Chapter 8
More Complex Models for Random Durations

When time is the random variable of interest, the simplest aleatory model is the exponential distribution, which was discussed in Chap. 3. For example, the exponential distribution was used to model fire suppression time in the guidance for fire PRA published jointly by the Nuclear Regulatory Commission and the Electric Power Research Institute (EPRI–NRC, 2005). However, there are numerous applications in which the exponential model, with its assumption of time-independent rate (of suppression, recovery, etc.) is not realistic. For example, past work described in [1] and [2] has shown that the rate of recovering offsite ac power at a commercial nuclear plant is often a decreasing function of time after power is lost. Therefore, the exponential distribution is not usually an appropriate model in this situation, and the analyst is led to more complex aleatory models that allow for time-dependent recovery rates, such as the Weibull and lognormal distributions.

Bayesian inference is more complicated when the likelihood function is other than exponential, and much past work has been done on various approximate approaches for these cases. In past PRAs, the difficulty of the Bayesian approach has led analysts to use frequentist methods instead, such as MLEs for point estimates and confidence intervals to represent uncertainties in the estimates. This was the approach adopted, for example, in both [1] and [2], and is the approach still in use for most commercial utility PRAs. Today, however, OpenBUGS can implement a fully Bayesian approach to the problem, allowing full propagation of parameter uncertainty through quantities derived from the aleatory model parameters, such as nonrecovery probabilities. Failure to consider parameter uncertainty can lead to nonconservatively low estimates of such quantities, and thus to overall risk metrics that are nonconservative. We will use an example of recovery of offsite ac power to illustrate the techniques in this chapter.

D. Kelly and C. Smith, *Bayesian Inference for Probabilistic Risk Assessment*, Springer Series in Reliability Engineering, DOI: 10.1007/978-1-84996-187-5_8,
© Springer-Verlag London Limited 2011

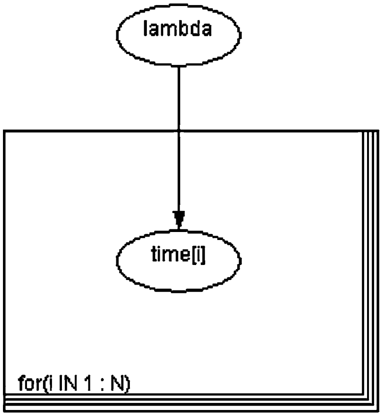

Fig. 8.1 DAG for exponential aleatory model

8.1 Illustrative Example

Consider the following times to recover offsite power following a failure of the offsite grid (a grid-related loss of offsite power (LOSP)). The units are hours: 0.1, 0.133, 0.183, 0.25, 0.3, 0.333, 0.333, 0.55, 0.667, 0.917, 1.5, 1.517, 2.083, 6.467. We will analyze these data with various alternative aleatory models that allow for a nonconstant recovery rate, and compare these models with the simpler exponential model, which has a constant recovery rate. We will use the posterior predictive checks and information criteria described in Chap. 4 and in this Chapter to select among the candidate aleatory models. We will also illustrate the impact of parameter uncertainty on a derived quantity of interest, namely the probability of not recovering the offsite grid by a critical point in time.

8.2 Analysis with Exponential Model

Recall from Chap. 3 that the simplest aleatory model for random durations is the exponential model. Recall also the assumption that the observed times constitute a random sample of size N from a common aleatory distribution. The exponential distribution has one parameter, λ, which in this case is the recovery rate of the offsite grid. This leads to the DAG shown in Fig. 8.1.

8.2.1 Frequentist Analysis

As stated above, many current PRAs have carried out a frequentist analysis for offsite power recovery. Therefore, for comparison purposes, and because the frequentist point estimates can be used as initial values in the Bayesian analysis, we present some salient frequentist results. For the exponential model, the MLE is given by

$$\hat{\lambda} = \frac{N}{\sum t_i}. \tag{8.1}$$

Confidence limits for λ are given by the following equations. In these equations, $\chi^2_\alpha(\mathrm{d})$ is the $100\times\alpha$th percentile of a chi-square distribution with d degrees of freedom.

$$\begin{aligned} \lambda_{\text{lower}} &= \frac{\chi^2_{\alpha/2}(2N)}{2\sum t_i} \\ \lambda_{\text{upper}} &= \frac{\chi^2_{1-\frac{\alpha}{2}}(2N)}{2\sum t_i} \end{aligned} \tag{8.2}$$

For our example, the MLE of λ is 0.913/h, and the 90% confidence interval for λ is (0.55, 1.35).

8.2.2 Bayesian Analysis

To allow the observed data to drive the results, we will use the Jeffreys prior for λ. Recall from Chap. 3 that this is an improper prior, proportional to $1/\lambda$, which we can think of for purposes of Bayesian updating as a gamma distribution with both parameters equal to zero. Because the gamma prior is conjugate to the exponential likelihood function, the posterior distribution is also gamma, with parameters N and Σt_i. The posterior mean is thus $N/\Sigma t_i$. Credible intervals must be found numerically, which can be done with a spreadsheet or any basic statistics package. Because the chi-square distribution is a gamma distribution with shape parameter equal to half the degrees of freedom, and scale parameter equal to ½, a 90% credible interval will be numerically identical to the 90% confidence interval calculated with Eq. 8.2.

Although it is not necessary to use OpenBUGS with this example, because of the conjugate nature of the prior and likelihood, we can do so. The script in Table 8.1 is the same as was used in Chap. 3, and implements the DAG shown in Fig. 8.1. Note the use of the barely proper gamma (0.0001, 0.0001) distribution to approximate the improper Jeffreys prior for λ.[1]

Note the specification of an initial value for λ, necessary because of the difficulty in initially sampling from the nearly improper prior distribution. Because this is a conjugate single-parameter problem, one Monte Carlo chain is sufficient.

[1] One can use a gamma (0, 0) prior in OpenBUGS, as long as initial values are loaded. However, we prefer to avoid the use of an improper prior generally, as it can lead to numerical difficulties on occasion, especially when more than one parameter is involved.

Table 8.1 OpenBUGS script for exponential aleatory model with Jeffreys prior for λ

```
model {
for(i in 1:N) {
time[i] ~ dexp(lambda) #Exponential aleatory model for recovery time
}
lambda ~ dgamma(0.0001, 0.0001) #Jeffreys prior for lambda
prob.nonrec <- exp(-lambda*time.crit)
}

data
list(time = c(0.1,0.133,0.183,0.25,0.3,0.333,0.333,0.55,
  0.667,0.917,1.5,1.517,2.083,6.467), N = 14, time.crit = 8)
inits
list(lambda = 1)
```

Using the conservative analysis criteria from Chap. 3 of 1,000 burn-in iterations, followed by 100,000 samples, we find a posterior mean for λ of 0.913/h, and a 90% credible interval of (0.55, 1.35). As expected, because we are using the Jeffreys prior and the observed random variable (time) is continuous, these are the same values, within Monte Carlo sampling error, as the frequentist estimates given above.

8.3 Analysis with Weibull Model

The Weibull distribution is an alternative aleatory model for random durations, but it has two parameters instead of one, and allows for a time-dependent recovery rate. It has a shape parameter, which we will denote as β. The second parameter, denoted α, is a scale parameter, and determines the time units. If $\beta = 1$, the Weibull distribution reduces to the exponential distribution. If $\beta > (<)\ 1$, the recovery rate is increasing (decreasing) with time. In the conventional parameterization, the Weibull density function is

$$f(t) = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1}\exp\left[-\left(\frac{t}{\alpha}\right)^{\beta}\right] \tag{8.3}$$

OpenBUGS uses a slightly different parameterization, one that has been used in the past for Bayesian inference in the Weibull distribution. In this parameterization, the scale parameter is $\lambda = \alpha^{-\beta}$. In this parameterization, the density and cumulative distribution function are given by the following equations.

$$\begin{aligned} f(t) &= \beta\lambda t^{\beta-1}\exp\left(-\lambda t^{\beta}\right) \\ F(t) &= 1 - \exp\left(-\lambda t^{\beta}\right) \end{aligned} \tag{8.4}$$

The frequentist analysis for the Weibull parameters is more complicated than for the simple exponential model. The MLEs cannot be written down in closed

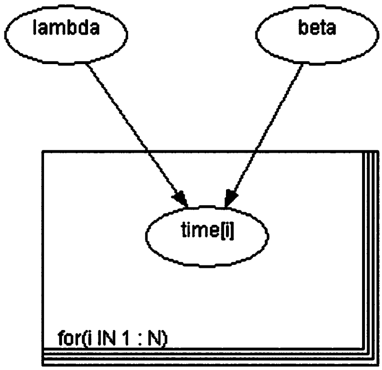

Fig. 8.2 DAG for Weibull aleatory model

form and must be found numerically, while confidence intervals are even trickier; see [3], for example. The MLEs for β and λ for our example data are 0.84 and 1.01, respectively. These will be used to aid in selecting initial values for the Markov chains in the Bayesian analysis below.

The DAG for this model is shown in Fig. 8.2. Note that in this DAG, before the times are observed, β and λ are independent, and as a result we will use independent diffuse prior distributions. Once the data are observed, β and λ become dependent, and this is reflected in the joint posterior distribution.

The OpenBUGS script used to analyze this model is shown in Table 8.2. Note that two Monte Carlo chains are used, as there are two parameters to estimate. Convergence is very quick for this model; however, a check for convergence should always be made in models with two or more parameters. We discarded the first 1,000 iterations to be conservative, and used another 100,000 iterations to estimate β and λ. The posterior means for β and λ are 0.83 and 1.01, respectively, very close to the MLEs, as expected because of the diffuse priors. The 90% credible intervals are (0.58, 1.11) for β and (0.59, 1.525) for λ. The correlation coefficient between β and λ in the joint posterior distribution is -0.32, reflecting the dependence induced by the observed data (Table 8.2).

8.4 Analysis with Lognormal Model

A lognormal aleatory model was used for recovery of offsite power in [2]. Like the Weibull model, there are two unknown parameters, commonly denoted by μ and σ. However, unlike the Weibull distribution, neither parameter determines the shape of the lognormal distribution, as the shape is fixed. The recovery rate increases initially, and then decreases monotonically. This makes the lognormal model attractive for situations where there is a mixture of easy-to-recover and hard-to-recover events. The median is determined by μ, with the median being e^{μ}. The other parameter, σ, determines the spread of the distribution. A commonly used measure of the spread is the error factor, defined as the ratio of the 95th percentile

Table 8.2 OpenBUGS script for Weibull aleatory model with independent, diffuse priors

```
model {
for(i in 1:N) {
time[i] ~ dweib(beta, lambda) #Weibull aleatory model for recovery time
}
#Independent, diffuse priors for Weibull parameters
lambda ~ dgamma(0.0001, 0.0001)
beta ~ dgamma(0.0001, 0.0001)
}

data
list(time = c(0.1,0.133,0.183,0.25,0.3,0.333,0.333,0.55,0.667,0.917,1.5,1.517,2.083,6.467),
    N = 14)
inits
Chain 1
list(beta = 0.5, lambda = 1)
Chain 2
list(beta = 1, lambda = 0.1)
```

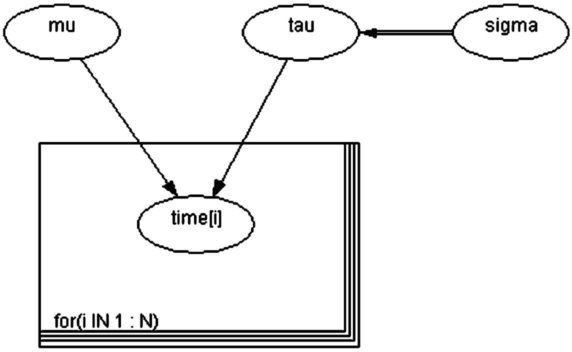

Fig. 8.3 DAG for lognormal aleatory model

to the median. The error factor is given by $e^{1.645\sigma}$. The MLEs of μ and σ are available in closed form:

$$\hat{\mu} = \frac{1}{N}\sum \log(t_i)$$
$$\hat{\sigma} = \frac{1}{N}\sum [\log(t_i) - \hat{\mu}]^2 \tag{8.5}$$

For the example data, the MLEs of μ and σ are −0.605 and 1.13, respectively.

OpenBUGS parameterizes the lognormal (and normal) distribution in terms of μ and the precision, τ, with $\tau = \sigma^{-2}$. The DAG for this model is shown in Fig. 8.3, and the accompanying OpenBUGS script is in Table 8.3. We use independent, diffuse priors for μ and σ, and let OpenBUGS infer the induced prior on τ. This is shown in the DAG by the double arrow connecting τ and σ, indicating the logical connection. Note the use of the *dflat*() prior for μ in the OpenBUGS script. This is an improper flat prior over the entire real axis, and is used because μ is not

Table 8.3 OpenBUGS script for lognormal aleatory model with independent, diffuse priors

```
model {
for(i in 1:N) {
time[i] ~ dlnorm(mu, tau) #Lognormal aleatory model for recovery time
}
#Independent, diffuse priors for lognormal parameters
mu ~ dflat()
sigma ~ dgamma(0.0001, 0.0001)
#Calculate tau
tau <- pow(sigma, -2)
}

 data
list(time = c(0.1,0.133,0.183,0.25,0.3,0.333,0.333,0.55,0.667,0.917,1.5,1.517,2.083,6.467),
   N = 14)
inits
Chain 1
list(mu = -0.5, sigma = 1)
Chain 2
list(mu = 0, sigma = 1.5)
```

restricted to positive values, as were the parameters in the Weibull and exponential models. We used the MLEs to determine initial values, and used two Monte Carlo chains to facilitate checking for convergence.

As with the Weibull model, convergence occurs very quickly. Conservatively discarding the first 1,000 iterations, and using another 100,000 iterations gives posterior means for μ and σ of $-$ 0.605 and 1.24, respectively, very close to the MLEs, as expected with diffuse priors. The 90% credible intervals are (-1.16, -0.05) for μ and (0.89, 1.73) for σ.

8.5 Analysis with Gamma Model

The gamma aleatory model is similar to the Weibull model in that it has a shape parameter, denoted by α, and a scale (really a rate) parameter, denoted by β. The density function is given by

$$f(t) = \frac{\beta^{\alpha} t^{\alpha-1} e^{-\beta t}}{\Gamma(\alpha)}$$

If $\alpha = 1$, the gamma distribution reduces to the exponential distribution. If $\alpha > (<)$ 1, the recovery rate is increasing (decreasing) with time. As with the Weibull distribution, frequentist analysis is more complicated than for the simple exponential model. The MLEs cannot be written down in closed form and must be found numerically. The MLEs for α and β for our example data are 0.85 and 0.77,

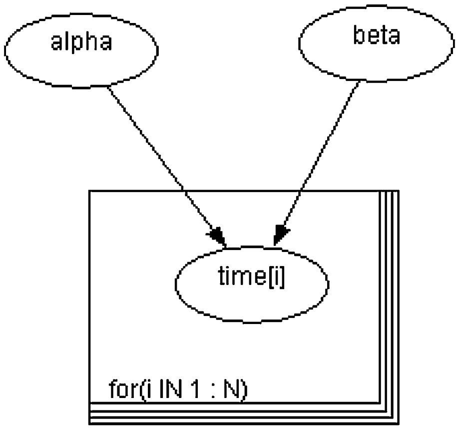

Fig. 8.4 DAG for gamma aleatory model

respectively. These will be used to aid in selecting initial values for the Markov chains in the Bayesian analysis below. The DAG for the gamma model is shown in Fig. 8.4. As before, we use independent diffuse priors for α and β. The OpenBUGS script is shown in Table 8.4.

Running this script as before we find the posterior mean of α and β to be 0.81 and 0.74, respectively, close to the MLEs, as expected with diffuse priors. The 90% intervals are (0.43, 1.30) and (0.28, 1.345).

8.6 Estimating Nonrecovery Probability

In a typical PRA, if offsite power is lost, the probability of not recovering power by a critical point in time is a quantity of prime interest. In general, the nonrecovery probability is given in terms of the aleatory cumulative distribution function by $1-F(t_{crit})$. For the exponential model, this becomes

$$P_{\mathrm{nonrec}}(t) = \exp(-\lambda t_{\mathrm{crit}}). \tag{8.6}$$

Obviously, this probability is a function of the recovery rate, λ, and because there is uncertainty in λ, this uncertainty should be propagated through this equation to give the uncertainty in the nonrecovery probability. For the Weibull model, the nonrecovery probability is a function of both β and λ, as shown by the following equation.

$$P_{\mathrm{nonrec}}(t) = \exp\left(-\lambda t_{\mathrm{crit}}^{\beta}\right) \tag{8.7}$$

For the lognormal model, the nonrecovery probability is given in terms of the cumulative standard normal distribution:

Table 8.4 OpenBUGS script for gamma aleatory model with independent, diffuse priors

```
model {
for(i in 1:N) {
time[i] ~ dgamma(alpha, beta #Gamma aleatory model for recovery time
    }
#Independent, diffuse priors for gamma parameters
alpha ~ dgamma(0.0001, 0.0001)
beta ~ dgamma(0.0001, 0.0001)
}

data
list(time = c(0.1,0.133,0.183,0.25,0.3,0.333,0.333,0.55,0.667,0.917,1.5,1.517,2.083,6.467),
    N = 14)
inits
Chain 1
list(alpha = -0.5, beta = 1)
Chain 2
list(alpha = 1, beta = 0.5)
```

$$P_{\text{nonrec}}(t) = 1 - \Phi\left(\frac{\log t_{\text{crit}} - u}{\sigma}\right) \tag{8.8}$$

For the gamma model, the cumulative distribution function cannot be written in closed form. However, even in this case, the nonrecovery probability can be estimated, with full and proper treatment of parameter uncertainty, simply by adding one additional line to the OpenBUGS script for the aleatory model under consideration. Letting `time.crit` denote the critical time of concern, the following line is added to the respective scripts:

- `prob.nonrec <– 1 – cumulative(time.rep, time.crit)`

This line instructs OpenBUGS to draw samples from the posterior distribution for the relevant parameters, and for each iteration, to calculate the nonrecovery probability, using the `cumulative()` function in OpenBUGS. This constitutes Monte Carlo evaluation of the integral of the nonrecovery probability over the posterior distribution. For example, in the case of the lognormal aleatory model, this extra line of script evaluates the following integral:

$$\int_0^{\infty}\int_{-\infty}^{\infty}\left[1 - \Phi\left(\frac{\log t_{crit} - \mu}{\sigma}\right)\right]\pi_1(\mu, \sigma|t_1, t_2, \ldots, t_n)d\mu d\sigma \tag{8.9}$$

Table 8.5 compares the point estimate nonrecovery probabilities for each of the four aleatory models we have considered with the posterior means and 90% credible intervals from the full Bayesian analysis. A critical time of eight hours was used in the calculations. Note the significance of including parameter uncertainty, as well as the influence of the aleatory model.

Table 8.5 Comparison of nonrecovery probabilities at 8 h for each aleatory model

Model	Point estimate	Posterior mean	90% interval
Exponential	6.7E-4	2.8E-3	(2.0E-5, 8.1E-4)
Weibull	2.8E-3	0.012	(5.6E-5, 0.05)
Lognormal	8.6E-3	0.022	(8.9E-4, 0.08)
Gamma	1.4E-3	8.3E-3	(3.0E-5, 0.04)

8.6.1 Propagating Uncertainty in Convolution Calculations

Consider a typical sequence cut set from a LOSP event tree:

IE-LOOP * EPS-DGN-FTS-A * EPS-DGN-FTR-B * OSP-NONREC

This cut set represents the joint occurrence of the LOSP (the initiating event, IE-LOOP), the failure of emergency diesel generator (EDG) A to start (EPS-DGN-FTS-A), the failure of EDG B to run for the required mission time (EPS-DGN-FTR-B), and the failure to recover offsite power in time to prevent core damage (OSP-NONREC). Let us focus just on the time-dependent portion of the cut set, namely EPS-DGN-FTR-B * OSP-NONREC.

To quantify basic event EPS-DGN-FTR-B, the analyst must specify a mission time for the EDG; however, this is difficult, given that the EDG only needs to run until offsite power is recovered, and the time of offsite power recovery is random. Many past PRAs have dealt with this problem by specifying a surrogate EDG mission time. However, it is becoming increasingly common to treat this problem via a convolution integral:

$$P(\text{EPS-DGN-FTR-B*OSP-NONREC}) = \int_0^{t_m} \lambda e^{-\lambda t}[1 - F(t + t_c)]dt \qquad (8.10)$$

In this equation, λ is the failure rate of EDG B, and F(t) is the cumulative distribution function of the offsite power recovery time. The mission time, t_m, is the overall mission time for the sequence, which is typically 24 h. The other variable in this equation is t_c, which is the difference between the time to core damage and the time needed to restore power to the safety buses, after the grid has been recovered.

There are a number of uncertainties in this equation, both explicit and implicit. Explicitly, there is epistemic uncertainty in the EDG failure rate, λ. There is also some uncertainty in the time to core damage and the time it takes operators to restore power to the safety buses, causing t_c to be uncertain. The aleatory model for offsite power recovery time has one or more parameters, and the epistemic uncertainty in these parameters leads to uncertainty in F. Finally, there is uncertainty associated with the choice of aleatory model for offsite power recovery. In the typical PRA treatment of this equation, none of these uncertainties are addressed; only point estimate values are used.

We will use the offsite power recovery times from our earlier example to illustrate the impacts of these uncertainties. We will assume that the uncertainty in the EDG failure rate, λ, is lognormally distributed with a mean of 10^{-3}/h and an error factor of 5. For the time to core damage, denoted t_{CD}, we will assume a best estimate time of 1.5 h, with uncertainty described by a uniform distribution between 0.75 and 2 h. For the time needed by operators to restore power to the safety buses, denoted t_0, we assume a best estimate of 0.1 h, with uncertainty described by a uniform distribution between 0.05 and 0.3 h.

To illustrate the impact that uncertainty has on this problem, we first illustrate the results using point estimates for all of the parameters. We use the posterior mean values for the recovery model parameters, the mean of the EDG failure rate, and the best estimate times, with the typical PRA mission time of 24 h. Evaluating the convolution integral in Eq. 8.10 gives a core damage probability of 2.4×10^{-4} for the exponential recovery model, and 3.7×10^{-4} for the Weibull model.

To address uncertainty, we use the OpenBUGS script shown in Table 8.6. Note the use of the integral functional in OpenBUGS to carry out the convolution. Note that the variable of integration must be denoted by "x," and x is indicated as missing (NA) in the data statement. The results are shown for each model, with and without uncertainty in t_{CD} and t_0 (Tables 8.7, 8.8).

Because of the manner in which most PRA software quantifies minimal cut sets, it is common to estimate a cut set "recovery factor," which is multiplied by the remaining events in the cut set to obtain the cut set frequency. For this particular example, the recovery factor would be given by the following equation.

$$RF = \frac{P(\text{EPS-DGN-FTR-B}^{*}\text{OSP-NONREC})}{P(\text{EPS-DGN-FTR-B})} \tag{8.11}$$

The numerator is the core damage probability from the convolution integral, and the denominator is the probability of EDG failure to run over the PRA mission time. Using the point estimate values for the convolution integral of 2.4×10^{-4} for the exponential recovery model, and 3.7×10^{-4} for the Weibull model, with a point estimate EDG failure probability of 0.024 over a 24 h mission time, we obtain point estimate recovery factors of 0.01 and 0.016 for the exponential and Weibull models, respectively.

To consider the uncertainty in the recovery factor, we included lines in the script in Table 8.6 to carry out the calculation. Tables 8.9 and 8.10 show the results. Note the considerable uncertainty expressed by the 90% credible intervals. This is the result of the recovery factor being a ratio of two uncertain quantities. The differences from the point estimates tend to increase as the number of recovery times decreases, and failure to address uncertainty in the convolution can lead to considerable nonconservatism with sparse recovery data (see Exercise 8.5).

Table 8.6 OpenBUGS script to propagate uncertainty through convolution calculation of core damage probability

```
model {
for(i in 1:N) {
# time[i] ~ dweib(beta, lambda) #Weibull aleatory model for recovery time
time[i] ~ dexp(lambda) #Exponential aleatory model for recovery time
}
#Diffuse prior distributions
#beta ~ dgamma(0.0001, 0.0001)
lambda ~ dgamma(0.0001, 0.0001)
#Specify integrand of convolution integral (uses OpenBUGS integral functional)
#Weibull aleatory model
#F(x) <- lambda.edg*exp(-lambda.edg*x)*exp(-lambda*pow(x + T.c, beta))
#Exponential aleatory model
F(x) <- lambda.edg*exp(-lambda.edg*x)*exp(-lambda*(x + T.c))
#Time available to restore offsite power to avert core damage
#T.cd is time to core damage, T.0 is time to restore power to safety loads after recovering offsite
    power
T.c <- T.cd - T.0
#No uncertainty in T.cd and T.0
#T.cd <- 1.5
#T.0 <- 0.1
#Uncertainty in T.cd and T.0
T.cd ~ dunif(0.75, 2)
T.0 ~ dunif(0.05, 0.3)
#Lognormal distribution for EDG failure rate, mean = 1.0E-3/hr, EF = 5
lambda.edg ~ dlnorm(mu, tau)
tau <- pow(sigma, -2)
sigma <- log(5)/1.645
mu <- log(1.0E-3) - pow(sigma, 2)/2
prob.nonrec <- integral(F(x),0,24,1.E-6)
rec.factor <- prob.nonrec/prob.edg
prob.edg <- 1 - exp(-lambda.edg*24)
}
data
list(time = c(0.1,0.133,0.183,0.25,0.3,0.333,0.333,0.55,0.667,0.917,1.5,1.517,2.083,6.467),
    N = 14, x = NA)
inits
list(beta = 0.5, lambda = 1)
list(beta = 1, lambda = 0.1)
list(lambda = 1)
```

8.7 Model Checking and Selection

Chapter 4 introduced the concept of using replicate draws from the posterior predictive distribution in order to check if the model (prior + likelihood) could replicate the observed data with a reasonable probability. In the current context,

Table 8.7 Results for convolution calculation of core damage probability considering uncertainty in recovery model parameters and EDG failure rate

Recovery model	Mean probability of core damage	90% credible interval
Exponential	3.6E-4	(2.3E-5, 1.3E-3)
Weibull	5.0E-4	(3.0E-5, 1.8E-3)

Table 8.8 Results for convolution calculation of core damage probability considering uncertainty in recovery model parameters, EDG failure rate, time to core damage, and power restoration time

Recovery model	Mean probability of core damage	90% credible interval
Exponential	4.4E-4	(2.6E-5, 1.6E-3)
Weibull	5.7E-4	(3.3E-5, 2.1E-3)

Table 8.9 Results for convolution calculation of cut set recovery factor considering uncertainty in recovery model parameters and EDG failure rate

Recovery model	Mean recovery factor	90% credible interval
Exponential	0.015	(3.4E-3, 3.5E-2
Weibull	0.021	(4.3E-3, 5.4E-2)

Table 8.10 Results for convolution calculation of cut set recovery factor considering uncertainty in recovery model parameters, EDG failure rate, time to core damage, and power restoration time

Recovery model	Mean recovery factor	90% credible interval
Exponential	0.018	(3.7E-3, 4.3E-2
Weibull	0.024	(4.7E-3, 5.9E-2)

we would generate a replicate time, and compare its distribution to the observed times. If one or more of the observed times falls in a tail of this predictive distribution, there may be a problem with the model.

The replicated time is generated by adding the following lines to the respective scripts:

- Exponential: `time.rep ~ dexp(lambda)`
- Weibull: `time.rep ~ dweib(beta, lambda)`
- Lognormal: `time.rep ~ dlnorm(mu, tau)`
- Gamma: `time.rep ~ dgamma(alpha, beta)`.

OpenBUGS generates parameter values from the posterior distribution, and then generates a time from the aleatory model with the sampled parameter values. In terms of an equation, the replicate times are samples from the posterior predictive distribution, which for a lognormal aleatory model, becomes

$$f(t) = \int_{0}^{\infty} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma t} \exp\left[-\frac{(\log t - u)^2}{2\sigma^2} \right] \pi_1(u, \sigma | t_1, t_2, \ldots, t_n) du d\sigma \quad (8.12)$$

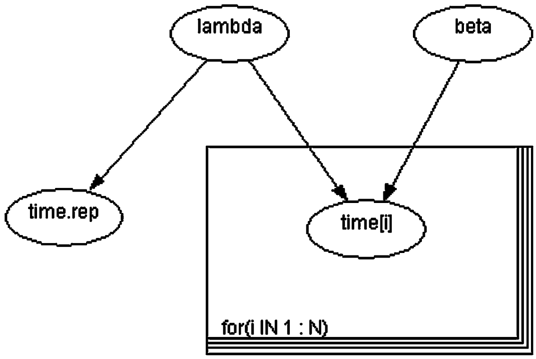

Fig. 8.5 DAG for Weibull aleatory model showing replicated time from posterior predictive distribution

Table 8.11 Summary of replicated times from posterior predictive distributions

Model	95% Credible interval
Exponential	(0.03, 4.61)
Weibull	(8.2E-3, 5.98)
Lognormal	(0.04, 7.5)
Gamma	(5.9E-3, 5.47)

In terms of a DAG, an additional node is added, as shown in Fig. 8.5 for the Weibull model.

Table 8.11 shows the 95% credible interval for the replicate time under each aleatory model. The exponential model cannot replicate the longest observed time with reasonable probability. The Weibull and gamma models do better, and the lognormal model, having the heaviest tail of the four, does the best at covering the range of observed data. However, it somewhat over-predicts the longest time.

One can also use summary statistics derived from the posterior predictive distribution. Using the Cramér-von Mises statistic, as described in [4], one can calculate a Bayesian p-value for each model, with a value near 0.5 being desirable. This statistic uses the cumulative distribution function of the ranked times, both replicated and observed. The following OpenBUGS scripts illustrate this calculation for each of the four aleatory models under consideration, using the `cumulative()` function in OpenBUGS to calculate the cumulative distribution function. The following Bayesian p-values were calculated for each of the three models: exponential, 0.42; Weibull, 0.44; lognormal, 0.58; gamma, 0.41. By this criterion, all four models are reasonable choices, with the exponential, gamma, and Weibull models tending to slightly under-predict recovery times, and the lognormal model tending to slightly over-predict (Tables 8.12, 8.13, 8.14).

It is also possible to use various information criteria, which penalize models with more parameters to protect against over-fitting. Note that these criteria are measures of relative fit, and that the best model from such a relative point of view may not be very good from an absolute point of view, so in practice it can be useful to examine replicated times and Bayesian p-value to assess model adequacy on an absolute basis. Two information criteria are discussed the Bayesian

Table 8.12 OpenBUGS script to calculate Bayesian p-value for exponential model, using Cramér-von Mises statistic from posterior predictive distribution

```
model {
for(i in 1:N) {
time[i] ~ dexp(lambda) #Exponential aleatory model for recovery time
time.rep[i] ~ dexp(lambda) #Replicate time from posterior predictive distribution
#Rank observed times and replicate times
time.ranked[i] <- ranked(time[], i)
time.rep.ranked[i] <- ranked(time.rep[], i)
#Calculate components of Cramer-von Mises statistic for observed and replicate data
F.obs[i] <- cumulative(time[i], time.ranked[i])
F.rep[i] <- cumulative(time.rep[i], time.rep.ranked[i])
diff.obs[i] <- pow(F.obs[i] - (2*i-1)/(2*N), 2)
diff.rep[i] <- pow(F.rep[i] - (2*i-1)/(2*N), 2)
}
lambda ~ dgamma(0.0001, 0.0001) #Jeffreys prior for lambda
#Calculate distribution of Cramer-von Mises statistic for observed and replicate data
CVM.obs <- sum(diff.obs[])
CVM.rep <- sum(diff.rep[])
p.value <- step(CVM.rep - CVM.obs) #Mean value should be near 0.5
}
data
list(time = c(0.1,0.133,0.183,0.25,0.3,0.333,0.333,0.55,0.667,0.917,1.5,1.517,2.083,6.467),
   N = 14)
inits
list(lambda = 1)
```

information criterion (BIC) and the deviance information criterion (DIC). BIC has been recommended for use in selecting among non-hierarchical models, while the DIC has been recommended for selecting among hierarchical models. However, in practice, we have found that a non-hierarchical model selected on the basis of BIC also tends to be the model selected on the basis of DIC (but not vice versa). Since no additional scripting is needed to calculate DIC in OpenBUGS, it is probably the easier of the two criteria to use. However, we will illustrate how to calculate BIC, also.

For BIC, we have

$$\text{BIC} = -2 \times \log \text{likelihood} + k \log N. \tag{8.13}$$

In this equation, k is the number of unknown parameters (one for the exponential model, two for the Weibull and lognormal models) and N is the number of data points (14 in our example). The model with the smallest BIC is preferred. Note that, some references swap the + and − signs in Eq. 8.13; under this alternative definition one selects the model with the *largest* BIC. The scripts below illustrate the calculation of BIC in OpenBUGS for each of the four aleatory models under consideration. In our example, the estimated BICs are 34.2, 36.99, 33.6, and 37.7 for the exponential, Weibull, lognormal, and gamma models, respectively.

Table 8.13 OpenBUGS script to calculate Bayesian p-value for Weibull model, using Cramér-von Mises statistic from posterior predictive distribution

```
model {
for(i in 1:N) {
time[i] ~ dweib(beta, lambda) #Weibull aleatory model
time.rep[i] ~ dweib(beta, lambda)
#Rank observed times and replicate times
time.ranked[i] <- ranked(time[], i)
time.rep.ranked[i] <- ranked(time.rep[], i)
#Calculate components of Cramer-von Mises statistic for observed and replicate data
F.obs[i] <- cumulative(time[i], time.ranked[i])
F.rep[i] <- cumulative(time.rep[i], time.rep.ranked[i])
diff.obs[i] <- pow(F.obs[i] - (2*i-1)/(2*N), 2)
diff.rep[i] <- pow(F.rep[i] - (2*i-1)/(2*N), 2)
}
#Diffuse prior distributions
lambda ~ dgamma(0.0001, 0.0001)
beta ~ dgamma(0.0001, 0.0001)
#Calculate distribution of Cramer-von Mises statistic for observed and replicate data
CVM.obs <- sum(diff.obs[])
CVM.rep <- sum(diff.rep[])
p.value <- step(CVM.rep - CVM.obs) #Mean value should be near 0.5
}
data
list(time = c(0.1,0.133,0.183,0.25,0.3,0.333,0.333,0.55,0.667,0.917,1.5,1.517,2.083,6.467),
   N = 14)
inits
#Weibull model (two chains)
list(beta = 0.5, lambda = 1)
list(beta = 1, lambda = 0.1)
```

So based on BIC, we would select the lognormal model over the other three (Tables 8.15, 8.16, 8.17, 8.18).

We now will calculate the DIC for each of the candidate models. The DIC is calculated automatically by OpenBUGS, with no need for additional scripting. The lognormal model also has the smallest DIC, 30.16 versus 32.57, 33.73, and 34.46 for the exponential, Weibull, and gamma models, so it would be selected under this criterion, also.

In summary, among the four candidate aleatory models for recovery time, the lognormal model would be chosen based on penalized likelihood criteria such as BIC or DIC. In terms of replicated recovery times, the exponential, gamma, and Weibull models tend to under-predict recovery times, with the exponential model being the worst in this regard. The lognormal model gives the best coverage of the observed recovery times, but tends to slightly over-predict these times, as indicated by the Bayesian p-value being slightly > 0.5. The choice of aleatory model can make a considerable difference in the estimated nonrecovery probability,

Table 8.14 OpenBUGS script to calculate Bayesian p-value for lognormal model, using Cramér-von Mises statistic from posterior predictive distribution

```
model {
for(i in 1:N) {
time[i] ~ dlnorm(mu, tau) #Lognormal aleatory model
time.rep[i] ~ dlnorm(mu, tau)
#Rank observed times and replicate times
time.ranked[i] <- ranked(time[], i)
time.rep.ranked[i] <- ranked(time.rep[], i)
#Calculate components of Cramer-von Mises statistic for observed and replicate data
F.obs[i] <- cumulative(time[i], time.ranked[i])
F.rep[i] <- cumulative(time.rep[i], time.rep.ranked[i])
diff.obs[i] <- pow(F.obs[i] - (2*i-1)/(2*N), 2)
diff.rep[i] <- pow(F.rep[i] - (2*i-1)/(2*N), 2)
}
#Diffuse prior distributions
mu ~ dflat()
sigma ~ dgamma(0.0001, 0.0001)
tau <- pow(sigma, -2)
#Calculate distribution of Cramer-von Mises statistic for observed and replicate data
CVM.obs <- sum(diff.obs[])
CVM.rep <- sum(diff.rep[])
p.value <- step(CVM.rep - CVM.obs) #Mean value should be near 0.5
}
data
list(time = c(0.1,0.133,0.183,0.25,0.3,0.333,0.333,0.55,0.667,0.917,1.5,1.517,2.083,6.467),
   N = 14)
inits
#Lognormal model (two chains)
list(mu = -0.5, sigma = 1.1)
list(mu = -1, sigma = 1.4)
```

especially when parameter uncertainty is accounted for properly in a fully Bayesian analysis. From this consideration, the exponential model gives the most nonconservative result in this example, because of its tendency to under-predict recovery times. The lognormal model is the most conservative of the four from this regard, because of its tendency to slightly over-predict recovery times.

8.8 Exercises

1. Reference [5] The following projector lamp failure times (in hours) have been collected: 387, 182, 244, 600, 627, 332, 418, 300, 798, 584, 660, 39, 274, 174, 50, 34, 1895, 158, 974, 345, 1755, 1752, 473, 81, 954, 1407, 230, 464, 380, 131, 1205.

Table 8.15 OpenBUGS script to calculate BIC for the exponential aleatory model

```
model {
for(i in 1:N) {
time[i] ~ dexp(lambda) #Exponential aleatory model for recovery time
#Exponential log-likelihood components
log.like[i] <- log(lambda) - lambda*time[i]
}
log.like.tot <- sum(log.like[])
#Calculate Bayesian information criterion
BIC <- -2*log.like.tot + log(N)
lambda ~ dgamma(0.0001, 0.0001) #Jeffreys prior for lambda
}
data
list(time = c(0.1,0.133,0.183,0.25,0.3,0.333,0.333,0.55,0.667,0.917,1.5,1.517,2.083,6.467),
   N = 14)
inits
list(lambda = 1)
```

Table 8.16 OpenBUGS script to calculate BIC for the Weibull aleatory model

```
model {
for(i in 1:N) {
time[i] ~ dweib(beta, lambda) #Weibull aleatory model for recovery time
#Weibull log-likelihood components
log.like[i] <- log(lambda) + log(beta) + (beta-1)*log(time[i]) - lambda*pow(time[i], beta)
}
log.like.tot <- sum(log.like[])
#Calculate Bayesian information criterion
BIC <- -2*log.like.tot + 2*log(N)
#Independent, diffuse priors for Weibull parameters
lambda ~ dgamma(0.0001, 0.0001)
beta ~ dgamma(0.0001, 0.0001)
}
data
list(time = c(0.1,0.133,0.183,0.25,0.3,0.333,0.333,0.55,0.667,0.917,1.5,1.517,2.083,6.467),
   N = 14)
inits
Chain 1
list(beta = 0.5, lambda = 1)
Chain 2
list(beta = 1, lambda = 0.1)
```

Table 8.17 OpenBUGS script to calculate BIC for the lognormal aleatory model

```
model {
for(i in 1:N) {
time[i] ~ dlnorm(mu, tau) #Lognormal aleatory model for recovery time
#Lognormal log-likelihood components
log.like[i] <- -0.5*(log(2) + log(3.14159)) - log(sigma) - log(time[i]) - pow(log(time[i])-mu, 2)/
    (2*pow(sigma, 2))
}
log.like.tot <- sum(log.like[])
#Calculate Bayesian information criterion
BIC <- -2*log.like.tot + log(N)*2
#Independent, diffuse priors for lognormal parameters
mu ~ dflat()
sigma ~ dgamma(0.0001, 0.0001)
#Calculate tau
tau <- pow(sigma, -2)
}
data
list(time = c(0.1,0.133,0.183,0.25,0.3,0.333,0.333,0.55,0.667,0.917,1.5,1.517,2.083,6.467),
    N = 14)
inits
Chain 1
list(mu = -0.5, sigma = 1)
Chain 2
list(mu = 0, sigma = 1.5)
```

a. Use a Weibull aleatory model for the failure time, with diffuse priors on the Weibull parameters. What is the posterior probability that the Weibull shape parameter exceeds 1? What does this suggest about the viability of the Weibull model compared with the exponential model?
b. Use DIC to compare exponential, Weibull, and lognormal aleatory failure time models for the lamp.

2. The following repair times have been observed. Assuming these are a random sample from an exponential aleatory model, update the Jeffreys prior for λ to find the posterior mean and 90% credible interval for λ. 105, 1, 1263, 72, 37, 814, 1.5, 211, 330, 7929, 296, 1, 120, 1.
3. Answer the questions below for the following times: 1.8, 0.34, 0.23, 0.55, 3, 1.1, 0.68, 0.45, 0.59, 4.5, 0.56, 1.6, 1.8, 0.42, 1.6.

a. Assuming an exponential aleatory model, and a lognormal prior for λ with a mean of 1 and error factor of 3, find the posterior mean and 90% interval for λ.
b. Assume a lognormal aleatory model with independent, diffuse priors on the lognormal parameters. Find the posterior mean and 90% interval for λ.

Table 8.18 OpenBUGS script to calculate BIC for the gamma aleatory model

```
model {
for(i in 1:N) {
time[i] ~ dgamma(alpha, beta) #Gamma aleatory model for recovery time
#Gamma log-likelihood components
log.like[i] <-alpha*log(beta) + (alpha-1)*log(time[i]) - beta*time[i] - loggam(alpha)
}
log.like.tot <- sum(log.like[])
#Calculate Bayeian information criterion
BIC <- -2*log.like.tot + 2*log(N)
#Independent, diffuse priors for Weibull parameters
alpha ~ dgamma(0.0001, 0.0001)
beta ~ dgamma(0.0001, 0.0001)
}
data
list(time = c(0.1,0.133,0.183,0.25,0.3,0.333,0.333,0.55,0.667,0.917,1.5,1.517,2.083,6.467),
   N = 14)
inits
Chain 1
list(alpha = 0.5, beta = 1)
Chain 2
list(alpha = 1, beta = 0.5)
```

4. The following are emergency diesel generator (EDG) repair times, in hours. 50, 4, 2, 56, 4.7, 3.7, 33, 1.1, 3.1, 13, 2.6, 12, 11, 51, 17, 28, 20, 49.
   a. Find the posterior means of the parameters of a Weibull aleatory model for these repair times. Use independent, diffuse priors for the Weibull parameters.
   b. Generate two, independent *predicted* repair times from the posterior distribution. What is the mean and 90% credible interval for each of these times?
   c. Define a new variable that is the *minimum* of each of the two times generated in part (b). What is the mean and 90% credible interval for this minimum time?
5. From past experience, we judge that the exponential distribution is not likely to be a reasonable model for duration of LOSP. This is because the recovery rate tends to decrease with time after the event. We decide to use an alternative model that allows for a decreasing recovery rate. The following 10 recovery times are observed, in hours: 46.7, 1581, 33.7, 317.1, 288.3, 2.38, 102.9, 113, 751.6, 26.4.

a. Use OpenBUGS to fit a Weibull distribution to these times. Use diffuse, independent hyperpriors for the two Weibull parameters. Find the posterior mean of the shape parameter. What is the probability that the recovery rate is decreasing with time?
b. Now use OpenBUGS to fit a lognormal model to the same times. Again, use diffuse, independent hyperpriors for the two lognormal parameters.
c. Use graphical and quantitative posterior predictive checks to decide if one of these models fits significantly better than the other.
d. Compare DIC for these models with that of a simpler exponential model.

6. Nine failure times were observed for a heat exchanger in a gasoline refinery (in year): 0.41, 0.58, 0.75, 0.83, 1.00, 1.08, 1.17, 1.25, 1.35. The analyst proposes a Weibull(β, λ) aleatory model for these failure times. The analyst believes that $\beta = 3.5$ and manufacturer reliability specifications were translated into a uniform (0.5, 1.5) prior distribution for λ. Find the posterior mean and 90% credible interval for λ. Are there any problems with replicating the observed data with this model?
7. In Ex. 6, find a 90% posterior interval for the MTTF of the heat exchanger.

References

1. Atwood CL et al (1997) Evaluation of Loss of Offsite Power Events at Nuclear Power Plants: 1980–1996, U. S. Nucl Regulatory Comm, NUREG/CR-5496
2. Eide S et al (2005) Reevaluation of station blackout risk at nuclear power plants, U. S. Nuclear Regulatory Commission, NUREG/CR-6890
3. Bain L, Engelhardt M (1991) Statistical theory of reliability and life-testing models. Marcel-Dekker, NY
4. Kelly DL, Smith CL (2009) Bayesian inference in probabilistic risk assessment—the current state of the art. Reliab Eng Syst Saf 94:628–643
5. Hamada MS, Wilson AG, Reese CS, Martz HF (2008) Bayesian reliability. Springer, New York