

A Bayesian Approach to Survival Analysis of Gompertz Model with R and JAGS

Najrullah Khan^{1,*} and Athar Ali Khan²

¹School of Basic Sciences and Technology, IIMT University, Meerut, 250001, India

²Department of Statistics and Operations Research, Aligarh Muslim University, Aligarh, 202002, India

Received: 22 Sep. 2022, Revised: 22 Oct. 2022, Accepted: 30 Nov. 2022

Published online: 1 Mar. 2023

Abstract: The Gompertz distribution of lifetimes was introduced in 1825 and a probability model was derived for studying human mortality. In this paper, an attempt has been made to Bayesian fitting of Gompertz distribution. The Bayesian approach is implemented with R and JAGS for analytic and simulation tools. Our main target is to explore the use of Laplace Approximation for Bayesian analysis. A real right censored data is used. Finally, the model is compared with Weibull distribution using Bayesian prediction tools.

Keywords: Bayesian Inference, Posterior, LaplaceApproximation, LaplacesDemon, JAGS, Simulation, R.

1 Introduction

For many statisticians analysis of survival data has become a topic of considerable interest. In this paper we consider the real right censored data and deal in inference for important parametric models - Gompertz and Weibull. Among several survival models we used two models that quite effectively analyze skewed data set and give best fit. The analysis of time to event data is, generally called survival analysis. The event can be the development of disease, response to a treatment, relapse, or death. Survival data are collected over a finite period of time and consequently the failure times may not be observed for all individuals, which results censored data. The problem of analyzing time to event arises in a number of applied fields, such as medicine, biology, public health, economics and demography. Bayesian analysis of survival data has received much recent attention due to advanced computational and modeling techniques. Bayesian methods are now becoming quite common for survival data analysis and have made their way into the medical and public health arena. Our main target is to explore the use of Laplace Approximation for Bayesian analysis. A Bayesian approach may help using MCMC techniques and available softwares e.g. LaplacesDemon and JAGS. The function of LaplaceApproximation of LaplacesDemon approximates the posterior results analytically and then after convergence it gives simulated results using sampling importance resampling (SIR). Another important function is LaplacesDemon, which maximizes the logarithm of unnormalized joint posterior density using one of the Morkov chain Monte Carlo(MCMC) algorithms. JAGS is Just Another Gibbs Sampler that was mainly written by Martyn Plummer. R2jags is an R package that allows running JAGS models from within R. R2jags package is used for the simulation from posterior density. Thus, Bayesian analysis of survival models have been made with following objectives:

- To define a Bayesian model, that is, specification of likelihood and prior distribution.
- To write down the R and JAGS code for approximating posterior densities with LaplaceApproximation and simulation tools.
- To illustrate numeric as well as graphic summaries of posterior densities.

2 The Gompertz Distribution

The Gompertz distribution (1825) is generally used as survival model. It was introduced as a parametric model as an approximation to the true law of mortality. Gompertz worked closely with data on mortality, and developed a

* Corresponding author e-mail: najrullahk90@gmail.com

theoretical basis of his distribution. The probability density function (pdf) of Gompertz distribution is given by

$$f(t) = b \exp(t/\theta) \exp(b\theta(1 - \exp(t/\theta))), \quad 0 < t < \infty \quad (1)$$

where $b > 0$ is shape and $\theta > 0$ is scale parameter.
The cdf of Gompertz distribution is given by

$$F(t) = 1 - \exp(b\theta(1 - \exp(t/\theta))) \quad (2)$$

survival function is

$$S(t) = \exp(b\theta(1 - \exp(t/\theta))) \quad (3)$$

and hazard function is

$$h(t) = b \exp(t/\theta) \quad (4)$$

2.1 Functions for Gompertz distribution in R

1. R code for probability density function is

```
dgomp<-function(x,b,theta) {
  d<-b*exp(x/theta)*exp(b*theta*(1-exp(x/theta)))
  return(d)
}
```

2. R code for cumulative distribution function is

```
pgomp<-function(x,b,theta) {
  p<-1-exp(b*theta*(1-exp(x/theta)))
  return(p)
}
```

3. R code for random generation function is

```
rgomp<-function(n,b,theta) {
  u<-runif(n)
  x<-theta*log(1-(log(1-u)/b*theta))
  return(x)
}
```

4. R code for survival function is

```
sgomp<-function(x,b,theta) {
  s<-exp(b*theta*(1-exp(x/theta)))
  return(s)
}
```

5. R code for hazard function

```
hgomp<-function(x,b,theta) {
  h<-dgomp(x,b,theta)/sgomp(x,b,theta)
  return(h)
}
```

3 Weibull Distribution

Weibull distribution plays very important role in survival analysis, that introduced by W. Weibull in 1951 in the context of industrial reliability testing. The Weibull distribution has been used extensively in medical studies also. The probability density function (pdf) of Weibull distribution is given by

$$f(t) = (b/\lambda)(t/\theta)^{(b-1)} \exp[-(t/\theta)^b], \quad 0 < t < \infty \quad (5)$$

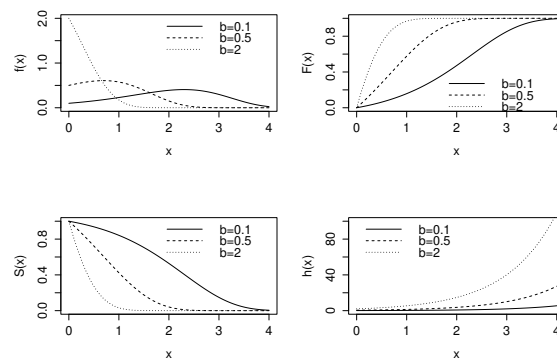


Fig. 1: The pdf, cdf, survival and hazard curves of Gompertz distribution for $\theta = 1$ and for different values of b .

where $b > 0$ is shape and $\theta > 0$ is scale parameter with cumulative distribution function is

$$F(t) = 1 - \exp[-(t/\theta)^b] \tag{6}$$

survival function is

$$S(t) = \exp[1 - (t/\theta)^b] \tag{7}$$

and hazard function is

$$h(t) = (b/\theta)(t/\theta)^{(b-1)} \tag{8}$$

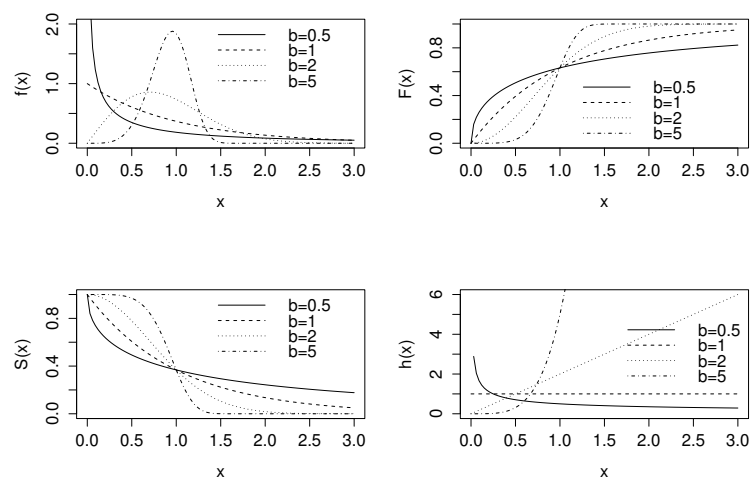


Fig. 2: The pdf, cdf, survival and hazard curves of Weibull distribution for $\theta = 1$ and for different values of b

4 Bayesian Analysis of Gompertz Model

4.1 The Model

The likelihood function of right censored data is given by

$$L = \prod_{i=1}^n [f(y_i)]^{\delta_i} [S(y_i)]^{1-\delta_i} \quad (9)$$

where $\delta_i=0$ is censored and $\delta_i=1$ is uncensored.

So, the likelihood function is given by

$$L = \prod_{i=1}^n [b \exp(t/\theta) \exp(b\theta(1 - \exp(t/\theta)))]^{\delta_i} [\exp(b\theta(1 - \exp(t/\theta)))]^{1-\delta_i}$$

4.2 Priors

Let $Y \sim \text{Gompertz}(b, \theta)$. Prior probabilities are specified for b and β :

$$b \sim \text{halfCauchy}(\alpha)$$

$$f(b) = \frac{2 \times \alpha}{\pi(b^2 + \alpha^2)}, \quad b > 0$$

The half-Cauchy distribution with scale parameter $\alpha=25$, used as a noninformative prior distribution for shape parameter. [4] suggested to use uniform or half Cauchy if more information is essential. For $\alpha=25$, the half-Cauchy distribution becomes almost flat (see FIGURE 3).

For positive parameter θ , logarithm link function is used to spread them on the whole real line, i.e.

$$\log \theta = X\beta$$

$$\theta = \exp(X\beta)$$

where X is model matrix, β is the regression coefficients.

Parameter β is assigned a weak informative Gaussian prior probability distribution. Assuming that β_j' is independently distributed as normal with mean=0 and standard deviation=1000, so that a flat prior can be obtained as (see FIGURE 3)

$$\beta_j \sim N(0, 1000).$$

Now, the joint posterior density is

$$\begin{aligned} p(b, \beta | y, X) &\propto L(y, X | b, \beta) \times p(b) \times p(\beta) \\ &\propto \prod_{i=1}^n [b \exp(t/\theta) \exp(b\theta(1 - \exp(t/\theta)))]^{\delta_i} [\exp(b\theta(1 - \exp(t/\theta)))]^{1-\delta_i} \\ &\quad \times \frac{2 \times \alpha}{\pi(b^2 + \alpha^2)} \times \prod_{j=1}^J \frac{1}{\sqrt{2\pi} \times 10^3} \exp\left(-\frac{1}{2} \frac{\beta_j^2}{10^3}\right) \end{aligned} \quad (10)$$

Marginal for b

$$p(b | y, X) = \int_0^\infty p(b, \beta | y, X) d\beta \quad (11)$$

Marginal for β

$$p(\beta | y, X) = \int_{-\infty}^\infty p(b, \beta | y, X) db \quad (12)$$

Closed forms for these densities are not available, so numerical integration or MCMC methods are required to evaluate them. Three methods can be used to solve the complex numerical integration including censoring mechanism by using LaplaceApproximation, LaplacesDemon and Jags functions.

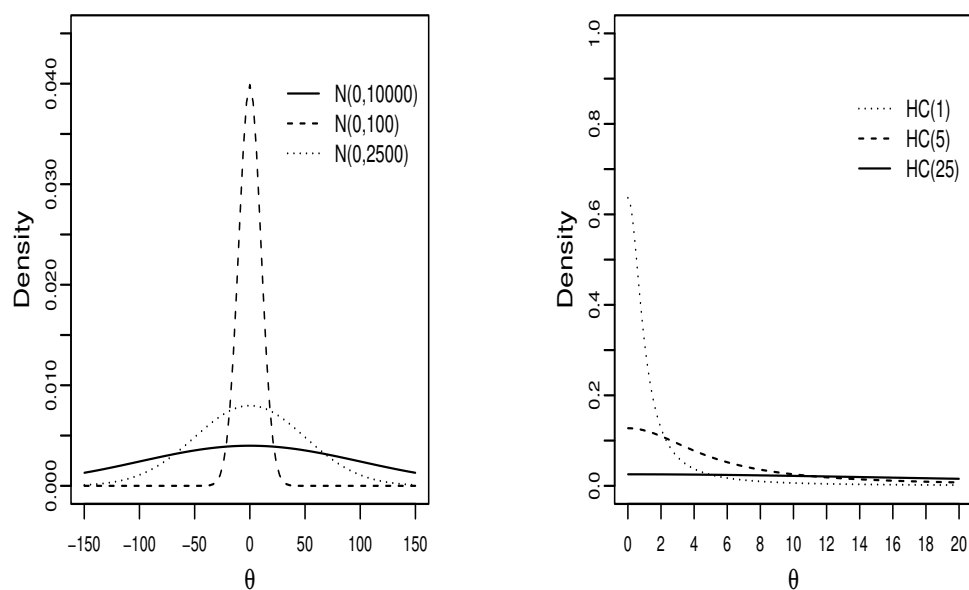


Fig. 3: Weakly informative priors for the parameters

4.3 Prognosis for women with breast cancer data

Breast cancer is one of the most common forms of cancer occurring in women living in the Western World. Survival data set is carried out at the Middlesex Hospital, documented in Leathem and Brooks (1987) and discussed by [1], was to evaluate a histochemical marker that discriminates between primary breast that has metastasised and that which has not. The marker under study was lectin from the albumin gland of the Roman snail, *Helix pomatia*, known as *Helix pomatia* agglutinin or HPA. The marker binds to those breast cancer cells associated with metastasis to local lymph nodes, and the HPA stained cells can be identified by microscopic examination. The data given in table 1 refer to survival times in months of women who had received a simple or radical mastectomy to treat a tumor of Grade II, III or IV, between January 1969 and December 1971. In the table, the survival times of each woman are classified according to whether their tumor was positively or negatively stained. Censored survival times are labelled with an asterisk.

Table 1: Survival times of women with tumours that were negatively or positively stained with HPA.

Negatively Stained	Positively Stained
23	5 68
47	8 71
69	10 76*
70*	13 105*
71*	18 107*
100*	24 109*
101*	26 113
148	26 116*
181	31 118
198*	35 143
208*	40 154*
212*	41 162*
224*	48 188*
	50 212*
	59 217*
	61 225*

4.4 Implementation using *LaplacesDemon*

Bayesian modeling of Gompertz(b, θ) distribution and fitting model using *LaplaceApproximation* and *LaplacesDemon* functions.

4.4.1 Creation of data for *LaplacesDemon*

The function of *LaplaceApproximation* requires data that is specified in a list. Response variable or survival times of women with tumors that were negatively and positively stained with HPA. Since an intercept term will be included, a vector of 1's is inserted into model matrix X, naming of predictors, naming of the parameters, information regarding censoring.

```
require(LaplacesDemon)
y<-c(23,47,69,70,71,100,101,148,181,198,208,212,224,5,
5,10,13,18,24,26,26,31,35,40,41,48,50,59,61,68,71,78,
105,107,109,113,116,118,143,154,162,188,212,217,225)
x1<-c(rep(0,13),rep(1,32))
censor<-c(1,1,1,0,0,0,0,1,1,0,0,0,0,rep(1,18),0,0,0,0,1,0,1,1,rep(0,6))
X<-cbind(1,x1)
J<-2
mon.names<-c("LP","b")
parm.names<-as.parm.names(list(beta=rep(0,J),log.b=0))
pos.beta<-grep("beta",parm.names)
pos.b<-grep("log.b",parm.names)
MyData<-list(J=J,X=X,mon.names=mon.names,parm.names=parm.names,
pos.beta=pos.beta,pos.b=pos.b,y=y,censor=censor)
```

Here $J=2$ independent variables (including the intercept), one for each column in model matrix X. *mon.names* is the monitored variables of log posterior LP and b . The *grep* function is used to populate *pos.beta* and *pos.b*. *pos.beta* and *pos.b* explain the position of β and b respectively in *parm.names*. However, there are 3 parameters including $\log.b$ are specified in a vector *parm.names*. Atlast, all these things are combined in a list of *MyData*.

4.4.2 Model Specification

To use *LaplacesDemon* package, Bayesian model for which likelihood is specified (e.g., [8]). We must specify a model

$$y \sim \text{Gompertz}(b, \theta)$$

with log-link function

$$\log \theta = X\beta.$$

The response variable, y follows Gompertz distribution with shape parameter b and scale parameter θ , the scale parameter θ is equal to the cross product of design matrix X and the parameter β . Prior probabilities are specified for b and β are

$$\beta_j \sim N(0,1000), \quad j = 1, \dots, J$$

$$b \sim HC(25).$$

The large variance indicates a lot of uncertainty about each β'_j and hence is a weakly informative prior distribution. Similarly, half-Cauchy is weakly informative prior for b [15].

The full Bayesian model code for the regression analysis is described below:

```

Model<-function (parm, Data)
{
  ## Parameters
  beta<-parm[Data$pos.beta]
  b<-exp (parm[Data$pos.b])
  # Log(Prior Densities)
  beta.prior<-dnormv (beta, 0, 1000, log=TRUE)
  b.prior<-dhalfcauchy (b, 25, log=TRUE)

  ## Loglikelihood
  mu<-tcrossprod (beta, Data$X)
  theta<-exp (mu)
  LL<-censor*log (b) + (y/theta) + (b*theta*(1-exp (y/theta)))
  + (1-censor) * (b*theta*(1-exp (y/theta)))
  LL<-sum (LL)

  ## log-Posterior
  LP<-LL+beta.prior+b.prior
  #LP<-LL
  LP<-sum (LP)
  Modelout<-list (LP=LP, Dev=-2*LL, Monitor=c (LP, b) ,
  yhat=rgomp (length (y) , b, theta) , parm=parm)
  return (Modelout)
}

```

4.5 Initial Values

LaplaceApproximation requires a vector of initial values for estimation of parameters. When all initial values are set to zero LaplaceApproximation will optimize initial values using trust region algorithm. Generally, parameter beta has been set equal to zero and parameter log.b has been set equal to log(1), which is also zero. However, instead of taking this default guess we have taken regression coefficients obtained from fitting lm function for

$$\log(y) = \beta_0 + \beta_1 x_1.$$

```
Initial.Values <-c (coef (lm (log (y) ~x1)) , log (1))
```

4.6 Fitting With LaplaceApproximation

The LaplaceApproximation is a family of asymptotic techniques used to approximate the integrals. It approximates accurately unimodal posterior moments and marginal posterior distributions in many cases. This function deterministically maximizes the logarithm of unnormalized joint posterior density with one of several optimization algorithms. The goal of LaplaceApproximation is to estimate the posterior mode and variance of each parameter. The function and arguments are as follows :

```
LaplaceApproximation (Model, parm, Data, Interval=1.0E-6,
Iterations=100, Method="SPG", Samples=1000, CovEst="Hessian",
sir=TRUE, Stop.Tolerance=1.0E-5, CPUs=1, Type="PSOCK")
```

First argument Model defines the model to be implemented, which contains specification of likelihood and prior. LaplaceApproximation passes two arguments to the model function, parm and Data. The parm argument requires a vector of initial values equal in length to the number of parameters. Data argument accepts a list of data. By default method is Method=SPG. We have found that trust region is better choice compared to other methods. The Trust Region algorithm of [11] is used.

To fit the model with LaplaceApproximation

```
Mg<-LaplaceApproximation(Model,Initial.Values,
Data=MyData,Method="TR",Iterations=1000)
print(Mg)
```

4.6.1 Summarizing Output

Table 2 and Table 3 describe the analytic and simulation posterior summaries, respectively.

Table 2: Posterior mode, posterior sd and their quantiles

Parameter	Mode	SD	LB	UB
beta[1]	3.66	0.12	3.41	3.91
beta[2]	-0.08	0.06	-0.21	0.05
log.b	-9.00	0.52	-10.04	-7.96

Table 3: Posterior mode, posterior sd and their quantiles

Parameter	Mode	SD	MCSE	ESS	LB	Median	UB
beta[1]	3.6787	0.1125	0.0035	1000.00	3.4430	3.6791	3.9039
beta[2]	-0.0885	0.0630	0.0019	1000.00	-0.2161	-0.0859	0.0228
log.b	-8.9945	0.4828	0.1527	1000.00	-9.9753	-8.987	-8.0225
Deviance	287.2947	1.0772	0.03406	1000.00	286.0408	287.0154	290.0349
LP	-303.3911	1.0772	0.03406	1000.00	-306.4430	-303.1087	-302.1339
b	0.0001	0.00006	0.000002	1000.00	0.00004	0.0001	0.0003

4.7 Fitting with *LaplaceDemon*

The *LaplaceDemon* function is the main function of *LaplaceDemon* package. This function maximizes the logarithm of the unnormalized joint posterior density with MCMC and provides samples of the marginal posterior distributions, deviance, and other monitored variables. The *LaplaceDemon* function for this model, simulates the data from posterior density with Independent Metropolis (IM) algorithm. The main arguments of the *LaplaceDemon* can be seen by using the function args as:

```
LaplaceDemon(Model, Data, Initial.Values, Covar= NULL,
Iterations= 10000, Status= 1000, Thinning= 100, Algorithm= "RWM", Specs=
NULL,...)
```

The arguments *Model* and *Data* specify the model to be implemented and list of data, which are specified in the previous section, respectively. The argument *Iterations* accepts integers larger than 10, and determines the number of iterations that Laplace's Demon will update the parameters while searching for target distributions.

The function *LaplaceDemon* is used to analyze the same data.

```
Initial.Values<-as.initial.values(Mg)
Mgld <- LaplaceDemon(Model, Data=MyData, Initial.Values,
Covar=Mg$Covar, Iterations=1000, Status=100, Thinning=1,Algorithm="IM",
Specs=list(mu=Mg$Summary1[1:length(Initial.Values),1]))
print(Mgld)
```


4.7.1 Summarizing Output

Table 4 describes the simulated posterior summary by Independent Metropolis algorithm.

Table 4: Posterior mean, posterior sd and their quantiles

Parameter	Mean	SD	MCSE	ESS	LB	Median	UB
beta[1]	3.6609	0.04793	0.0016	1000.00	3.5718	3.6598	3.7572
beta[2]	-0.08482	0.0272	0.0010	1000.00	-0.1396	-0.0872	-0.03274
log.b	-9.0177	0.1987	0.0067	1000.00	-9.4017	-9.0165	-8.6157
Deviance	286.1642	0.2154	0.0068	1000.00	285.9281	286.1118	286.6808
LP	-302.2574	0.2154	0.0068	1000.00	-302.7741	-302.2051	-302.0214
b	0.0001	0.00002	0.000	1000.00	0.00008	0.0001	0.0001

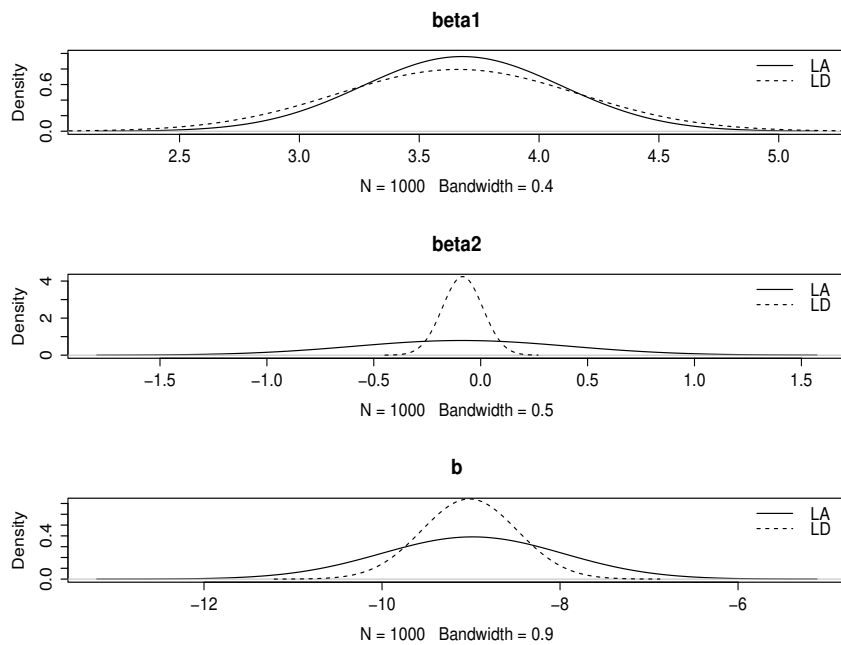


Fig. 4: Posterior density plots for Gompertz(b, θ) model, LA stands for LaplaceApproximation and LD for LaplacesDemon

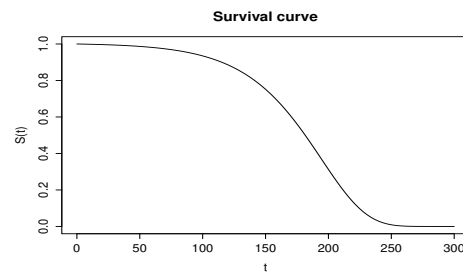


Fig. 5: Survival curve of Gompertz(b, θ) Model

5 Fitting Bayesian model in JAGS

The software JAGS is called from within R via R2jags [12, 13] to conduct the Bayesian analysis of Gompertz model for Prognosis of women breast cancer data. R2jags is designed for inference on Bayesian models using Markov chain Monte Carlo (MCMC) simulation. Running a model refers to generating samples from the posterior distribution of the model parameters. The JAGS function takes data and starting values as input. It automatically writes a jags script, calls the model, and saves the simulations for easy access in R. There are four steps for Bayesian fitting of Gompertz model in JAGS.

- Creation of data
- Model definition
- Initial values
- Model Fitting

5.1 Creation of data

```
require(R2jags)
y<-c(23, 47, 69, 70, 71, 100, 101, 148, 181, 198, 208, 212, 224, 5,
5, 10, 13, 18, 24, 26, 26, 31, 35, 40, 41, 48, 50, 59, 61, 68, 71, 78,
105, 107, 109, 113, 116, 118, 143, 154, 162, 188, 212, 217, 225)
x1<-c(rep(0, 13), rep(1, 32))
censor<-c(1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, rep(1, 18), 0, 0, 0, 0, 1, 0, 1, 1, rep(0, 6))
X<-cbind(1, x1)
J<-ncol(X)
n<-length(y)
zeros<- rep(0, n)
C<-10000
data<-list(n=n, J=J, y=y, X=X, zeros=zeros, censor=censor, C=C)
```

5.2 Model definition

For modeling the breast cancer data, the Gompertz model is used and defined as

$$y_i \sim \text{Gompertz}(b, \theta)$$

with log-link function

$$\log \theta = X\beta.$$

where, X is model matrix and β is the vector of regression coefficients. Moreover, prior probabilities for parameters are specified as

$$\beta \sim N(0, 0.001)$$

$$b \sim U(0, 100).$$

Thus, the JAGS code of the Gompertz model is

```
cat ("model{
for (i in 1:n){
zeros[i]~dpois(phi[i])
phi[i]<- -l[i]+C
l[i]<-censor[i]*log(b)+(y[i]/theta[i])+(b*theta[i]*
(1-exp(y[i]/theta[i])))+(1-censor[i])*(b*theta[i]*
(1-exp(y[i]/theta[i])))
log(theta[i])<-inprod(X[i,],beta[])
}
## Deviance
dev<-sum(l[])
Deviance<- -2*dev
## Priors
b~dunif(0,100)
for (j in 1:J){
beta[j]~dnorm(0,0.001)
}
} ", file="modelgomp.txt")
```

5.3 Initial Values

To Start the MCMC simulation, starting values for one chain is a named list, names are the parameters used in the model. Here we use two chains, and the initial values is given below

```
inits<-function() {list (b=runif(1),beta=rnorm(2)) }
```

5.4 Model Fitting

Finally, Bayesian model using jags function is fitted and results are assigned with object Mg.Jags.

```
Mg.jags<-jags(data=data,inits=inits, param=c("beta","b"),
n.chains=2, n.iter=10000,model.file="modelgomp.txt")
set.seed(245)
print(Mg.jags)
denplot(Mg.jags,parms=c("beta[1]","beta[2]"),main=c("beta1","beta2"))
denplot(Mg.jags,parms=c("beta[1]","beta[2]"),main=c("beta1","beta2"))
```

Table 5: Summary of JAGS simulations after being fitted to the Gompertz(b, θ) model for the breast cancer data

Parameter	mu.vect	sd.vect	2.5%	50% %	97.5%	Rhat	n.eff
beta[1]	3.864	0.222	3.502	3.833	4.379	1.01	150
beta[2]	-0.111	0.123	-0.399	-0.100	0.101	1.00	2000
b	0.0001	0.00006	0.00004	0.0001	0.001	1.01	110
deviance	289.986	3.431	286.213	289.032	298.973	1.00	1

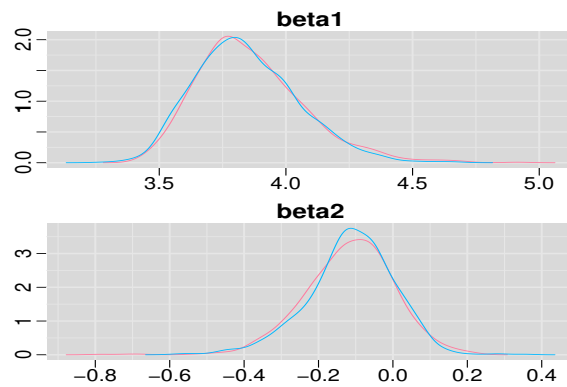


Fig. 6: Posterior density plot for β 's using the function jags

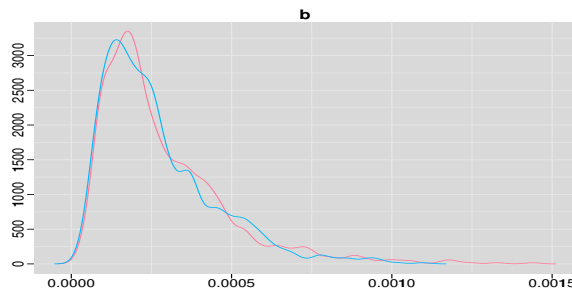


Fig. 7: Posterior density plot for b, using the function jags

6 Bayesian Analysis of Weibull Model

6.1 The Model

The likelihood function of right censored is given by

$$L = \prod_{i=1}^n [f(y_i)]^{\delta_i} [S(y_i)]^{1-\delta_i} \quad (13)$$

where $\delta_i=0$ is censored and $\delta_i=1$ is uncensored.

So, the likelihood function is given by

$$L = \prod_{i=1}^n \left[(b/\theta)(t/\theta)^{(b-1)} \exp[-(t/\theta)^b] \right]^{\delta_i} \left[\exp(-(t/\theta)^b) \right]^{1-\delta_i}$$

The joint posterior density is

$$\begin{aligned} p(b, \beta | y, X) &\propto L(y, X | b, \beta) \times p(b) \times p(\beta) \\ &\propto \prod_{i=1}^n \left[(b/\theta)(t/\theta)^{(b-1)} \exp[-(t/\theta)^b] \right]^{\delta_i} \left[\exp(-(t/\theta)^b) \right]^{1-\delta_i} \\ &\quad \times \frac{2 \times \alpha}{\pi(b^2 + \alpha^2)} \times \prod_{j=1}^J \frac{1}{\sqrt{2\pi \times 10^3}} \exp\left(-\frac{1}{2} \frac{\beta_j^2}{10^3}\right) \end{aligned} \quad (14)$$

Marginal for b

$$p(b | y, X) = \int_0^{\infty} p(b, \beta | y, X) d\beta \quad (15)$$

Marginal for β

$$p(\beta|y, X) = \int_{-\infty}^{\infty} p(b, \beta|y, X) db \quad (16)$$

6.2 Implementation using LaplacesDemon

Bayesian modeling of Weibull (b, θ) distribution and fitting model using LaplaceApproximation and LaplacesDemon functions.

6.2.1 Creation of data for LaplacesDemon

```
require(LaplacesDemon)
y<-c(23, 47, 69, 70, 71, 100, 101, 148, 181, 198, 208, 212, 224, 5,
5, 10, 13, 18, 24, 26, 26, 31, 35, 40, 41, 48, 50, 59, 61, 68, 71, 78,
105, 107, 109, 113, 116, 118, 143, 154, 162, 188, 212, 217, 225)
x1<-c(rep(0, 13), rep(1, 32))
censor<-c(1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, rep(1, 18), 0, 0, 0, 0, 1, 0, 1, 1, rep(0, 6))
X<-cbind(1, x1)
J<-2
mon.names<-c("LP", "b")
parm.names<-as.parm.names(list(beta=rep(0, J), log.b=0))
pos.beta<-grep("beta", parm.names)
pos.b<-grep("log.b", parm.names)
MyData<-list(J=J, X=X, mon.names=mon.names, parm.names=parm.names,
pos.beta=pos.beta, pos.b=pos.b, y=y, censor=censor)
```

6.2.2 Model Specification

To use LaplacesApproximation method must specify a model

$$y \sim Weibull(b, \lambda).$$

To specify a model and create a function called Model.

```
Model<-function(parm, Data)
{
beta<-parm[1:Data$J]
b<-exp(parm[Data$J+1])
beta.prior<-sum(dnorm(beta, 0, 1000, log=T))
b.prior<-dhalfcauchy(b, 25, log=T)
mu<-tcrossprod(beta, Data$X)
lambda<-exp(mu)
LL<-censor*dweibull(Data$y, b, lambda, log=TRUE) +
(1-censor)*pweibull(Data$y, b, lambda, log.p=TRUE, lower.tail=FALSE)
LL<-sum(LL)
LP<-LL+beta.prior+b.prior
Modelout<-list(LP=LP, Dev=-2*LL, Monitor=c(LP, b),
yhat=rweibull(length(y), b, lambda), parm=parm)
return(Modelout)
}
```

6.2.3 Fitting With LaplaceApproximation

```
Initial.Values <-c(coef(lm(log(y) ~ x1)), log(1))
Mw<-LaplaceApproximation(Model, Initial.Values, Data=MyData,
Iterations=8000, Method="TR")
print(Mw)
```

6.2.4 Summarizing Output

Table 6 shows the analytic results using LaplaceApproximation function while Table 7 shows the simulated results using sampling importance resampling method.

Table 6: Posterior mode, posterior sd and their quantiles

Parameter	Mode	SD	LB	UB
beta[1]	5.87	0.51	4.85	6.88
beta[2]	-1.01	0.55	-2.11	0.10
log.b	-0.08	0.17	-0.41	0.26

Table 7: Posterior mode, posterior sd and their quantiles

Parameter	Mode	SD	MCSE	ESS	LB	Median	UB
beta[1]	6.03	0.50	0.02	1000.00	5.19	6.02	7.08
beta[2]	-1.13	0.54	0.02	1000.00	-2.19	-1.13	-0.17
log.b	-0.13	0.15	0.00	1000.00	-0.38	-0.14	0.19
Deviance	316.18	2.04	0.06	1000.00	313.67	315.69	321.07
LP	-177.41	1.02	0.03	1000.00	-179.86	-177.17	-176.16
b	0.89	0.14	0.00	1000.00	0.68	0.87	1.21

6.2.5 Fitting With LaplacesDemon

Now, the data has been fitted using LaplacesDemon function for Weibull distribution using Independent Metropolis (IM) algorithm.

```
Initial.Values<-as.initial.values(Mw)
Mwld <- LaplacesDemon(Model, Data=MyData, Initial.Values,
  Covar=Mw$Covar, Iterations=1000, Status=100, Thinning=1,
  Algorithm="IM",
  Specs=list(mu=Mw$Summary1[1:length(Initial.Values),1]))
print(Mwld)
```

6.2.6 Summarizing Output

Table 8 describes simulated posterior summary.

Table 8: Posterior mean, posterior sd and their quantiles

Parameter	Mean	SD	MCSE	ESS	LB	Median	UB
beta[1]	5.89	0.21	0.01	1000.00	5.49	5.89	6.29
beta[2]	-1.02	0.23	0.01	1000.00	-1.49	-1.02	-0.56
log.b	-0.09	0.06	0.00	1000.00	-0.21	-0.09	0.03
Deviance	313.92	0.43	0.02	882.42	313.44	313.81	314.93
LP	-176.28	0.21	0.01	882.49	-176.79	-176.23	-176.04
b	0.92	0.06	0.00	1000.00	0.81	0.91	1.03

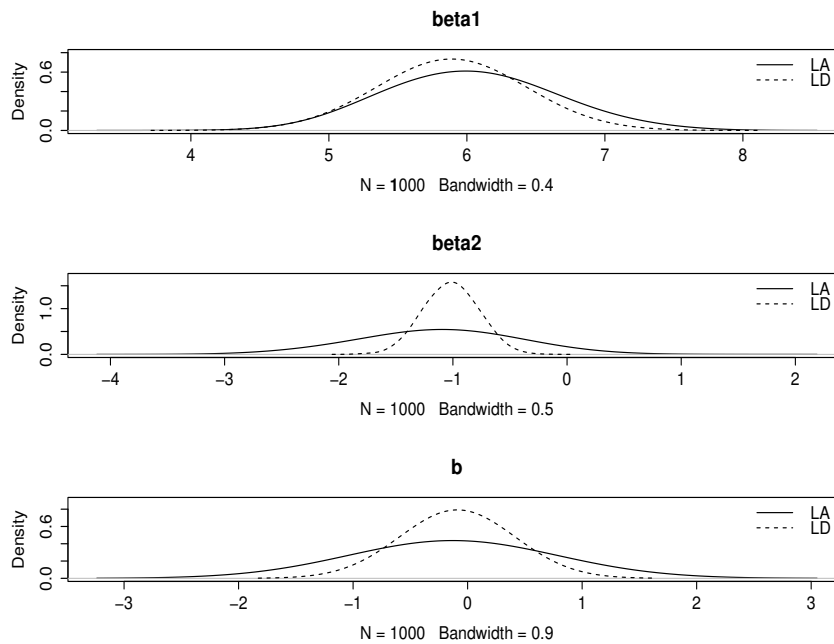


Fig. 8: Posterior density plots for Weibull(b, θ) model, LA stands for LaplaceApproximation and LD for LaplacesDemon

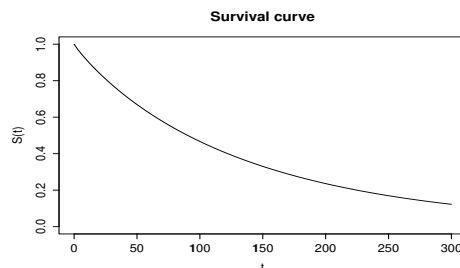


Fig. 9: Survival curve of Weibull (b, θ) Model

7 Fitting Bayesian model in JAGS

Let us consider the Bayesian analysis with JAGS using an interface of R, i.e., R2jags. For modeling the Prognosis of women with breast cancer data in JAGS, one must specify the model to run, and to load data which is created in a separate file and the initial values of the model parameters for a specified number of Markov chains.

7.1 Creation of data

```
require(R2jags)
BC<-data.frame(time=c(23, 47, 69, 70, 71, 100, 101, 148, 181, 198, 208, 212, 224, 5,
5, 10, 13, 18, 24, 26, 26, 31, 35, 40, 41, 48, 50, 59, 61, 68, 71, 78,
105, 107, 109, 113, 116, 118, 143, 154, 162, 188, 212, 217, 225),
censor=c(1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, rep(1, 18), 0, 0, 0, 0, 1, 0, 1, 1, rep(0, 6)))
y<-BC$time
x1<-c(rep(0, 13), rep(1, 32))
```

```
X<-cbind(1, x1)
J<-ncol(X)
censored=!BC$ censor
is.censored=as.numeric(censored)
y[censored]=NA
N<-length(y)
y.cens=BC$time
jags.data=list("y", "N", "is.censored", "y.cens", "J", "X")
```

7.2 Model definition

For modeling the breast cancer data, the Weibull model is used and defined as

$$y_i \sim \text{Weibull}(b, \theta)$$

with log-link function

$$\log \theta = X\beta$$

and priors

$$\beta \sim N(0, 0.0001)$$

$$b \sim U(0, 5).$$

Now, the JAGS code of the Weibull model is

```
cat("model{
for(i in 1:N){
is.censored[i]~dinterval(y[i], y.cens[i])
y[i]~dweib(b, 1/scale[i])
log(scale[i])<-inprod(X[i,], beta[])
}
b~dunif(0, 5)
for(j in 1:J){
beta[j]~dnorm(0.0, .0001)
}
}", file="modelw.txt")
```

7.3 Initial Values

To Start the MCMC simulation, it is required to specify different starting values for each chain, for the censored observations. Hence the initial values is a list of three chains

```
y.init=rep(NA, length(y))
y.init[censored]=y.cens[censored]+1
inits1=list(beta=rnorm(2), b=0.85, y=y.init)
inits2=list(beta=rnorm(2), b=0.80, y=y.init)
inits3=list(beta=rnorm(2), b=0.75, y=y.init)
inits=list(inits1, inits2, inits3)
```

7.4 Model Fitting

Finally, fitting the above specified model using jags function

```
Mw.jags=jags(data=jags.data, inits, param=c("b", "beta"), n.chains=3,
n.iter=10000, model.file="modelw.txt")
print(Mw.jags)
denplot(Mw.jags, parms=c("beta[1]", "beta[2]"), main=c("beta1", "beta2"))
denplot(Mg.jags, parm=c("b"))
```


Table 9: Summary of JAGS simulations after being fitted to the Weibull (b, θ) model for the same data

Parameter	mu.vect	sd.vect	2.5%	50% %	97.5%	Rhat	n.eff
beta[1]	5.716	0.901	4.089	5.677	7.616	1.002	2300
beta[2]	-1.021	0.507	-2.031	-1.009	-0.067	1.003	3000
b	0.961	0.155	0.685	0.950	1.279	1.012	890
deviance	283.572	6.868	271.323	283.040	298.700	1.002	1900

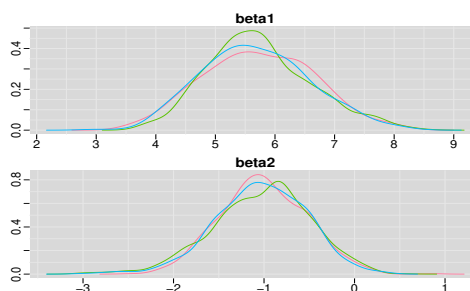


Fig. 10: Posterior density plot for β 's using the function jags

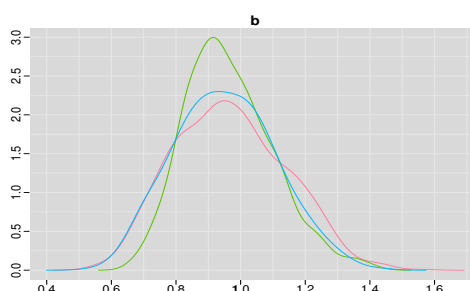


Fig. 11: Posterior density plot for b, using the function jags

8 Model Comparison

Comparison of models can be done by Deviance, Deviance information criterion (DIC) and Bayesian predictive information criterion (BPIC). Deviance is defined as -2 times the log-likelihood [2]. In Bayesian inference the lowest expected deviance has the highest posterior probability [2]. DIC is an important method for assessing model fit and comparing models. In this paper, DIC using JAGS and BPIC are used as techniques for model fitting. The goal is to select most appropriate model. BPIC is a variation of DIC where effective parameters are computed by 2pD. Comparing all the three criteria, it is seen that (see Table 10) Gompertz model has better explained the data than the Weibull model. So, Gompertz model would be most suitable for the Prognosis of women with breast cancer.

Table 10: Model Comparison for Gompertz and Weibull Model

Model	Deviance (SIR)	Deviance (IM)	Deviance	DIC	BPIC	DIC with Jags
Gompertz	287.29	286.16	289.98	286.55	288.45	293.3
Weibull	316.18	313.92	283.57	314	321.63	307.3

9 Conclusion

In this paper, Gompertz and Weibull model are used to analyze the lifetime data in Bayesian paradigm. A real survival data set is used for illustration. The Bayesian approach is implemented with R and JAGS for analytic and simulation

process. Analytic posterior summary and simulated posterior summary by Sampling importance resampling (SIR) are obtain by the function `LaplaceApproximation` in package `LaplacesDemon`. Simulated posterior summaries are produced by `LaplacesDemon` and `Jags`. On the basis of Deviance, DIC and BPIC it is clear that Gompertz model provides better results as compared to Weibull model.

Acknowledgments

The authors acknowledge with thanks to both the referees and the Editor-in-Chief for their valuable suggestions and comments which led to the overall improvement in the manuscript.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this article.

References

- [1] D. Collet, *Modelling Survival Data in Medical Research*, second edition, Chapman & Hall, London (2003).
- [2] A. Gelman, J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis*, 2nd edition, Chapman Hall, Boca Raton, FL, (2004).
- [3] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*, Third Edition, Chapman and Hall/CRC, Boca Raton, FL, (2014).
- [4] A. Gelman and J. Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models*, Cambridge University Press, New York, (2007).
- [5] J. G. Ibrahim, M. H. Chen and D. Sinha, *Bayesian Survival Analysis*, (2001).
- [6] N. Khan, M. T. Akhtar and A. A. Khan, A Bayesian Approach to Survival Analysis of Inverse Gaussian Model with Laplace Approximation, *International Journal of Statistics and Applications*, **6**, 391-398 (2016).
- [7] N. Khan, M. T. Akhtar and A. A. Khan, Bayesian Analysis of Marshall and Olkin Family of Distributions, *International Journal of Recent Scientific Research*, **8**, 18692-18699 (2017).
- [8] N. Khan and A. A. Khan, Bayesian Analysis of Topp-Leone Generalized Exponential Distribution, *Austrian Journal of Statistics*, **47**(4), 1-15 (2018).
- [9] Y. Khan and A. A. Khan Bayesian regression analysis of Weibull model with R and Bugs, *Statistical Methodologies and Applications* **2**, 131-140, (2014).
- [10] F. Mosteller and D. L. Wallace, *Inference and Disputed Authorship: The Federalist*. Reading: Addison-Wesley (1964).
- [11] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag (1999).
- [12] M. Plummer, JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling. *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, March 20–22, Vienna, Austria, ISSN 1609-395X, (2003)
- [13] M. Plummer, JAGS version 3.3.0 user manual, International Agency for Research on Cancer, Lyon, France, (2012).
- [14] R Core Team, *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>. (2015).
- [15] Statisticat LLC (2016), *LaplacesDemon: Complete environment for Bayesian inference*, R package version 3.4.3, <http://www.bayesian-inference.com/software>.
- [16] M. A. Tanner, *Tools for Statistical Inference*. Springer-Verlag, New York (1996).
- [17] L. Tierney, J. Kadane, Accurate Approximations for Posterior Moments and Marginal Densities, *Journal of the American Statistical Association*, **393**, 82-86, (1986).