Analysis of Software Reliability Data using Exponential Power Model

Ashwini Kumar Srivastava
Department of Computer Application,
S.K.P.G. College, Basti, U.P., India
ashwini.skpg@gmail.com

Vijay Kumar
Departments of Mathematics & Statistics,
D.D.U. Gorakhpur University, Gorakhpur, U.P., India
vkgkp@rediffmail.com

***Abstract*—In this paper, Exponential Power (EP) model is proposed to analyze the software reliability data and the present work is an attempt to represent that the model is as software reliability model. The approximate MLE using Artificial Neural Network (ANN) method and the Markov chain Monte Carlo (MCMC) methods are used to estimate the parameters of the EP model. A procedure is developed to estimate the parameters of the EP model using MCMC simulation method in OpenBUGS by incorporating a module into OpenBUGS. The R functions are developed to study the various statistical properties of the proposed model and the output analysis of MCMC samples generated from OpenBUGS. A real software reliability data set is considered for illustration of the proposed methodology under informative set of priors.**

Keywords- EP model, Probability density, function, Cumulative density function, Hazard rate function, Reliability function, Parameter estimation, MLE, Bayesian estimation.

I. INTRODUCTION

Exponential models play a central role in analyses of lifetime or survival data, in part because of their convenient statistical theory, their important 'lack of memory' property and their constant hazard rates. In circumstances where the one-parameter family of exponential distributions is not sufficiently broad, a number of wider families such as the gamma, Weibull and lognormal models are in common use. Adding parameters to a well-established family of models is a time honoured device for obtaining more flexible new families of models. The Exponential Power model is introduced by [14] as a lifetime model. This model has been discussed by many authors [4], [9] and [12].

A model is said to be an Exponential Power model with shape parameter $\alpha>0$ and scale parameter $\lambda>0$, if the survival function of the model is given by

$$R(x) = \exp\left\{1 - e^{(\lambda x)^{\alpha}}\right\}, (\alpha, \lambda) > 0 \text{ and } x \in (0, \infty) \ .$$

A. *Model Analysis*

For $\alpha > 0$ and, $\lambda > 0$ the two-parameter Exponential Power model has the distribution function

$$F(x;\alpha,\lambda) = 1 - \exp\left\{1 - e^{(\lambda x)^{\alpha}}\right\};\ (\alpha,\lambda) > 0, x \geq 0 \qquad (1)$$

The probability density function (pdf) associated with eq (1) is given by

$$f(x;\alpha,\lambda) = \alpha \lambda^{\alpha} x^{\alpha-1} e^{(\lambda x)^{\alpha}} \exp\left\{1 - e^{(\lambda x)^{\alpha}}\right\};\ (\alpha,\lambda) > 0, x \geq 0 \qquad (2)$$

We shall write EP(α, λ) to denote Exponential Power model with parameters α and λ. The parameter α is named as 'shape parameter' by [4] and [14]. The R functions *dexp.power( )* and *pexp.power( )* given in `SoftreliaR` package can be used for the computation of pdf and cdf, respectively.

Some of the typical EP density functions for different values of α and for $\lambda = 1$ are depicted in Figure1. It is clear from the Figure 1 that the density function of the Exponential Power model can take different shapes.

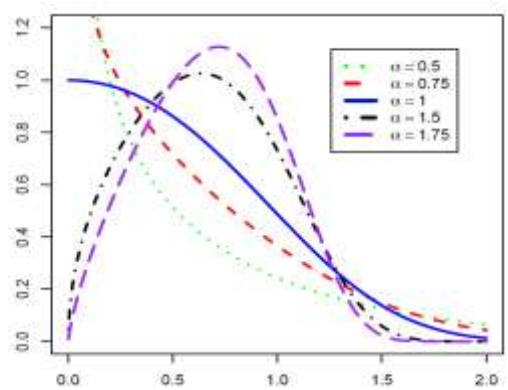

Figure 1 Plots of the probability density function of the Exponential Power model for λ=1 and different values of α

1) *Mode*

The mode can be obtained by solving the non-linear equation

$$(\alpha - 1) + \alpha (\lambda x)^{\alpha} \left\{1 - e^{(\lambda x)^{\alpha}}\right\} = 0 \ . \qquad (3)$$

2) *The quantile function*

For a continuous distribution F(x), the *p* percentile (also referred to as fractile or quantile), x_p, for a given *p*, $0 < p < 1$, is a number such that

$$P(X \leq x_p) = F(x_p) = p \ . \qquad (4)$$

The quantile for p=0.25 and p=0.75 are called first and third quartiles and the p=0.50 quantile is called the median(Q_2). The five parameters

Minimum(x), Q_1, Q_2, Q_3, Maximum(x)

are often referred to as the five-number summary or explanatory data analysis**.** Together, these parameters give a great deal of information about the model in terms of the centre, spread, and skewness. Graphically, the five numbers are often displayed as a boxplot. The quantile function of Exponential Power model can be obtained by solving

$$1-\exp\left\{1-e^{(\lambda x)^{\alpha}}\right\}=p$$

or, $$x_p=\frac{1}{\lambda}\log\{1-\log(1-p)\}^{1/\alpha} \quad ; \; 0<p<1. \qquad (5)$$

The computation of quantiles the R function *qexp.power( )*, given in `SoftreliaR` package, can be used. In particular, for p=0.5 we get

$$\text{Median}(x_{0.5})=\frac{1}{\lambda}\left(\log\{1-\log(0.5)\}\right)^{1/\alpha} \; . \qquad (6)$$

3) The random deviate generation

Let U be the uniform (0,1) random variable and F(.) a cdf for which $F^{-1}(.)$ exists. Then $F^{-1}(u)$ is a draw from distribution F(.) . Therefore, the random deviate can be generated from EP(α, λ) by

$$x=\frac{1}{\lambda}\log\{1-\log(1-u)\}^{1/\alpha} \; ; 0<u<1 \qquad (7)$$

where u has the U(0, 1) distribution. The R function *rexp.power( )*, given in `SoftreliaR` package, generates the random deviate from EP(α, λ).

4) Reliability function/survival function

The reliability/survival function

$$S(x;\alpha,\lambda)=\exp\left\{1-\exp(\lambda x)^{\alpha}\right\}, (\alpha,\lambda)>0 \text{ and } x\ge 0 \qquad (8)$$

The R function *sexp.power( )* given in `SoftreliaR` package computes the reliability/ survival function.

5) The Hazard Function

The hazard function of Exponential Power model is given by

$$h(x;\alpha,\lambda)=\alpha\lambda^{\alpha}x^{\alpha-1}\exp(\lambda x)^{\alpha}, (\alpha,\lambda)>0 \text{ and } x\ge 0 \qquad (9)$$

and the allied R function *hexp.power( )* given in `SoftreliaR` package. Since the shape of h(x) depends on the value of the shape parameter α. When $\alpha \ge 1$, the failure rate function is increasing. When $\alpha < 1$, the failure rate function is of bathtub shape. Thus the shape parameter α plays an important role for the model.

Since differentiating equation (9) w.r.to x, we have

$$h'(x)=\frac{1}{x}\left\{(\alpha-1)+\alpha(\lambda x)^{\alpha}\right\} \; . \qquad (10)$$

Setting $h'(x) = 0$ and after simplification, we obtain the change point as

$$x_0=\frac{1}{\lambda}\left(\frac{1-\alpha}{\alpha}\right)^{1/\alpha}. \qquad (11)$$

It easily follows that the sign of $h'(x)$ is determined by $(\alpha-1)+\alpha(\lambda.x)^{\alpha}$ which is negative for all $x \le x_0$ and positive for all $x \ge x_0$.

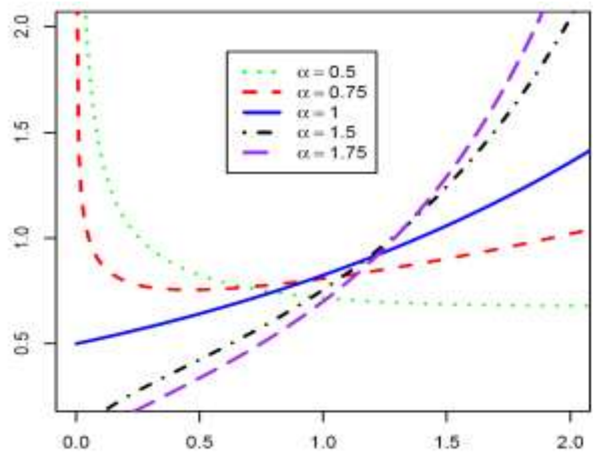

Figure 2 Plots of the hazard function of the Exponential Power model for λ=1 and different values of α

Some of the typical Exponential Power Model hazard functions for different values of α and for λ= 1 are depicted in Figure 2. It is clear from the Figure 2 that the hazard function of the Exponential Power model can take different shapes.

6) The cumulative hazard function

The cumulative hazard function H(x) defined as

$$H(x)=-\{1-\log F(x)\} \qquad (12)$$

can be obtained with the help of *pexp.power( )* function given in `SoftreliaR` package by choosing arguments *lower.tail=FALSE* and *log.p=TRUE.* i.e.

- *pexp.power(x, alpha, lambda, lower.tail = FALSE, log.p = TRUE)*

7) Failure rate average (fra) and Conditional survival function(crf)

Two other relevant functions useful in reliability analysis are failure rate average (fra) and conditional survival function (crf) The failure rate average of X is given by

$$FRA(x)=\frac{H(x)}{x} \; , x>0, \qquad (13)$$

where H(x) is the cumulative hazard function.

The survival function (s.f.) and the conditional survival of X are defined by

$$R(x) = 1 - F(x)$$

and $$R(x \mid t) = \frac{R(x+t)}{R(x)}, \quad t > 0,\ x > 0,\ R(\cdot) > 0, \qquad (14)$$

respectively, where $F(\cdot)$ is the cdf of X. Similarly to h(x) and FRA(x), the distribution of X belongs to the new better than used (NBU), exponential, or new worse than used (NWU) classes, when $R(x \mid t) < R(x)$, $R(t \mid x) = R(x)$, or $R(x \mid t) > R(x)$, respectively.

The R functions *hra.exp.power( )* and *crf.exp.power( )* given in `SoftreliaR` package can be used for the failure rate average (fra) and conditional survival function(crf), respectively.

II. Maximum Likelihood Estimation and Information Matrix

Let $x=(x_1, \ldots, x_n)$ be a sample from a distribution with cumulative distribution function (1). The log likelihood function of the parameter $L(\alpha, \lambda)$ is given by

$$\log L(\alpha,\lambda) = n\log\alpha + n\alpha\log\lambda + (\alpha-1)\sum_{i=1}^{n}\log x_i + \lambda^{\alpha}\sum_{i=1}^{n} x_i^{\alpha} + n - \sum_{i=1}^{n}\exp\left\{(\lambda x_i)^{\alpha}\right\} \qquad (15)$$

Therefore, to obtain the MLE's of α and λ we can maximize eq.(15) directly with respect to α and λ or we can solve the following two non-linear equations using iterative procedure [2] and [4]:

$$\frac{\partial \log L}{\partial\alpha} = \frac{n}{\alpha} + n\log\lambda + \sum_{i=1}^{n}\log x_i + \sum_{i=1}^{n}(\lambda x_i)^{\alpha}\log(\lambda x_i)\left\{1-\exp(\lambda x_i)^{\alpha}\right\} = 0 \qquad (16)$$

$$\frac{\partial \log L}{\partial\lambda} = \frac{n\alpha}{\lambda} + \sum_{i=1}^{n}\alpha\lambda^{\alpha-1}x_i^{\alpha}\left\{1-\exp(\lambda x_i)^{\alpha}\right\} = 0 \qquad (17)$$

Let us denote $\hat{\theta} = (\hat{\alpha}, \hat{\lambda})$ as the MLEs of $\underline{\theta} = (\alpha, \lambda)$. It is not possible to obtain the exact variances of $\hat{\theta} = (\hat{\alpha}, \hat{\lambda})$. The asymptotic variances of $\hat{\theta} = (\hat{\alpha}, \hat{\lambda})$ can be obtained from the following asymptotic property of $\hat{\theta} = (\hat{\alpha}, \hat{\lambda})$

$$\left(\underline{\hat{\theta}} - \underline{\theta}\right) \to N_2\left(0, (I(\underline{\theta}))^{-1}\right) \qquad (18)$$

where $I(\underline{\theta})$ is the Fisher's information matrix given by

$$I(\theta) = -\begin{bmatrix} E\left(\frac{\partial^2 \ln L}{\partial\alpha^2}\right) & E\left(\frac{\partial^2 \ln L}{\partial\alpha\,\partial\lambda}\right) \\ E\left(\frac{\partial^2 \ln L}{\partial\alpha\,\partial\lambda}\right) & E\left(\frac{\partial^2 \ln L}{\partial\lambda^2}\right) \end{bmatrix} \qquad (19)$$

In practice, it is useless that the MLE has asymptotic variance $(I(\underline{\theta}))^{-1}$ because we do not know $\underline{\theta}$. Hence, we approximate the asymptotic variance by "plugging in" the estimated value of the parameters. The common procedure is to use observed Fisher information matrix $O(\hat{\theta})$ (as an estimate of the information matrix $I(\theta)$) given by

$$O(\hat{\theta}) = -\begin{pmatrix} \frac{\partial^2 \ln L}{\partial\alpha^2} & \frac{\partial^2 \ln L}{\partial\alpha\,\partial\lambda} \\ \frac{\partial^2 \ln L}{\partial\alpha\,\partial\lambda} & \frac{\partial^2 \ln L}{\partial\lambda^2} \end{pmatrix}\Bigg|_{(\hat{\alpha},\hat{\lambda})} = -H(\theta)\Big|_{\theta=\hat{\theta}} \qquad (20)$$

where H is the Hessian matrix, $\theta=(\alpha, \lambda)$ and $\hat{\theta} = (\hat{\alpha}, \hat{\lambda})$. The observed Fisher information is evaluated at MLE rather than determining the expectation of the Hessian at the observed data. This is simply the negative of the Hessian of the log-likelihood at MLE. If the Newton-Raphson algorithm is used to maximize the likelihood then the observed information matrix can easily be calculated. Therefore, the variance-covariance matrix is given by

$$\left(-H(\theta)\Big|_{\theta=\hat{\theta}}\right)^{-1} = \begin{pmatrix} Var(\hat{\alpha}) & cov(\hat{\alpha},\hat{\lambda}) \\ cov(\hat{\alpha},\hat{\lambda}) & Var(\hat{\lambda}) \end{pmatrix}. \qquad (21)$$

Hence, from the asymptotic normality of MLEs, approximate 100(1-γ)% confidence intervals for α and λ can be constructed as

$$\hat{\alpha} \pm z_{\gamma/2}\sqrt{Var(\hat{\alpha})} \quad \text{and} \quad \hat{\lambda} \pm z_{\gamma/2}\sqrt{Var(\hat{\lambda})} \qquad (22)$$

where $z_{\gamma/2}$ is the upper percentile of standard normal variate.

III. Bayesian Estimation in OpenBUGS

The most widely used piece of software for applied Bayesian inference is the OpenBUGS. It is a fully extensible modular framework for constructing and analyzing Bayesian full probability models. This open source software requires incorporation of a module (code) to estimate parameters of Exponential Power model.

A module ***dexp.power_T(alpha, lambda)*** is written in component Pascal, enables to perform full Bayesian analysis of Exponential Power model into OpenBUGS using the method described in [15] and [16].

A. Implementation of Module - dexp.power_T(alpha, lambda)

The developed module is implemented to obtain the Bayes estimates of the Exponential Power model using MCMC method. The main function of the module is to generate

MCMC sample from posterior distribution under informative set of priors, i.e. Gamma priors.

1) Data Analysis

We are using software reliability data set *SYS2.DAT* - 86 time-between-failures [10] is considered for illustration of the proposed methodology. In this real data set, Time-between-failures is converted to time to failures and scaled.

B. Computation of MLE and Approximate ML estimates using ANN

The Exponential Power model is used to fit this data set. We have started the iterative procedure by maximizing the log-likelihood function given in eq.(15) directly with an initial guess for $\alpha = 0.5$ and $\lambda = 0.06$, far away from the solution. We have used *optim*( ) function in R with option Newton-Raphson method. The iterative process stopped only after 7 iterations. We obtain $\hat{\alpha} = 0.905868898$, $\hat{\lambda} = 0.001531423$ and the corresponding log-likelihood value = -592.7172. The similar results are obtained using *maxLik* package available in R. An estimate of variance-covariance matrix, using eq.(22), is given by

$$\begin{pmatrix} \mathrm{Var}(\hat{\alpha}) & \mathrm{cov}(\hat{\alpha},\hat{\lambda}) \\ \mathrm{cov}(\hat{\alpha},\hat{\lambda}) & \mathrm{Var}(\hat{\lambda}) \end{pmatrix} = \begin{bmatrix} 7.265244\text{e-}03 & -1.474579\text{e-}06 \\ -1.474579\text{e-}06 & 1.266970\text{e-}08 \end{bmatrix}$$

Thus using eq.(23), we can construct the approximate 95% confidence intervals for the parameters of EP model based on MLE's. Table I shows the MLE's with their standard errors and approximate 95% confidence intervals for α and λ.

TABLE I MAXIMUM LIKELIHOOD ESTIMATE, STANDARD ERROR AND 95% CONFIDENCE INTERVAL

Parameter	**MLE**	**Std. Error**	**95% Confidence Interval**
alpha	0.905868	0.085236	(0.7388055, 1.0729322)
lambda	0.001531	0.000112	(0.0013108, 0.0017520)

An approximate ML estimates based on Artificial Neural Networks are obtained by using the `neuralnet` package available in R. We have chosen one hidden- layer feedforward neural networks with sigmoid activation function [1]. The results are quite close to exact ML estimates.

C. Model Validation

To study the goodness of fit of the Exponential Power model, we compute the Kolmogorov-Smirnov statistic between the empirical distribution function and the fitted distribution function when the parameters are obtained by method of maximum likelihood. For this we can use R function ks.*exp.power*( ), given in `SoftreliaR` package. The result of K-S test is D = 0.0514 with the corresponding p-value = 0.9683, Therefore, the high p-value clearly indicates that Exponential Power model can be used to analyze this data set, and we also plot the empirical distribution function and the fitted distribution function in Figure 3. From above result and Figure 3, it is clear that the estimated Exponential Power model provides excellent fit to the given data.

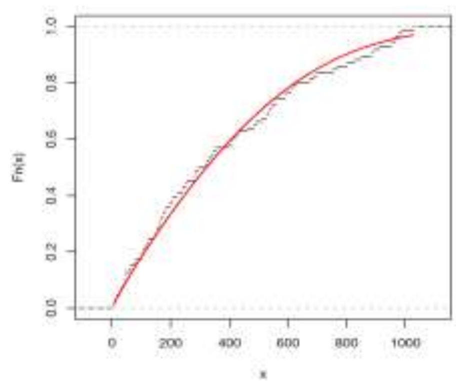

Figure 3 The graph of empirical distribution and fitted distribution function.

The other graphical method widely used for checking whether a fitted model is in agreement with the data is Quantile-Quantile (Q-Q) plots.

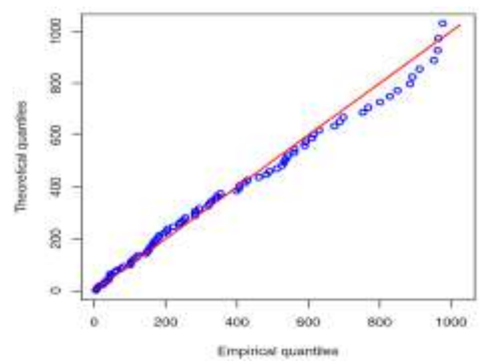

Figure 4 Quantile-Quantile(Q-Q) plot using MLEs as estimate.

The Q-Q plots show the estimated versus the observed quantiles. If the model fits the data well, the pattern of points on the Q-Q plot will exhibit a 45-degree straight line. Note that all the points of a Q-Q plot are inside the square

$$\left[\hat{F}^{-1}(p_{1:n}),\ \hat{F}^{-1}(p_{n:n})\right] \times \left[x_{1:n}, x_{n:n}\right].$$

The corresponding R function *qq.exp.power*( ) is given in `SoftreliaR` package. As can be seen from the straight line pattern in Figure 4, the Exponential Power model fits the data very well.

IV. BAYESIAN ANALYSIS UNDER INFORMATIVE PRIORS, I.E., GAMMA PRIORS

***OpenBUGS code to run MCMC*:**

```
Model
  {
    for( i in 1 : N )
   {
      x[i] ~ dexp.power_T(alpha, lambda)
     }
 # Prior distributions of the Model parameters
```

```
        # Gamma prior for alpha
            alpha ~ dgamma(0.001, 0.001)
        # Gamma prior for lambda
            lambda ~ dgamma(0.001, 0.001)
    }
```

Data

```
list(N=86, x=c(4.79, 7.45, 10.22, 15.76, 26.10, 28.59, 35.52, 41.49,
42.66, 44.36, 45.53, 58.27, 62.96, 74.70, 81.63, 100.71, 102.06, 104.83,
110.79, 118.36, 122.73, 145.03, 149.40, 152.80, 156.85, 162.20, 164.97,
168.60, 173.82, 179.95, 182.72, 195.72, 203.93, 206.06, 222.26, 238.27,
241.25, 249.99, 256.17, 282.57, 282.62, 284.11, 294.45, 318.86, 323.46,
329.11, 340.30, 344.67, 353.94, 398.56, 405.70, 407.51, 422.36, 429.93,
461.47, 482.62, 491.46, 511.83, 526.64, 532.23, 537.13, 543.06, 560.75,
561.60, 589.96, 592.09, 610.75, 615.65, 630.52, 673.74, 687.92, 698.15,
753.05, 768.25, 801.06, 828.22, 849.97, 885.02, 892.27, 911.90, 951.69,
962.59, 965.04, 976.98, 986.92, 1025.94))
```

Initial values

```
# chain 1
list(alpha=0.2 , lambda=0.01)
# chain 2
list(alpha= 1.0, lambda=0.10)
```

We run the model to generate two Markov Chains at the length of 40,000 with different starting points of the parameters. The convergence is monitored using trace and ergodic mean plots, we find that the Markov Chain converge together after approximately 2000 observations. Therefore, burnin of 5000 samples is more than enough to erase the effect of starting point(initial values). Finally, samples of size 7000 are formed from the posterior by picking up equally spaced every fifth outcome, i.e. thin=5, starting from 5001.This is done to minimize the auto correlation among the generated deviates. Therefore, we have the posterior sample $\{\alpha_{1i}, \lambda_{1i}\}$, i = 1,…,7000 from chain 1 and $\alpha_{2i}, \lambda_{2i}\}$, i = 1,…,7000 from chain 2.

The chain 1 is considered for convergence diagnostics plots. The visual summary is based on posterior sample obtained from chain 2 whereas the numerical summary is presented for both the chains.

A. Convergence diagnostics

Sequential realization of the parameters α and λ can be observed in figure 5. The Markov chain is most likely to be sampling from the stationary distribution and is mixing well.

1) History(Trace) plot

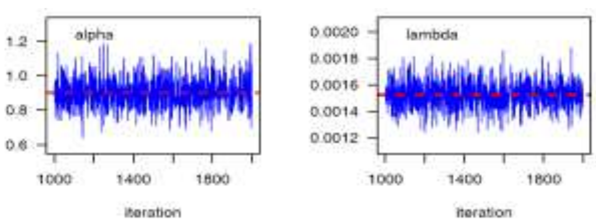

Figure 5 Sequential realization of the parameters α and λ.

There is ample evidence of convergence of chain as the plots show no long upward or downward trends, but look like a horizontal band, then we has evidence that the chain has converged.

2) Running Mean (Ergodic mean) Plot

The convergence pattern based on Ergodic average as shown in figure 6 is obtained after generating a time series (Iteration number) plot of the running mean for each parameter in the chain. The running mean is computed as the mean of all sampled values up to and including that at a given iteration.

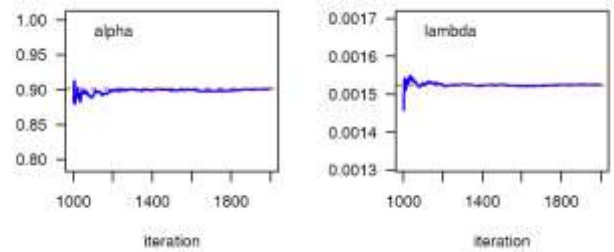

Figure 6 The Ergodic mean plots for α and λ.

3) Autocorrelation

The graph shows that the correlation is almost negligible. We may conclude that the samples are independent.

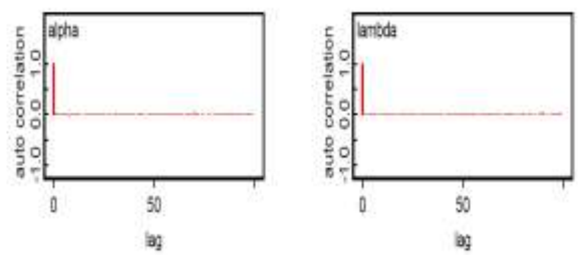

Figure7 The autocorrelation plots for α and λ.

4) Brooks-Gelman-Rubin Plot

Uses parallel chains with dispersed initial values to test whether they all converge to the same target distribution. Failure could indicate the presence of a multi-mode posterior distribution (different chains converge to different local modes) or the need to run a longer chain (burn-in is yet to be completed).

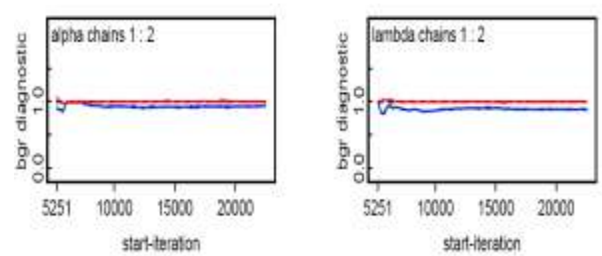

Figure 8 The BGR plots for α and λ.

From the Figure 8, it is clear that convergence is achieved. Thus we can obtain the posterior summary statistics.

B. Numerical Summary

In Table II, we have considered various quantities of interest and their numerical values based on MCMC sample of posterior characteristics for Exponential Power model under Gamma priors.

TABLE II NUMERICAL SUMMARIES UNDER GAMMA PRIORS

Characteristics	Chain 1		Chain 2	
	alpha	lambda	alpha	lambda
Mean	0.895628	0.001525208	0.8971537	0.001525661
Standard Deviation	0.08527441	0.0001170499	0.0841966	0.0001159336
Monte Carlo(MC) error	0.001108	1.322E-6	9.216E-4	1.402E-6
Minimum	0.607600	0.001102000	0.5909000	0.001089
2.5th Percentile($P_{2.5}$)	0.733700	0.001307975	0.7332975	0.001305
First Quartile (Q_1)	0.837400	0.001447000	0.8394750	0.001447
Median	0.893800	0.001522000	0.8952000	0.001523
Third Quartile (Q_3)	0.949925	0.001600000	0.9522000	0.001603
97.5th Percentile($P_{97.5}$)	1.068000	0.001762000	1.0690000	0.001761
Maximum	1.234000	0.002138000	1.2910000	0.001989
Mode	0.8911972	0.001506012	0.8915055	0.001502559
95% Credible Interval	(0.7337, 1.0680)	(0.0013, 0.00176)	(0.7332975, 1.069	(0.0013, 0.00176)
95% HPD Credible Interval	(0.7324, 1.0650)	(0.0013, 0.00176)	(0.7313, 1.0650)	(0.0012, 0.00174)

C. Visual summary

1) Box plots

The boxes represent inter-quartile ranges and the solid black line at the (approximate) centre of each box is the mean; the arms of each box extend to cover the central 95 per cent of the distribution - their ends correspond, therefore, to the 2.5% and 97.5% quantiles. (Note that this representation differs somewhat from the traditional.)

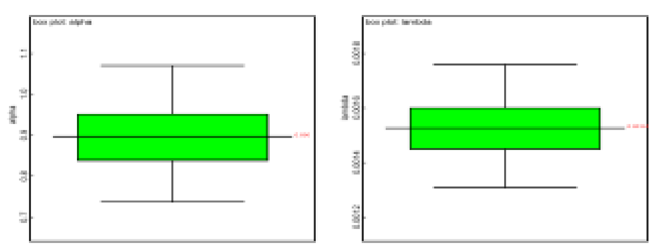

Figure 9 The boxplots for alpha and lambda.

2) Kernel density estimates

Histograms can provide insights on symmetric, behaviour in the tails, presence of multi-modal behaviour, and data outliers; histograms can be compared to the fundamental shapes associated with standard analytic distributions.

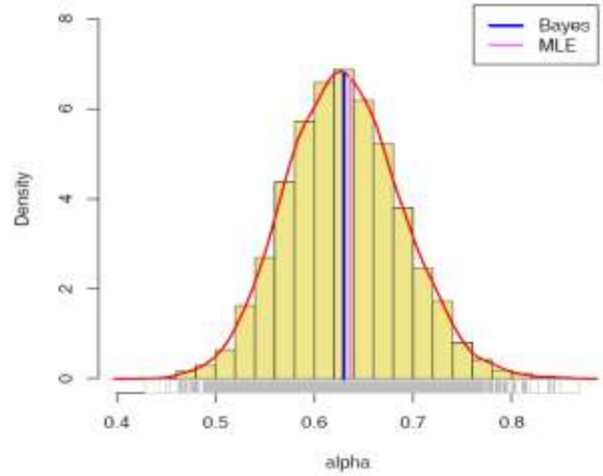

Figure 10 Kernel density estimate and histogram of α based on MCMC samples, vertical lines indicates the corresponding ML and Bayes estimates.

Figure 10 and 11 provide the kernel density estimate of α and λ respectively. It can be seen that α and λ both are symmetric.

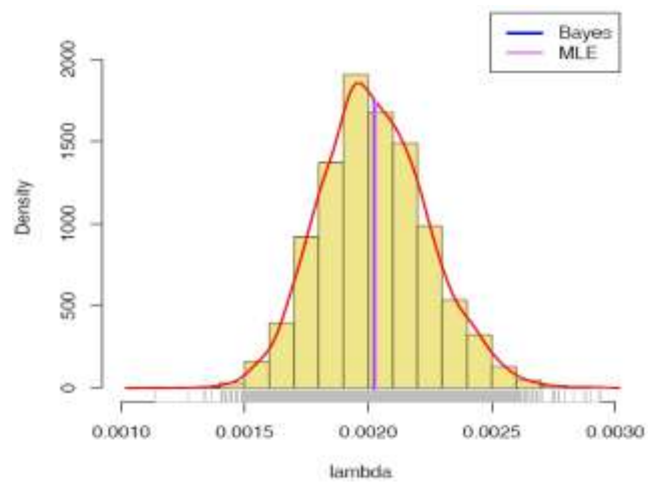

Figure 11 Histogram and kernel density estimate of λ based on MCMC samples

D. Comparison with MLE

For the comparison with MLE we have plotted two graphs. In Figure 12, the density functions $f(x; \hat{\alpha}, \hat{\lambda})$ using MLEs and Bayesian estimates, computed via MCMC samples under gamma priors, are plotted.

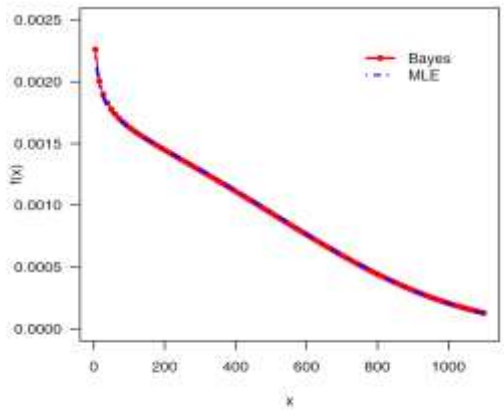

Figure 12 The density functions $f(x; \hat{\alpha}, \hat{\lambda})$ using MLEs and Bayesian estimates, computed via MCMC samples under gamma priors.

The Figure 13, exhibits the estimated reliability function(dashed line) using Bayes estimate under gamma priors and the empirical reliability function(solid line).

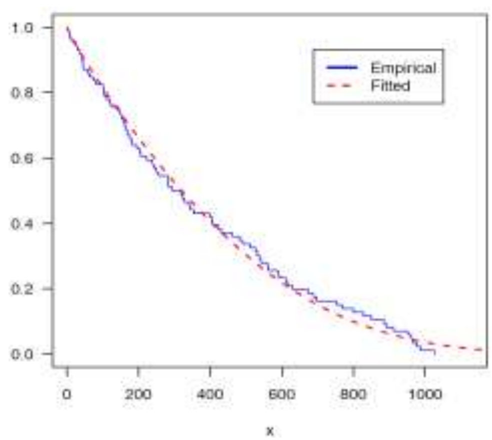

Figure 13 The estimated reliability function(dashed line) and the empirical reliability function (solid line).

It is clear from the Figures, the MLEs and the Bayes estimates with respect to the gamma priors are quite close and fit the data very well.

V. CONCLUSION

In this research paper, we have presented the Exponential Power model as software reliability model which was motivated by the fact that the existing models were inadequate to describe the failure process underlying some of the data sets.

We have developed the tools for empirical modelling, e.g., model analysis, model validation and estimation. The exact as well as approximate ML estimates using ANN of the parameters alpha (α) and lambda (λ) have been obtained.

An attempt has been made to estimate the parameters in Bayesian setup using MCMC simulation method under gamma priors. The proposed methodology is illustrated on a real data set. We have presented the numerical summary and visual summary under different priors which includes Box plots, Kernel density estimates based on MCMC samples. The Bayes estimates are compared with MLE. We have shown that the Exponential Power model is suitable for modeling the software reliability data and the tools developed for analysis can also be used for any other type of data sets.

ACKNOWLEDGEMENT

Our thanks to Dr. Andrew Thomas, St. Andrew's University, UK, Prof. Uwe Ligges, TU Dortmund University, Germany and Prof R.S. Srivastava, DDU Gorakhpur University, Gorakhpur, for their valuable suggestions to make this work a success.

REFERENCES

[1] Cervellera, C., Macciò,D. And Muselli,M.,(2008) Deterministic Learning For Maximum-Likelihood Estimation Through Neural Networks, IEEE Transactions on neural networks, vol. 19, no. 8.

[2] Chen, M. H. and Shao, Q. M. (1999). Monte Carlo estimation of Bayesian credible intervals and HPD intervals, Journal of Computational and Graphical Statistics. 8(1).

[3] Chen, M., Shao, Q. and Ibrahim, J.G. (2000). *Monte Carlo Methods in Bayesian Computation*, Springer, NewYork.

[4] Chen, Z.(1999). Statistical inference about the shape parameter of the exponential power distribution, Statistical Papers 40, 459-468 (1999)

[5] Hornik, K., (2004). *The R FAQ* (on-line). Available at http://www.ci.tuwien. ac.at/~hornik/R/.

[6] Ihaka, R.; Gentleman, R.R. (1996). R: A language for data analysis and graphics, Journal of Computational and Graphical Statistics, 5, 299–314.

[7] Jalote, P. (1991). An Integrated Approach to Software Engineering. Springer – Verlag, New York.

[8] Lawless, J. F., (2003). *Statistical Models and Methods for Lifetime Data*, 2nd ed., John Wiley and Sons, New York.

[9] Leemis, L.M., (1986). *Lifetime distribution identities*, IEEE Transactions on Reliability, 35, 170-174.

[10] Lyu M.R., (1996). *Handbook of Software Reliability Engineering*, IEEE Computer Society Press, McGraw Hill, 1996.

[11] R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

[12] Rajarshi, M.B. & Rajarshi, S. (1988). *Bathtub distributions: A review*, Communications in Statistics A: Theory and Methods, 17, 2597-2621.

[13] Singpurwalla, N.D. and S. Wilson (1994). Software Reliability Modeling. International Statist. Rev., 62 3:289-317.

[14] Smith, R.M. & Bain, L.J. (1975). *An exponential power life-test distribution*, Communications in Statistics, 4, 469-481.

[15] Thomas, A. (2004). OpenBUGS, URL http://mathstat.helsinki.fi/ openbugs/.

[16] Thomas, A. (2007). *OpenBUGS Developer Manual*, Version 3.0.2. URLhttp://mathstat.helsinki.fi/openbugs/

AUTHORS PROFILE

ASHWINI KUMAR SRIVASTAVA is a research Scholar and submitted thesis for the award of Ph.D. in Computer Science. He received his M.Sc in Mathematics from D.D.U.Gorakhpur University, MCA(Hons.) from U.P.Technical university and M. Phil in Computer Science from Allagappa University. Currently working as Assistant Professor in Department of Computer Application in Shivharsh Kisan P.G. College, Basti, U.P. He has got 6 years of teaching experience as well as 3 years research experience. His main research interests are Software Reliability, Artificial Neural Networks, Bayesian methodology and Data Warehousing.

VIJAY KUMAR received his M.Sc and Ph.D. in Statistics from D.D.U.Gorakhpur University. Currently working as Reader(Associate Professor) in Department of Mathematics and Statistics in DDU Gorakhpur University, Gorakhpur U.P. He has got 16 years of teaching/research experience. He is visiting Faculty of Max-Planck-Institute, Germany. His main research interests are Reliability Engineering, Bayesian Statistics, and Actuarial Science.