Open Journal of Statistics, 2014, 4, 742-750
Published Online October 2014 in SciRes. http://www.scirp.org/journal/ojs
http://dx.doi.org/10.4236/ojs.2014.49069

Two-Sample Bayesian Predictive Analyses for an Exponential Non-Homogeneous Poisson Process in Software Reliability

Albert Orwa Akuno, Luke Akong'o Orawo, Ali Salim Islam

Department of Mathematics, Egerton University, Egerton, Kenya
Email: orwaakuno@gmail.com, orawo2000@yahoo.com, asislam54@yahoo.com

Received 4 August 2014; revised 9 September 2014; accepted 29 September 2014

Abstract

The Goel-Okumoto software reliability model is one of the earliest attempts to use a non-homogeneous Poisson process to model failure times observed during software test interval. The model is known as exponential NHPP model as it describes exponential software failure curve. Parameter estimation, model fit and predictive analyses based on one sample have been conducted on the Goel-Okumoto software reliability model. However, predictive analyses based on two samples have not been conducted on the model. In two-sample prediction, the parameters and characteristics of the first sample are used to analyze and to make predictions for the second sample. This helps in saving time and resources during the software development process. This paper presents some results about predictive analyses for the Goel-Okumoto software reliability model based on two samples. We have addressed three issues in two-sample prediction associated closely with software development testing process. Bayesian methods based on non-informative priors have been adopted to develop solutions to these issues. The developed methodologies have been illustrated by two sets of software failure data simulated from the Goel-Okumoto software reliability model.

Keywords

Nonhomogeneous Poisson Process, Software Reliability Models, Non-Informative Priors, Bayesian Approach

1. Introduction

Software reliability is defined as the probability of failure free software operations for a specified period of time in a specified environment [1]. The reliability of any software is of great interest to the software developers be-

How to cite this paper: Akuno, A.O., Orawo, L.A. and Islam, A.S. (2014) Two-Sample Bayesian Predictive Analyses for an Exponential Non-Homogeneous Poisson Process in Software Reliability. *Open Journal of Statistics*, **4**, 742-750.
http://dx.doi.org/10.4236/ojs.2014.49069

fore a decision is made to release the software into the market. Software developers need correct and concise information about how reliable software is before they decide to release the software into the market as single software defect can cause system failure and to avoid these failures, reliable software is required [2]. Software reliability is achieved through testing during the software development stage [3]. The usual way of removing bugs from a software system is by running test cases on the software system similar to the way users will operate it in their particular environment. However, the emulation of end-user environment during the test interval is difficult, expensive and time consuming especially when there are multiple types of end-users in different environments. Software reliability modeling can be used to address this dilemma especially when reliability testing on two software systems can be achieved in one testing period. Software reliability modeling can provide the basis for planning reliability growth tests, monitoring progress, estimating current reliability, forecasting and predicting future reliability improvements [4]. Predictive analyses help in conducting forecasting and prediction. A prediction interval is usually constructed to provide the time frame when the kth $(k>0)$ future failure observation will occur with a pre-determined confidence level [5].

An Exponential Nonhomogeneous Poisson Process with intensity function

$$\lambda(t)=\alpha\beta e^{-\beta t} \tag{1}$$

is the earliest software reliability model to be developed. Such a model is a NHPP and is mostly referred to as the Goel-Okumoto (1979) software reliability model, after the researchers Goel and Okumoto who first introduced it in 1979.

The model described in Equation (1) is a software reliability model and has been applied to a number of software testing environments and its application and usefulness in describing and assessing software failures has been conducted by various authors. For instance, [6] used Kolmorgorov-Sminorv goodness-of-fit test for checking the adequacy of the software reliability model and they also presented they also presented software failure data which, after study, depicted that the failure rate, *i.e.* the number of failures per hour, seemed to be decreasing with time. One-sample Bayesian predictive analysis on the model has also been conducted, [7]. However, there is no literature on two-sample Bayesian predictive analyses on the model.

This paper therefore focuses on two-sample Bayesian predictive analyses on the model whose intensity function is described in Equation (1). First, three issues in two-sample predictions that may be experienced during the development testing stage of the software are identified and their corresponding predictive distributions are thereafter developed in Section 2. The main results for the two-sample prediction are presented in Section 3. The developed methodologies are illustrated in Section 6 using simulated two-software failure data. Discussion is given in Section 7 and finally, mathematical proofs are given in the Appendix.

2. Issues in Two-Sample Software Reliability Prediction

In this section, three issues associated closely with software development testing process are presented and their predictive distributions are developed using Bayesian approach. For the purposes of the three predictive issues, it is assumed that a reliability growth testing is performed on a software and the cumulative number of failures of the software in the time interval $(0,t]$, denoted by $N(t)$ is observed. It is further assumed that $[N(t),t>0]$ follows the NHPP with intensity function given in Equation (1).

Let $0<t_1<t_2<\cdots$ be the observed failure times. Failure data is said to be failure-truncated when testing stops after a predetermined n number of failures occur. The n failure times are denoted by $Z_{\text{obs}}^{f}=[t_i]_{i=1}^{n}$ where $0<t_1<t_2<\cdots<t_n$. Failure data is said to be time truncated if testing stops at a predetermined time t. The corresponding observed failure data is denoted by $Z_{\text{obs}}^{t}=[n,t_1,\cdots,t_n;t]$, where $0<t_1<t_2<\cdots<t_n\le t$. Now, let us consider two software systems and assume that their cumulative inter-failure times obey the Goel-Okumoto (1979) software reliability model with observed data being either Z_{obs}^{f} or Z_{obs}^{t}. Based on Z_{obs}^{f} or Z_{obs}^{t}, we are interested in the following problems:

A1: How to predict the rth failure time of the second software system;

B1: How to predict the number of failures that will occur in the time interval $(0,t_2]$ for the second software system.

C1: How to predict the rth, $(1\le r\le m)$ failure time y_r of the second software system supposing that the number of failures in $(0,t_2]$ for the second software system is m but the exact occurrence times are unavailable.

Posterior and Predictive Distributions

Let Z_{obs} represent Z_{obs}^{f} or Z_{obs}^{t}. The joint density of Z_{obs} is therefore

$$f\left(Z_{\text{obs}} \mid \alpha, \beta\right) = \alpha^{n} \beta^{n} \mathrm{e}^{-\beta \sum_{i=1}^{n} t_i} \mathrm{e}^{-\alpha\left(1-\mathrm{e}^{-\beta T}\right)}. \tag{2}$$

Case 1: when the shape parameter β is known, the following non informative prior distribution of α is adopted

$$\pi(\alpha) \propto \frac{1}{\alpha}, \quad \alpha > 0. \tag{3}$$

Thus, the posterior distribution of α is given by

$$h\left(\alpha \mid Z_{\text{obs}}\right) = \left[\Gamma(n)\right]^{-1} \alpha^{n-1} \mathrm{e}^{-\alpha\left(1-\mathrm{e}^{-\beta T}\right)} \left(1-\mathrm{e}^{-\beta T}\right)^{n}. \tag{4}$$

Let y^{+} be the random variable being predicted. The posterior predictive distribution of y^{+} is then given as

$$f\left(y^{+} \mid Z_{\text{obs}}\right) = \int_{0}^{\infty} f\left(y^{+} \mid Z_{\text{obs}}, \alpha\right) h\left(\alpha \mid Z_{\text{obs}}\right) \mathrm{d}\alpha. \tag{5}$$

Hence the Bayesian UPL of y^{+} with level γ denoted as $y_U^{(\beta)}$ must satisfy

$$\gamma = \int_{-\infty}^{y_U^{(\beta)}} f\left(y^{+} \mid Z_{\text{obs}}\right) \mathrm{d}y^{+}. \tag{6}$$

3. Main Results for the Two-Sample Prediction

Proposition 1 (for issue A1)

The Bayesian UPL of y_r (*i.e.* the rth failure time of the second software system) with level γ when β is known is

$$\gamma = \Gamma(r+n)\left[\Gamma(r)\Gamma(n)\right]^{-1} \beta\left(1-\mathrm{e}^{-\beta T}\right)^{n} \int_{0}^{y_U^{(\beta)}} \frac{\mathrm{e}^{-\beta y_r}\left(1-\mathrm{e}^{-\beta y_r}\right)^{r-1}}{\left(2-\mathrm{e}^{-\beta y_r}-\mathrm{e}^{-\beta T}\right)^{r+n}} \mathrm{d}y_r. \tag{7}$$

Proposition 2 (for issue B1)

The probability that the number of failures $N(t_2)$ in the time interval $(0, t_2]$ for the second system does not exceed a pre-determined nonnegative integer m, when β is known is

$$\gamma = \sum_{k=0}^{m} \frac{\left(1-\mathrm{e}^{-\beta t_2}\right)^{k}}{\left(2-\mathrm{e}^{-\beta t_2}-\mathrm{e}^{-\beta T}\right)^{k}} \binom{n+k-1}{k} \frac{\left(1-\mathrm{e}^{-\beta T}\right)^{n}}{\left(2-\mathrm{e}^{-\beta t_2}-\mathrm{e}^{-\beta T}\right)^{n}}. \tag{8}$$

Proposition 3 (for issue C1)

Given that the number of failures in $(0, t_2]$ for the second software is m, the Bayesian UPL of y_r, $(1 \le r \le m)$ with level γ is $y_U^{(\beta)}$ satisfying the equation

$$\gamma = \frac{\left(1-\mathrm{e}^{-\beta y_U^{(\beta)}}\right)^{m}}{\left(1-\mathrm{e}^{-\beta t_2}\right)^{m}}. \tag{9}$$

4. Data Simulation

In this section, two software failure data sets are generated from the Goel-Okumoto (1979) software reliability model. The two data sets are simulated using the same model and parameters. The simulated data is used to illustrate the methodologies developed for the two sample Bayesian predictive analyses. The simulation procedure was as follows. The Goel-Okumoto (1979) model is as given in Equation (1).

The values of $\alpha = 100$ and $\beta = 0.0010741$ were fixed. A value of T from the set $S = \left[200, 500, 10^3, 5\times10^3, 10^4\right]$ was selected. The study used T = 200. The simulation used in the study is for illustrative purposes only. Nevertheless, there is a practical interpretation to the choices of α, β and T. Case studies e.g. [8] have shown that a software fault density at the system testing stage is frequently on the order of five bugs per 1000 lines of code. The choice of α = 100 could be thought of as symbolizing a practically large software system that is on the order of 20,000 lines of codes. The choices for β and T together imply that most of the failures will be discovered during the simulated test period. Following the forgoing discussion, the following steps were used to simulate two data sets from the Goel-Okumoto (1979) software reliability model:

Step 1: $t = 0,\ I = 0$.

Step 2: Generate a random number U.

Step 3: $t = t - \frac{1}{\lambda}\log U$, if $t > T$, stop.

Step 4: Generate a random number U.

Step 5: If $U \le \lambda(t)/\lambda$, set $I = I + 1$, $S(I) = t$.

Step 6: Go to step 2.

In the above steps, $\lambda(t)$ is known as the intensity function and λ is such that $\lambda(t) \le \lambda$. the last value of I represents the number of events time T, and $S(1), \cdots, S(I)$ are the event times. The above procedure of simulation is referred to as the thinning algorithm since it 'thins' the homogeneous Poisson points. It is the most efficient simulation procedure in the sense that it has the fewest number of rejected events times when $\lambda(t)$ is near λ throughout the interval [9]. Using the above procedure, the following two data sets were generated. The first data set is assumed to be the software failure times from the first software and the second data set is assumed to be the failure times from the second software.

Software one: 8.9345, 27.0177, 34.5816, 54.8606, 83.5715, 111.4006, 139.8851, 157.4743, 181.0868, 182.8410.

Software two: 2.3159, 16.2530, 20.5721, 23.3416, 42.8030, 46.7417, 61.0926, 63.8807, 75.1330, 80.7768, 97.3435, 117.9091, 129.3157, 138.0590, 169.3410, 172.7516, 186.0293, 193.1918, 198.5999.

5. Maximum Likelihood Estimation

Suppose the observation of the failure times occurred in the time interval $(0, T]$ where T = 200, and n faults were observed at the failure times $0 < t_1 < t_2 < \cdots < t_n < T$. The joint density of the failure times is as in Equation (2). Taking the log-likelihood function of Equation (2) gives

$$L = n\log\alpha + n\log\beta - \beta\sum_{i=1}^{n} t_i - \alpha\left(1 - e^{-\beta T}\right). \tag{10}$$

Differentiating L with respect to α and β and equating to zero gives

$$\frac{\partial L}{\partial \alpha} = \frac{n}{\alpha} - \left(1 - e^{-\beta T}\right) = 0 \tag{11}$$

$$\frac{\partial L}{\partial \beta} = \frac{n}{\beta} - \sum_{i=1}^{n} t_i - \alpha T e^{-\beta T} = 0. \tag{12}$$

Solving Equation (11) and Equation (12) we obtain

$$\hat{\alpha} = \frac{n}{1 - e^{-\hat{\beta} T}} \tag{13}$$

$$\frac{n}{\hat{\beta}} = \sum_{i=1}^{n} t_i + \frac{nTe^{-\hat{\beta} T}}{1 - e^{-\hat{\beta} T}}. \tag{14}$$

A necessary and sufficient condition for Equation (13) and Equation (14) to have a unique and positive solution $(\hat{\alpha}, \hat{\beta})$ is that $2\sum_{i=1}^{n} t_i / n < T$, [10]. That is, the ML estimates of α and β will exist only and only if two times the mean failure time is less than T. In most cases, the precision in the difference $1-\mathrm{e}^{-\hat{\beta}T}$ in the denominator of the second part in the RHS of Equation (14) will be poor since $\mathrm{e}^{-\hat{\beta}T}$ will always be very close to unity. This brings a numerical difficulty in finding the root of Equation (14). An alternative form of Equation (14) that overcomes this difficulty is

$$\frac{n}{\hat{\beta}}=\sum_{i=1}^{n}t_i+\frac{nT}{\sum_{j=1}^{\infty}\frac{\left(\hat{\beta}T\right)^j}{j!}}. \tag{15}$$

A numerical procedure known as the Newton Raphson method can be used to solve Equation (13) and Equation (15). The Newton Raphson method requires choosing of initial values of α and β. Consequently, $\alpha = 95$ and $\beta = 0.0012$ were chosen as the initial values. There is no any other explanation to the choosing of the initial values other than the fact that they are very close to the values $\alpha = 100$ and $\beta = 0.0010741$ that were used during the simulation of the two software failure data sets in Section 4.6. Consequently, the ML estimates $\hat{\alpha}=102.756$ and $\hat{\beta}=0.001022177$ for software one were obtained.

6. Real Example for Two-Sample Bayesian Prediction

Here, we use the two software data sets simulated in Section 4.6 to illustrate the developed propositions in Section 4.4 for two sample Bayesian prediction problems. Assuming that the two software systems were observed in the time interval $\left(0,200\right]$, and their successive failure times are given by:

Software one: 8.9345, 27.0177, 34.5816, 54.8606, 83.5715, 111.4006, 139.8851, 157.4743, 181.0868, 182.8410.

Software two: 2.3159, 16.2530, 20.5721, 23.3416, 42.8030, 46.7417, 61.0926, 63.8807, 75.1330, 80.7768, 97.3435, 117.9091, 129.3157, 138.0590, 169.3410, 172.7516, 186.0293, 193.1918, 198.5999.

The two software failure times are simulated from the same Goel-Okumoto (1979) software reliability model. The three issues in the two sample prediction in chapter three are addressed as follows:

Issue A2: First, we assume that the failure times of the second software were not observed. Based on the failure data of software one, the maximum likelihood estimate of β is given by 0.001022177. When β is known to be 0.001022177, and from Equation (7), the Bayesian UPL for the 15th failure time of the second software with level $\gamma=0.90$ is $y_U^{(\beta)}=33.737$ such that

$$\gamma=\Gamma\left(r+n\right)\left[\Gamma\left(r\right)\Gamma\left(n\right)\right]^{-1}\beta\left(1-\mathrm{e}^{-\beta T}\right)^n\int_0^{y_U^{(\beta)}}\frac{\mathrm{e}^{-\beta y_r}\left(1-\mathrm{e}^{-\beta y_r}\right)^{r-1}}{\left(2-\mathrm{e}^{-\beta y_r}-\mathrm{e}^{-\beta T}\right)^{r+n}}\mathrm{d}y_r .$$

Issue B2: if $\beta=0.001022177$, then from Equation (8), the probability that the number of failures in the time interval $\left(0,200\right]$ for the second software not exceeding a pre-determined nonnegative integer $m=16$, is $\gamma=0.9157$.

Issue C2: suppose that the number of observed failures of the second software during $\left(0,200\right]$ is $m=15$. Based on the failure data of the second software, if $\beta=0.001022177$, then from Equation (9), the Bayesian UPL for y_{15} with level $\gamma=0.90$ is $y_U^{\beta}=199.00$

7. Discussion

Several issues may arise during development testing of a software system especially when the Goel-Okumoto (1979) software reliability model has been used to model the failure process of the software system. This paper has provided solutions to three issues associated closely with software development testing process. Bayesian approach with non-informative prior has been used to address the three issues. Explicit solutions to the issues have been obtained. These solutions may prove useful to software engineers in determining when to modify, debug and terminate the software development testing process.

Non-informative prior has been used in this paper to develop the methodologies to the said three issues. However, informative priors may also prove useful in deriving the methodologies. We leave this open for future research. Further, this paper has only derived the methodologies for known shape parameter β. It may be interesting to derive solutions for the same problems for the case when the shape parameter β is unknown. The procedures presented in this paper can also be extended to other NHPPs such as the Musa-Okumoto process, the delayed *S*-shaped process and the Cox-Lewis process.

References

[1] Nuria, T.R. (2011) Stochastic Comparisons and Bayesian Inference in Software Reliability. Ph.D. Thesis, Universidad Carlos III de Madrid, Madrid.

[2] Satya, P., Bandla, S.R. and Kantham, R.R.L. (2011) Assessing Software Reliability Using Inter Failures Time Data. *International Journal of Computer Applications*, **18**, 975-978.

[3] Daniel, R.J. and Hoang, P. (2001) On the Maximum Likelihood Estimates for the Goel-Okumoto Software Reliability Model. *The American Statistician*, **55**, 219-222. http://dx.doi.org/10.1198/000313001317098211

[4] Meth, M. (1992) Reliability Growth Myths and Methodologies: A Critical View. *Proceedings of the Annual Reliability and Maintainability Symposium*, New York, 230-238.

[5] Yu, J.-W., Tian, G.-L. and Tang, M.-L. (2007) Predictive Analyses for Nonhomogeneous Poisson Processes with Power Law Using Bayesian Approach. *Computational Statistics & Data Analysis*, **51**, 4254-4268. http://dx.doi.org/10.1016/j.csda.2006.05.010

[6] Razeef, M. and Mohsin, N. (2012) Software Reliability Growth Models: Overview and Applications. *Journal of Emerging Trends in Computing and Information Sciences*, **3**, 1309-1320.

[7] Akuno, A.O., Orawo, L.A. and Islam, A.S. (2014) One-Sample Bayesian Predictive Analyses for an Exponential Non-Homogeneous Poisson Process in Software Reliability. *Open Journal of Statistics*, **4**, 402-411. http://dx.doi.org/10.4236/ojs.2014.45039

[8] Musa, J. (1987) Software Reliability: Measurement, Prediction, Application. McGraw-Hill, New York.

[9] Sheldon, R. (2002) Simulation. 3rd Edition, Academic Press, Waltham.

[10] Hossain, S.A. and Dahiya, R.C. (1993) Estimating the Parameters of a Non-Homogenous Poisson-Process Model for Software Reliability. *IEEE Transactions on Reliability*, **42**, 604-612.

Appendix (Proofs of Proposition 1-3)

The following identity is used in proving some of the propositions. The identity is given without proof.

$$\int_{D(m;a,b)} \mathrm{d}F(t_1)\cdots \mathrm{d}F(t_m) = \left[F(b)-F(a)\right]^m \big/ m! \tag{A.1}$$

where m is any positive integer, a and b are two real numbers $a<b$, $F(t)$ is an increasing and differentiable function, and $D(m;a,b)=\left[(t_1,\cdots,t_m): a<t_1<\cdots<t_m<b\right]$.

Proof of Proposition 1

We know that given $N(t)=n$, the n failure times $x_1,x_2,\cdots,x_n$ have the same distribution as the order statistics corresponding to n independent random variables with density $f(u)=\alpha\beta\mathrm{e}^{-\beta u}\Big/\int_0^t \alpha\beta\mathrm{e}^{-\beta u}\mathrm{d}u$, $0\le u\le t$ which reduces to $f(u)=\dfrac{\beta\mathrm{e}^{-\beta u}}{\left(1-\mathrm{e}^{-\beta t}\right)}$. This implies that $f(x)=\dfrac{\beta\mathrm{e}^{-\beta x}}{\left(1-\mathrm{e}^{-\beta t}\right)}$. This is to say that $\left(x_1,\cdots,x_n \mid N(t)=n\right)\sim n!\prod_{i=1}^{n} f(x_i)=n!\prod_{i=1}^{n}\left[\dfrac{\beta\mathrm{e}^{-\beta x_i}}{1-\mathrm{e}^{-\beta t}}\right]=n!\dfrac{\beta^n\mathrm{e}^{-\beta\sum_{i=1}^{n}x_i}}{\prod_{i=1}^{n}\left(1-\mathrm{e}^{-\beta t}\right)}$. Consequently,

$$\left(x_1,x_2,\cdots,x_n \mid x_n\right)\sim(n-1)!\frac{\beta^{n-1}\mathrm{e}^{-\beta\sum_{i=1}^{n-1}x_i}}{\prod_{i=1}^{n-1}\left(1-\mathrm{e}^{-\beta x_n}\right)}. \tag{A.2}$$

The joint density of $(x_1,\cdots,x_n)$ is also given by Equation (2). Equation (2) divided by Equation (A.2) yields the density of x_n and we have

$$\begin{aligned} f(x_n\mid\alpha,\beta) &= \frac{\alpha^n\beta^n\mathrm{e}^{-\beta\sum_{i=1}^{n}x_i}\mathrm{e}^{-\alpha\left(1-\mathrm{e}^{-\beta x_n}\right)}}{(n-1)!\beta^{n-1}\mathrm{e}^{-\beta\sum_{i=1}^{n-1}x_i}\Big/\prod_{i=1}^{n-1}\left(1-\mathrm{e}^{-\beta x_n}\right)} = \frac{\alpha^n\beta^n\mathrm{e}^{-\beta\sum_{i=1}^{n}x_i}\mathrm{e}^{-\alpha\left(1-\mathrm{e}^{-\beta x_n}\right)}\left(1-\mathrm{e}^{-\beta x_n}\right)^{n-1}}{(n-1)!\beta^{n-1}\mathrm{e}^{-\beta\sum_{i=1}^{n-1}x_i}} \\ &= \left[(n-1)!\right]^{-1}\beta\mathrm{e}^{-\beta x_n}\mathrm{e}^{-\alpha\left(1-\mathrm{e}^{-\beta x_n}\right)}\left(1-\mathrm{e}^{-\beta x_n}\right)^{n-1}. \end{aligned} \tag{A.3}$$

Replacing x_n by y_r, for the second system, we have the density of y_r being given as

$$f(y_r\mid\alpha,\beta)=\left[\Gamma(r)\right]^{-1}\beta\alpha^r\mathrm{e}^{-\beta y_r}\mathrm{e}^{-\alpha\left(1-\mathrm{e}^{-\beta y_r}\right)}\left(1-\mathrm{e}^{-\beta y_r}\right)^{r-1}. \tag{A.4}$$

From Equation (5) and Equation (A.4) we have

$$\begin{aligned} & f(y_r\mid Z_{\text{obs}}) \\ &= \int_0^\infty \left[\Gamma(r)\Gamma(n)\right]^{-1}\beta\alpha^r\mathrm{e}^{-\beta y_r}\mathrm{e}^{-\alpha\left(1-\mathrm{e}^{-\beta y_r}\right)}\left(1-\mathrm{e}^{-\beta y_r}\right)^{r-1}\alpha^{n-1}\mathrm{e}^{-\alpha\left(1-\mathrm{e}^{-\beta T}\right)}(1-\mathrm{e}^{-\beta T})^n\mathrm{d}\alpha \\ &= \left[\Gamma(r)\Gamma(n)\right]^{-1}\left(1-\mathrm{e}^{-\beta y_r}\right)^{r-1}\left(1-\mathrm{e}^{-\beta T}\right)^n\mathrm{e}^{-\beta y_r}\beta\int_0^\infty\alpha^{r+n-1}\mathrm{e}^{-\alpha\left(2-\mathrm{e}^{-\beta y_r}-\mathrm{e}^{-\beta T}\right)}\mathrm{d}\alpha \\ &= \left[\Gamma(r)\Gamma(n)\right]^{-1}\left(1-\mathrm{e}^{-\beta y_r}\right)^{r-1}\left(1-\mathrm{e}^{-\beta T}\right)^n \\ &\quad\cdot\mathrm{e}^{-\beta y_r}\beta\frac{\Gamma(r+n)}{\left(2-\mathrm{e}^{-\beta y_r}-\mathrm{e}^{-\beta T}\right)^{r+n}}\int_0^\infty\frac{\left(2-\mathrm{e}^{-\beta y_r}-\mathrm{e}^{-\beta T}\right)^{r+n}}{\Gamma(r+n)}\alpha^{r+n-1}\mathrm{e}^{-\alpha\left(2-\mathrm{e}^{-\beta y_r}-\mathrm{e}^{-\beta T}\right)}\mathrm{d}\alpha \\ &= \left[\Gamma(r)\Gamma(n)\right]^{-1}\left(1-\mathrm{e}^{-\beta y_r}\right)^{r-1}\left(1-\mathrm{e}^{-\beta T}\right)^n\mathrm{e}^{-\beta y_r}\beta\frac{\Gamma(r+n)}{\left(2-\mathrm{e}^{-\beta y_r}-\mathrm{e}^{-\beta T}\right)^{r+n}}. \end{aligned} \tag{A.5}$$

From Equation (6) and Equation (A.5), we have

$$\gamma=\Gamma(r+n)\left[\Gamma(r)\Gamma(n)\right]^{-1}\beta\left(1-e^{-\beta T}\right)^{n}\int_{0}^{y_U^{(\beta)}}\frac{e^{-\beta y_r}\left(1-e^{-\beta y_r}\right)^{r-1}}{\left(2-e^{-\beta y_r}-e^{-\beta T}\right)^{r+n}}dy_r\ . \quad (A.6)$$

Equation (A.6) implies the formula in Equation (7) .

Proof of Proposition 2

The study is interested in predicting the number of failures (denoted by $N(t_2)$) of the second system occurring in the time interval $(0,t_2]$. Obviously,

$$\Pr\left[N(t_2)=k\,|\,\alpha,\beta\right]=\left[k!\right]^{-1}\alpha^{k}\left(1-e^{-\beta t_2}\right)^{k}e^{-\alpha\left(1-e^{-\beta t_2}\right)},\quad k=0,1,\cdots. \quad (A.7)$$

For any level γ, the Bayesian Upper prediction limit for $N(t_2)$ is Nu satisfying $\gamma=\Pr\left[N(t_2)\le Nu\,|\,Z_{obs}\right]$.

Here, an equivalent problem is considered. For any given positive integer m, we want to compute the probability that $N(t_2)\le m$ *i.e.*

$$\gamma=\Pr\left[N(t_2)\le m\,|\,Z_{obs}\right]=\sum_{k=0}^{m}\Pr\left[N(t_2)=k\,|\,Z_{obs}\right]. \quad (A.8)$$

When β is known, from Equation (A.7) and Equation (4) we have

$$\Pr\left[N(t_2)=k\,|\,Z_{obs}\right]=\int_0^\infty\Pr\left[N(t_2)=k\,|\,\alpha\right]h(\alpha\,|\,Z_{obs})d\alpha=\frac{\left(1-e^{-\beta t_2}\right)^{k}}{\left(2-e^{-\beta t_2}-e^{-\beta T}\right)^{k}}\frac{\Gamma(k+n)}{k!\Gamma(n)}\frac{\left(1-e^{-\beta T}\right)^{n}}{\left(2-e^{-\beta t_2}-e^{-\beta T}\right)^{n}}. \quad (A.9)$$

Rearranging Equation (A.9) we obtain

$$\gamma=\sum_{k=0}^{m}\frac{\left(1-e^{-\beta t_2}\right)^{k}}{\left(2-e^{-\beta t_2}-e^{-\beta T}\right)^{k}}\binom{n+k-1}{k}\frac{\left(1-e^{-\beta T}\right)^{n}}{\left(2-e^{-\beta t_2}-e^{-\beta T}\right)^{n}}. \quad (A.10)$$

Equation (A.9) implies the formula in Equation (8) .

Proof of Proposition 3

First, we want to find the conditional density of y_r given $N(t_2)=m$, from Equation (2),

$$f\left(y_1,\cdots,y_m;N(t_2)=m\right)=\alpha^{m}\beta^{m}e^{-\beta\sum_{i=1}^{m}y_i}e^{-\alpha\left(1-e^{-\beta t_2}\right)}. \quad (A.11)$$

After integrating Equation (A.11) with respect to $y_1,\cdots,y_{r-1}$ we obtain

$$f\left(y_r,\cdots,y_m;N(t_2)=m\right)\overset{A.1}{=}\frac{1}{(r-1)!}\alpha^{m}\beta^{m-r+1}\left(1-e^{-\beta y_r}\right)^{r-1}e^{-\beta y_r}e^{-\beta\sum_{i=r+1}^{m}y_i}e^{-\alpha\left(1-e^{-\beta t_2}\right)}. \quad (A.12)$$

Further integrating Equation (A.12) with respect to $(y_{r+1},\cdots,y_m)$, yields

$$\begin{aligned}f\left(y_r;N(t_2)=m\right)&=\left[\beta^{m-r}(m-1)!(r-1)!\right]^{-1}\alpha^{m}\beta^{m-r+1}\left(1-e^{-\beta y_r}\right)^{r-1}e^{-\beta y_r}\left(e^{-\beta y_r}-e^{-\beta t_2}\right)^{m-r}e^{-\alpha\left(1-e^{-\beta t_2}\right)}\\&=\left[(m-1)!(r-1)!\right]^{-1}\alpha^{m}\beta\left(1-e^{-\beta y_r}\right)^{r-1}e^{-\beta y_r}\left(e^{-\beta y_r}-e^{-\beta t_2}\right)^{m-r}e^{-\alpha\left(1-e^{-\beta t_2}\right)}.\end{aligned} \quad (A.13)$$

Therefore, the conditional density of y_r given $N(t_2)=m$ is

$$f\left(y_r\,|\,N(t_2)=m\right)=\frac{f\left(y_r;N(t_2)=m\right)}{\Pr\left\{N(t_2)=m\right\}}\overset{A.7}{=}\frac{m!}{(m-r)!(r-1)!\left(1-e^{-\beta t_2}\right)^{m}}\beta\left(1-e^{-\beta y_r}\right)^{r-1}e^{-\beta y_r}\left(e^{-\beta y_r}-e^{-\beta t_2}\right)^{m-r},$$

$$y_r<t_2. \quad (A.14)$$

Which is independent of α. When β is known, Equation (5) can be re-written as

$$f\left(y_r \middle| N(t_2)=m, Y_{\text{obs}}\right)=\int_0^\infty f\left(y_r \middle| N(t_2)=m,\alpha\right)h\left(\alpha \middle| N(t_2)=m, Y_{\text{obs}}\right)\mathrm{d}\alpha$$

where $f\left(y_r \middle| N(t_2)=m,\alpha\right)=f\left(y_r \middle| N(t_2)=m\right)$ is given by Equation (A.14) and $h\left(\alpha \middle| N(t_2)=m, Z_{\text{obs}}\right)=h\left(\alpha \middle| Z_{\text{obs}}\right)$ is given by Equation (4). Hence $f\left(y_r \middle| N(t_2)=m, Z_{\text{obs}}\right)=f\left(y_r \middle| N(t_2)=m\right)$. Given $N(t_2)=m$, the Bayesian UPL of y_r with level γ is $y_U^{(\beta)}$ such that

$$\gamma=\int_0^{y_U^{(\beta)}} f\left(y_r \middle| N(t_2)=m, Z_{\text{obs}}\right)\mathrm{d}y_r=\frac{m!}{(m-r)!(r-1)!\left(1-\mathrm{e}^{-\beta t_2}\right)^m}\int_0^{y_U^{(\beta)}}\beta\left(1-\mathrm{e}^{-\beta y_r}\right)^{r-1}\mathrm{e}^{-\beta y_r}\left(\mathrm{e}^{-\beta y_r}-\mathrm{e}^{-\beta t_2}\right)^{m-r}\mathrm{d}y_r. \quad \text{(A.15)}$$

If $r=m$, Equation (A.15) becomes

$$\gamma=\frac{m!\beta}{(m-1)!\left(1-\mathrm{e}^{-\beta t_2}\right)^m}\int_0^{y_U^{(\beta)}}\left(1-\mathrm{e}^{-\beta y_m}\right)^{m-1}\mathrm{e}^{-\beta y_m}\mathrm{d}y_m. \quad \text{(A.16)}$$

Solving the integral part of Equation (A.16), we obtain

$$\gamma=\frac{\left(1-\mathrm{e}^{-\beta y_U^{(\beta)}}\right)^m}{\left(1-\mathrm{e}^{-\beta t_2}\right)^m}. \quad \text{(A.17)}$$

Thus, the Bayesian UPL of y_m with confidence level γ is $y_U^{(\beta)}$ that satisfies Equation (A.17).