac{1+akt^{a-1}}{kt^a}, \quad t > 0. \tag{25}$$

3.2. Linear Regression Approach

Encouraged by the results of Section 3.1, the question is: what happens if we take directly a simple linear regression model, instead of assuming PLP model? That is consider the basic model as

$$TBF = b + at + \epsilon,$$

Data	System40 log(t), t is in seconds	Project 1 log(t), t is in seconds	Project 5 log(t), t is in seconds
MSE	2.56	1.80	2.71
MAVD	1.24	0.95	1.33

Table 2: MSE and MAVD values using the *SRRM1 model*

where TBF (time between failure) and t (time of failure) denote the dependent and independent variables respectively, and a and b are parameters that need to be estimated and ϵ represents the error term with mean zero. Now, the least squares estimates of the parameters a and b are given by

$$\hat{a} = \frac{\sum_{i=1}^{n} (t_i)\,(TBF_i) - \left(\sum_{i=1}^{n} t_i\right)\left(\sum_{i=1}^{n} (TBF_i)\right)/n}{\sum_{i=1}^{n} t_i^2 - \left(\sum_{i=1}^{n} t_i\right)^2 /n} \tag{26}$$

and

$$\hat{b} = \overline{TBF} - \hat{a}\bar{t}, \tag{27}$$

where $\overline{TBF}$ denotes the average. Thus, the following prediction equation represents the estimating mean time between failures

$$MT\hat{B}F = \hat{b} + \hat{a}t. \tag{28}$$

For this simple linear model (we will call this SRRM1 model), MSE and MAVD are given in Table 2.

Even though these values are higher than the values obtained in Table 1 for the MTBFa-regression model, the advantage here is that we do not assume the power law process model, instead just a simple linear regression is assumed. We will observe in Section 5 that these values are still comparable to the corresponding values obtained in the literature described in Section 1.

As in Section 3.1, one of the questions that can come up is: what should the intensity function corresponding to the simple linear regression model be? For the linear model, now we will derive the intensity function. We will once more use the mean residual time (MRT).

For a simple linear regression model,

$$MTTF = at + b \tag{29}$$

equating the MRT with MTTF in order to find the intensity failure function $v(t)$,

$$MTTF = m(t),$$

we have

$$\frac{\int_t^\infty R(u)\,du}{R(t)} = at + b, \tag{30}$$

following the same steps as in Section 3.1, we obtain the intensity function corresponding to the simple linear model as

$$v(t) = \frac{a+1}{at+b}, \quad t \geq 0, \quad t \neq -\frac{b}{a}. \tag{31}$$

It should be noted that in a simple linear regression model,

$$MTTF = at + b,$$

for $t = -(b/a)$ leads to $MTTF = 0$.

Remarks. (1) If the slope of the regression line is positive, then the MTBF is increasing. Let k be the desired optimal time of satisfactory operation (specified by the design). The testing of the software will be terminated when $MTBF = k$.

(2) If the slope of the regression line remains negative for a certain software failure data, then we have to discard the system.

3.3. Successive Prediction Using Regression

One of the main objectives of the model is to predict the software failure ahead of time. This could be done using the two models described in Section 3.1 and Section 3.2. Here we will describe another method where we did not use the power law process or power law process through regression, but our prediction task is achieved by doing repeated one-step ahead predictions by using the estimated parameters of the past failure data, which we call iterative (successive) prediction regression method. This method is a variation of the SRRM1 model in the way that we can predict time to failure (PTBF) directly.

The successive prediction method works as follows: it predicts only the next failure time one step ahead by applying the linear regression model with the iterative estimated parameters from past failure data. That is apply the linear regression line for computing the estimated parameters, then by using the parameters at every single failure time successively, we found the predicted time

Data	System 40 log(t), t is in seconds	Project 1 log(t), t is in seconds	Project 5 log(t), t is in seconds
MSE	4.35	2.30	3.01
MAVD	1.69	1.20	1.40

Table 3: MSE and MAVD values of the *successive recurrence regression model*

to next failure, then find the error, follow this pattern to find all the predicted values of failure time, then compute MSE and MAVD.

The predictive performance of this method is assumed by computing mean squared error (MSE) and mean absolute error (MAVD).

4. Model Validation

In this section, we will present some results related to validation of the assumptions made in Section 3. First we will give some *qqplots*, and then give some goodness of fit results.

4.1. Quantile-Quantile Plot of TBF and Predicted Values

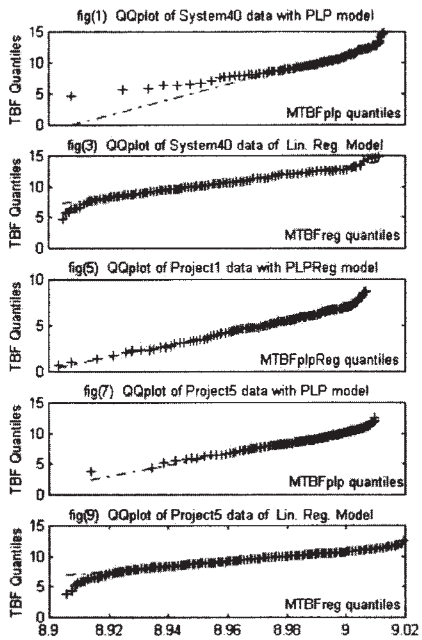
Our objective is to verify if the actual times between failure and the predicted values of TBF of the models specified in this paper are coming from the same assumed distribution. For this purpose, we applied the *qqplot(MTBF,TBF)*; this displays a quantile-quantile plot of the two samples, if these samples are coming from the same distribution, then the plot will be linear.

After applying the *qqplot* for the PLP and linear regression models by using system 40, Project 1, and Project 5 data sets, we notice that the actual TBF fits the linear regression model better than the PLP model. We can see that clearly by looking at the following graphs.

System 40 data: The *qqplot* in Figures 1 and 2 are very closely the same, while Figure 3 for the linear model fits the data better.

Project 1 data: Figures 4 and 6 for the PLP and linear regression models behave somehow the same, while in Figure 5 for the PLP_Reg models seem to fit the data better.

Project 5 data: Figures 7 and 8 are very close, while Figure 9 fits most of the data except for lowest quartiles.



Figures 1-9

4.2. Using Goodness-of-Fit-Test for Checking the PLP Process Model and the Regression Approach

Now we will test the adequacy of the power law process model by a goodness-of-fit test. Consider the hypothesis:

H_0: The power law process is the correct model.
H_1: The power law process is not the correct model.

The ratio power transformation is used for constructing the goodness-of-fit test. If $\Lambda(t)$ is the expected number of failures before time t, then

$$R_i = \frac{\Lambda(t_i)}{\Lambda(t_n)}, \quad i = 1, 2, \ldots, n$$

are distributed approximately as order statistics from a uniform distribution over the interval $[0, 1]$, Rigdon et al [16].

For the power law process of intensity function $v(t) = \lambda\beta t^{\beta-1}$

$$\Lambda(t) = \int_0^t v(t)\, dx = \int_0^t \lambda\beta x^{\beta-1} dx = \lambda t^{\beta}.$$

Then, the ratio power transformation is

$$R_i = \frac{\Lambda(t_i)}{\Lambda(t_n)} = \frac{\lambda t_i^{\beta}}{\lambda t_n^{\beta}} = \left(\frac{t_i}{t_n}\right)^{\beta}, \quad i = 1, 2, 3, \dots, n-1.$$

But in most cases, the parameter β is unknown and must be estimated by using the observed failure time data. We used the MLE estimates $\hat{\beta} = n / \left(\sum_{i=1}^{n-1} \log(t_n/t_i)\right)$, but it is biased. The unbiased estimator value β of β is

$$\beta = \frac{n-2}{n}, \quad \hat{\beta} = \frac{n-2}{\sum_{i=1}^{n-1} \log(t_n/t_i)}.$$

Then

$$\hat{R}_i = (t_i/t_n)^{\beta}, \tag{32}$$

The test statistic for the Cramer-von-Mises test denoted by C_R^2 can be applied for the goodness-of-fit test

$$C_R^2 = \frac{1}{12(n-1)} + \sum_{i=1}^{n-1} \left(\hat{R}_i - E\left(\hat{R}_i\right)\right)^2,$$

where $E\left(\hat{R}_i\right) = \frac{2i-1}{2(n-1)}$. Thus

$$C_R^2 = \frac{1}{12(n-1)} + \sum_{i=1}^{n-1} \left(\hat{R}_i - \frac{2i-1}{2(n-1)}\right)^2. \tag{33}$$

Large values of C_R^2 means that there is an evidence of a departure from the power law process Rigdon et al [16]. For certain values of confidence level, the critical values for the Cramer-von Mises goodness-of-fit test with $m = (n-1)$ can be obtained using Table A 6, p. 256 of Rigdon et al. [16]. For $m > 100$, we know the critical values will be smaller than the values given in this table. This information will be used in our decision making for the data sets we have.

Now we will test the adequacy of the linear regression model goodness-of-fit test using the Cramer-von-Mises test. Let

H_0: The linear regression is the correct model.
H_1: The linear regression is not the correct model.

Software Model	System 40 $n = 101$	Project1 $n = 133$	Project5 $n = 810$
PLP	0.04231272276	0.0121209429	0.008258820298
Simple Linear Regression	0.009400859828	0.007124003038	0.01730674729

Table 4: Goodness-of-fit test. Test statistics value

From equation (31), the intensity function for the simple linear regression model is given by

$$v(t) = \frac{a+1}{at+b}.$$

The expected number of failures before time t is:

$$\Lambda(t) = E\left(v(t)\right) = \int_0^t v\left(x\right)dx = \int_0^t \frac{a+1}{ax+b}\,dx = \frac{a+1}{a}\ln\left|\frac{at+b}{b}\right|.$$

Thus

$$\hat{R}_i = \frac{\Lambda\left(t_i\right)}{\Lambda\left(t_n\right)}, \quad i = \frac{\ln\left|\left(at_i+b\right)/b\right|}{\ln\left|\left(at_n+b\right)/b\right|}.$$

The parameters a and b are unknown, we can compute the estimators of a and b by using the least squares method of the observed failure data. The least squares estimators $\hat{a}$ and $\hat{b}$ are unbiased.

Now, the ratio power transformation is:

$$\hat{R}_i = \frac{\ln\left|\left(\hat{a}t_i+\hat{b}\right)/\hat{b}\right|}{\ln\left|\left(\hat{a}t_n+\hat{b}\right)/\hat{b}\right|}, \quad i = 1, 2, \ldots, n. \tag{34}$$

Thus, the test statistic for the Cramer-Von Mises test is:

$$C_R^2 = \frac{1}{12(n-1)} + \sum_{i=1}^{n-1}\left[\frac{\ln\left|\left(\hat{a}t_i+\hat{b}\right)/\hat{b}\right|}{\ln\left|\left(\hat{a}t_n+\hat{b}\right)/\hat{b}\right|} - \frac{2i-1}{2(n-1)}\right]^2. \tag{35}$$

We conclude that based on this test, both the PLP assumption and the derived linear intensity function assumptions are valid. Thus, not only do we have good error values, but the assumptions are valid too.

Model Type & Error	System 40 Measurement	Project 1 log(t), t is in seconds	Project 5 log(t), t is in seconds	log(t), t is in seconds
MTBFa_reg	MSE	0.0653	0.1074	0.0483
MTBFa_mle	MSE	23.4327	16.6756	292.5728
MTBFa_reg	MAVD	0.1660	0.2312	0.1588
MTBFa_mle	MAVD	4.7957	3.7663	16.3511

Table 5: Comparing MSE and MAVD of MLE and regression methods for models using logarithm of the failure data

5. Comparison of the MSE and MAVD Values

In this section, we will present comparison of our models with that of some relevant models from the literature. The error measures MSE and MAVD are used for these comparisons.

The following table gives the comparison between MLE approach and regression approach with assuming the logarithmic failure data of system 40, Project 1, and Project 5 follow PLP, where we use the notations, MTBFa_mle for the PLP model by using the MLE estimates. and MTBFa_reg the based PLP model through regression.

We can observe that the linear regression model, with the PLP assumption, outperforms the MLE method across all of the different data set. In the following table, we will see that this continues to be true in comparison with the models suggested in the literature.

We will now give results corresponding to simple linear regression (SRRM1) model, where we did not assume the PLP structure to the data. Given that there are virtually no distributional assumptions (except that is required in the least squares method of regression analysis), our results are competitive.

The following figures represent MSE and MAVD for all four models proposed in this paper to facilitate the comparison of these models.

6. Some Other Evaluation Criteria

In this section, we present another popular evaluation criteria used in Nyrtveit et al [12] to assess the performance of predictive repairable models. We used the

Model Name	Mean Square Error “MSE”	Mean Absolute Value Difference “MAVD”
Singpurwalla and Soyer Model I	4.92	Not Available (NA)
Singpurwalla and Soyer Model II	12.99	NA
Singpurwalla and Soyer Model III	9.58	NA
Singpurwalla and Soyer Model IV	16.2029	NA
MTBFhs (Horigome, Singpurwalla, & Soyer)	5.19	1.6865
MTBFa / SRGM / Suresh	4.72	1.85
MTBFa / Computed by Henry	4.15	NA
MTBFq / Henry	3.17	1.5677
SRRM1 /(Linear Regression)	2.57	1.2412

Table 6: Mean square errors comparison for different models. system 40 data (logarithm of the actual failure data, t is in seconds)

mean magnitude of relative error (MMRE) Nyrtveit et al [12], The magnitude of relative errors defined by

$$MRE = \frac{\left|TBF - M\hat{T}BF\right|}{TBF}, \tag{36}$$

where TBF is the actual time between failures, and $M\hat{T}BF$ is the predicted mean time between failures. Conte et al [3] considered $MMRE \leq 0.25$ to be an acceptable value for prediction models effort. There are advantages for this assessment:

1. Comparisons can be made easy across failure time data sets, Brian et al [3] and Walkerden et al [22].

2. The Mean magnitude of relative error is independent of units of data.

3. Comparisons can be made across all types of prediction models, Conte et al [3]. $MMRE$ is independent of scale, that is the expected value of MRE does not vary with size.

Kitchenham et al [7] proposed another measure, the magnitude of error relative to the estimate (MER). In Nyrtveit [12], they showed that claims 2 and 4 hold, and mentioned that MER measure seems preferable to MRE because it measures the error relative to the estimate value of mean time between failures.

Model Name	% Reduction Achieved by SRGM / Suresh & Rao Model	% Reduction Achieved by MTBFq / Henry, Tsokos & Rao Model	% Reduction Achieved by SRRM1 Model
Singpurwalla and Soyer Model I	4.06	35.57	47.76
Singpurwalla and Soyer Model II	63.66	75.60	80.22
Singpurwalla and Soyer Model III	50.73	66.91	73.17
Singpurwalla and Soyer Model IV	70.87	80.44	84.14
MTBFhs (Horigome, Singpurwalla & Soyer)	9.06	38.92	50.48
MTBFa / SRGM / Suresh & Rao Model	—	32.84	45.55
MTBFa / Computed by Henry	—	23.61	38.07
MTBFq /Henry & Tsokos	—	—	18.93

Table 7: Percentage reduction achieved by SRRM1. System 40 Data (logarithm of the actual failure data, t is in seconds)

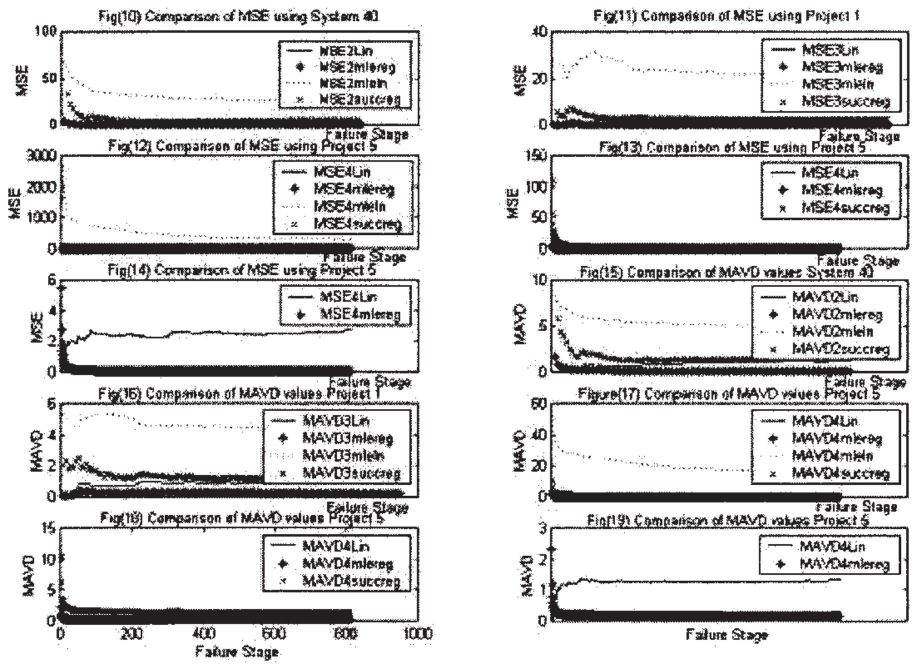
The MER is defined as

$$MER = \frac{\left|TBF - M\hat{T}BF\right|}{M\hat{T}BF}. \tag{37}$$

They used the mean of MER (MMER) as a measure of prediction error. Another measure proposed in Kitchenham [7] is median of the absolute residual (MdAR) instead of MMRE, where

$$AR = \left|TBF - M\hat{T}BF\right|. \tag{38}$$

Then, MdAR is the median of the values of AR. Also, they proposed MAR, which is the mean of AR. But MAR is nothing but MAVD that we have already calculated. Table 8 shows different methods of measurement (MSE, MAVD,



Figures 9-19

MMRE, MdAR) for the software reliability model for System 40, Project 1, and Project 5 software failure time data sets.

Based on the observations made in Nyrtveit [12], the standard MLE method under the PLP assumption does not perform well, and the successive recurrence model for Project 1 data set is also not a good model. In general, the models introduced in this paper, the linear regression, PLP_Reg, and successive recurrence models used for prediction perform better in comparison to MLE method for prediction. Overall, for the data sets considered, the PLP regression method outperforms all other models.

7. Conclusion

The power law process has proved to be a good model for repairable systems, from which the maximum likelihood estimators can be found through the likelihood function of the failure times. In this paper, we presented regression procedure for estimating the parameters of the power law process, which resulted in much-improved results in terms of the error measure MSE and MAVD than

System 40	PLP_mle	Simple Lin_reg	PLP_Reg	Successive Recurrence
MSE	23.4327	2.5657	0.0653	4.3483
MAVD	4.7957	1.2370	0.1660	1.6919
MMRE	2.1000	0.1320	0.0762	0.1737
MdAR	4.7961	1.0701	0.1375	1.3917
Project 1				
MSE	16.6756	1.7965	0.1074	2.3011
MAVD	3.7663	0.9479	0.2313	1.2164
MMRE	0.7424	0.2733	0.1704	0.3298
MdAR	4.1110	0.6596	0.1628	1.0105
Project 5				
MSE	292.6700	2.7122	0.0483	3.0068
MAVD	16.3511	1.3260	0.1588	1.4071
MMRE	1.8385	0.1679	0.0778	0.1714
MdAR	15.2034	1.1517	0.1343	1.2794

Table 8: Comparing predictive models by using different measurement methods

the MLE-based method. In addition, for the data sets considered here, we have shown that the simple linear regression model with out any NHPP assumptions gives a comparable results to that derived in the literature. Also, based on the mean magnitude of relative error values, PLP regression and simple linear regression models perform well. This suggests that, in software reliability modeling, first step should be to consider a simple linear model for software reliability prediction. The error measures resulting from any other model then could be compared to the error measure resulting from the simple linear regression model. A comparison of error measures with some of the models from the literature suggests that the linear regression models suggested here is competitive and in some cases outperforms. If the assumptions of the PLP model can be validated then the resulting linear regression model by far outperforms all the other models that we have compared. One of the natural questions is that, can we improve on error measures using higher order regression models? This, and some other relevant questions will be pursued in the future work.

References

[1] H. Ascher, H. Feingold, *Repairable Systems Reliability, Inference, Miscon-*

ceptions and their Causes, Marcel Dekker, New York (1984).

[2] R. Bartoszynski, M. Niewiadomska-Bugaj, *Probability and Statistical Inference*, Wiley, New York (1996).

[3] L.C. Brian, T. Langley, I. Wieczorek, A replicated assessment and comparison of common software cost modeling techniques, In: *Proc. Int'l. Conf. Software Eng.*, Volume **22** (2000), 377-386.

[4] D.R. Cox, P.A. Lewis, *The Statistical Analysis of Series of Events*, Chapman and Hall, London (1996).

[5] L.H. Crow, N.D. Singpurwalla, An empirically developed Fourier series model for describing failures, *IEEE Trans. Reliability*, **R-33**, No. 2 (1984), 176-183.

[6] M. Horigome, N.D. Singpurwalla, R. Soyer, *A Bayes Empirical Bayes Approach for (Software) Reliability Growth* (1985), 47-56.

[7] B.A. Kitchenham, S.G. MacDonell, L. Pickard, M.J. Shepperd, What accuracy statistics really measure, *IEEE Pro. Software Engineering*, **148** (2001), 81-85.

[8] B. Littlewood, J.L. Verral, A Bayesian reliability model with a stochastically monotone failure rate, *IEEE Trans. Reliab.*, **R-23**, No. 1 (1974), 108-114.

[9] T.A. Mazzuchi, R. Soyer, A Bayes empirical-Bayes model for software reliability, *IEEE Trans. Reliability*, **37**, No. 2 (1988), 248-254.

[10] J.D. Musa, http://www.dacs.dtic.mil/databases/sled/swrel.shtml.

[11] J.D. Musa, A. Iannino, K. Okumoto, *Software Reliability*, McGraw-Hill, New York (1987).

[12] I. Nyrtveit, E. Stensrud, Martin Shepperd, Reliability and validity in comparative studies of software prediction models, *IEEE Transactions on Software Engineering*, **31**, No. 5 (2005), 380-391.

[13] H. Qiao, *Parametric and Nonparametric Statistical Modeling: Reliability Analysis*, Ph.D. Thesis, University of South Florida (1993).

[14] H. Qiao, C. Tsokos, Best efficient estimates of the intensity function of the power law process, *Journal of Applied Statistics*, **25**, No. 1 (1998), 111-120.

[15] S.E. Rigdon, A.P. Basu, The power law process: a model for the reliability of repairable systems, *Journal of Quality Technology*, **21**, No. 4 (1989), 251-259.

[16] S.E. Rigdon, A.P. Basu, *Statistical Methods for the Reliability of Repairable Systems*, Wiley, New York (2000).

[17] S.E. Rigdon, M. Xiaolin, K.M. Boden, Statistical inference for repairable systems using the power law process, *Journal of Quality Technology*, **30**, No. 4 (1998).

[18] H. Roberts, *Predicting the Performance of Software Systems via the Power Law Process*, Ph.D. Thesis, University of South Florida, Tampa, FL (2000).

[19] P.T. Ryan, *Modern Regression Methods*, Wiley, New York (1996).

[20] N.D. Singpurwalla, S.P. Wilson, *Statistical Methods in Software Engineering*, Springer, New York (1999).

[21] N. Suresh, *Modeling and Analysis of Software Reliability*, Ph.D. Thesis, University of South Florida, Tampa, FL (1992).

[22] F. Walkerden, R. Jeffrey, An empirical study of analogy-based software effort estimation, *Empirical Software Eng.*, **4**, No. 2 (1999), 135-158.

[23] M. Xie, *Software Reliability Modeling*, World Scientific, Singapore (1991).