UNIVERSIDAD CARLOS III DE MADRID working papers

Working Paper 03-31
Statistics and Econometrics Series 07
July 2003

Departamento de Estadística y Econometría
Universidad Carlos III de Madrid
Calle Madrid, 126
28903 Getafe (Spain)
Fax (34-91) 624-9849

A BAYESIAN ANALYSIS OF BETA TESTING *

Michael Wiper[1] & Simon Wilson[1,2]

Abstract

In this article, we define a model for fault detection during the beta testing phase of a software design project. Given sampled data, we illustrate how to estimate the failure rate and the number of faults in the software using Bayesian statistical methods with various different prior distributions. Secondly, given a suitable cost function, we also show how to optimise the duration of a further test period for each one of the prior distribution structures considered.

Keywords: Software reliability, Beta testing, Test period optimisation, Bayesian inference.

*** Acknowledgements**

Michael Wiper acknowledges assistance from the Spanish Ministry of Science and Technology via the project BEC2000-0167.

[1] Departamento de Estadística y Econometría, Universidad Carlos III de Madrid
[2] Department of Statistics, Trinity College Dublin

A Bayesian Analysis of Beta Testing

Michael P. Wiper & Simon P. Wilson
Departamento de Estadística y Econometría. Universidad Carlos III de Madrid.

July 18, 2003

Abstract

In this article, we define a model for fault detection during the beta testing phase of a software design project. Given sampled data, we illustrate how to estimate the failure rate and the number of faults in the software using Bayesian statistical methods with various different prior distributions. Secondly, given a suitable cost function, we also show how to optimize the duration of a further test period for each one of the prior distribution structures considered.

Keywords: Software reliability, Beta testing, Test period optimization, Bayesian inference.

1 Introduction

Software reliability models attempt to describe the process of fault occurrence and detection in software. Such models have been applied to different aspects of the fault process using a variety of different probability structures. A summary of the different modeling strategies is to be found in Singpurwalla and Wilson (1999).

One particular, commonly used, fault detection process is "beta testing". Beta testing is usually carried out by a software producer when the in-house (alpha) testing phase has been completed. Then, the software is given to a number of users, who use the software under real conditions and report the occurrence of failures. These users do not have access to the software code and therefore cannot try to look for or correct the cause of a failure. Furthermore, various users will often observe multiple failures caused by the same fault in the software. Beta testing is useful to software manufacturers because it can be a rapid way to detect faults (there is the possibility to have many testers) and is usually cheap (the testers may be ordinary users of the software and not professional testers, for example). However, beta testing is only a fault detection process and still leaves faults to be corrected. A second disadvantage is that not all observed failures will necessarily be reported by the testers.

In this paper we develop a model for the process of beta testing. The goal is to use this model to propose an optimal beta testing strategy in terms of knowledge about the software, and the costs and benefits of the testing. We use the ideas of decision theory and Bayesian statistics to achieve this.

The paper is organized as follows. Firstly, in the following section, we introduce a simple mathematical model to represent the beta testing process. Then in section 3, we consider the possibilities of Bayesian inference, introducing three different prior distribution structures for the unknown model parameters and in section 4, we show how posterior distributions can be calculated given the different priors. In section 5, we introduce a cost function to represent the costs of introducing a second test period and we show how the choice of test period can be optimized. We illustrate our procedure with simulated and real examples in section 6 and we finish with some conclusions and extensions in section 7.

2 A Model for the Beta Testing Process

We wish to test a software program that initially contains an unknown number of faults N. The faults are labeled $1, \ldots, N$ and it is presumed that, when the program is run by a single tester, the time to observe fault k is denoted S_k and is exponentially distributed independently of the other faults, as $S_k|\lambda_k \sim \exp(\lambda_k)$. Thus, the time, T, to observe the first fault is given by $T = \min\{S_1, \ldots, S_N\}$ and therefore,

$$T|\boldsymbol{\lambda} \sim \exp(\lambda_0), \tag{1}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots \lambda_N)$ and $\lambda_0 = \lambda_1 + \ldots + \lambda_N$. Furthermore, if Z is an indicator of which fault caused the first failure, then we have

$$P(Z = i|\boldsymbol{\lambda}) = \frac{\lambda_i}{\lambda_0} \quad \text{for } i = 1, \ldots, N$$

and it can be shown that Z and T are statistically independent.

Now assume that there are M_0 testers who each test the software during a fixed time period of length T_0, where the time to observe the first failure for each tester follows the distribution given in (1). Usually, whenever a failure is observed by a tester then the fault causing that failure can be identified and re-labeled if necessary. Occasionally however, it may also occur that when a failure is observed, its cause is not identified. This leads to the existence of missing data. An extreme case of this is in a system in general use such as Netscape. Here, when a user encounters a failure, they have the option of sending an email detailing what they were doing when the failure occurred which, presumably, allows the fault causing the failure to be identified by the Netscape designers. However, the proportion of users who actually send such emails is very low and thus, there are many observed failures where the cause of the failure cannot be ascertained.

Here, we will assume that the chance that the cause of a given failure is identified is equal to some value p for each tester and each failure, independently of the failure times. For more general approaches to the missing data problem, see e.g. Little and Rubin (1987).

Since beta testing is just a fault discovery process, faults are not corrected until the end of the test period and therefore multiple failures caused by the same faults can be observed by various testers. Because of the assumption of exponentiality, the distribution of the number of faults discovered in the test period will be that of a single tester running the program during a time $M_0 T_0$.

Suppose now that tester i observes n_i failures in the test period with inter-failure times $t_{i1}, \ldots, t_{in_i}$ and the labels of the faults causing these failures are $z_{i1}, \ldots, z_{in_i}$, for $i = 1, \ldots, M_0$ where, formally we define $z_{ij} = 0$ if the fault causing failure j for tester i is not identified.

The likelihood function in this case is given by

$$\begin{aligned} l(p, N, \boldsymbol{\lambda}|\text{data}) &= \prod_{i=1}^{M_0} \left(\prod_{j=0}^{n_i} p^{I(z_{ij})} (1-p)^{1-I(z_{ij})} \frac{\lambda_{z_{ij}}}{\lambda_0} \lambda_0 e^{-\lambda_0 t_{ij}} \right) e^{-\lambda_0 \left(T_0 - \sum_{j=1}^{n_i} t_{ij}\right)} \\ &= p^{r-r_0}(1-p)^{r_0} \prod_{k=0}^{K} \lambda_k^{r_k} \exp\left(-M_0 T_0 \lambda_0\right) \quad \text{for } N \geq K, \qquad (2) \end{aligned}$$

where the indicator $I(z_{ij}) = 1$ if $z_{ij} > 0$ and $I(z_{ij}) = 0$ if $z_{ij} = 0$. Also, $r_k = \sum_{i=1}^{M_0} \sum_{j=1}^{n_i} I_k(z_{ij})$ is the total number of observed failures idenitified as caused by fault k, r_0 is the total number of unidentified failures, $r = r_0 + \ldots + r_K$ and K is the number of distinct bugs that have been discovered. It is assumed here that the discovered faults have been labeled as 1 through K.

Although the rates λ_k of the identified faults can be estimated classically, by maximum likelihood for example, those of any faults not yet identified cannot be estimated. Furthermore, the MLE for N is equal to the number of identified faults and thus the estimated rate of the unidentified faults is zero. Therefore, MLE does not provide a reasonable solution in this case. An alternative, discussed in the following section is to use Bayesian methods.

3 Bayesian inference

Bayesian methods are appealing because they allow us to use prior knowledge in order to produce more reasonable inferences about the model parameters. Firstly, we may well be able to estimate the proportion of identified failures p on the basis of our past experience with earlier projects. Secondly, by assuming that fault rates λ_k are exchangeable, a reasonable assumption since it merely implies that the prior distribution of the rates is invariant under a permutation of their labels, we can estimate the sum of rates of unobserved faults. Furthermore, we should typically have fairly good prior information about N; experts will have typically have worked on previous projects, have seen prior versions of the program etc. and informative covariates such as software metrics may be available. Several methods have been proposed in the literature for specifying prior distributions for the number of faults in a program given expert judgements, e.g. Campodónico and Singpurwalla (1994), or given software metrics, e.g. Rodríguez Bernal and Wiper (2001).

Thinking about the rates of the individual faults will be difficult, especially as they are not identified a priori. However it will be possible to estimate the overall rate λ_0 which can be interpreted as the mean number of failures in unit time. This suggests reparameterizing the problem in terms of p, N, λ_0 and the normalised rates $\rho_k = \lambda_k/\lambda_0$, for $k = 1, \ldots, N$. Under this formulation, the likelihood function becomes:

$$l(p, N, \lambda_0, \boldsymbol{\rho}|\text{data}) = p^{r-r_0}(1-p)^{r_0}\lambda_0^r \exp\left(-M_0 T_0 \lambda_0\right)\left(\prod_{k=1}^{N} \rho_k^{r_k}\right),$$

where $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_N)$ and $0 < p < 1$, $N \geq K$, $\lambda_0 \geq 0$, $0 \leq \rho_k \leq 1$ and $\sum_{k=1}^{N} \rho_k = 1$.

There are several prior structures for $(p, N, \lambda_0, \boldsymbol{\rho})$ that can be considered. First, we will assume that p, N and λ_0 are independent a priori, and that $\boldsymbol{\rho}$ depends only on N, thus

$$P(p, N, \lambda_0, \boldsymbol{\rho}) = P(p)\, P(N)\, P(\lambda_0)\, P(\boldsymbol{\rho} \,|\, N).$$

As, a priori we are assuming that the individual faults are unidentified, the distribution $P(\boldsymbol{\rho} \,|\, N)$ is required to be symmetric.

For convenience, we now choose the following straightforward prior distribution models for p, N and λ_0 where we assume that the prior parameters can be derived from the relevant prior information:

$$\begin{aligned} p &\sim \text{Beta}(v,w) \\ N &\sim \text{Poisson}(\theta) \\ \lambda_0 &\sim \text{Gamma}(a,b) \end{aligned}$$

This leaves us with the problem of formulating a symmetric, exchangeable prior distribution for $\boldsymbol{\rho}$ given N. We consider 3 possible structures for $\boldsymbol{\rho}$ which reflect different levels of prior knowledge:

$\mathcal{F}$ A fixed, deterministic structure: $\rho_1 = \ldots = \rho_N = \frac{1}{N}$. Under this model, we are assuming that all faults are of the same size, in the same spirit as the model of Jelinski and Moranda (1973).

$\mathcal{D}$ A Dirichlet prior $\boldsymbol{\rho}|\mathcal{D}, N \sim \text{Dirichlet}(\phi,\ldots,\phi)$ for some fixed value $\phi > 0$. For example, $\phi = 1$ gives a uniform distribution. Under this prior, we have $E(\rho_k \,|\, N) = \frac{1}{N}$, but there is some uncertainty allowed, with larger values of ϕ implying less variance in the ρ_k.

$\mathcal{H}$ A hierarchical prior distribution, with the parameters of the Dirichlet distribution in model 3 allowed to differ. We suggest

$$\begin{aligned} \boldsymbol{\rho}|\mathcal{H}, N, \boldsymbol{\phi} &\sim \text{Dirichlet}(\phi_1,\ldots,\phi_N) \quad \text{where } \boldsymbol{\phi} = (\phi_1,\ldots,\phi_N) \\ \phi_i|\psi &\sim \text{Exponential}(\psi) \quad \text{for } i = 1,\ldots,N \\ \psi &\sim \text{Gamma}(\alpha,\beta) \end{aligned}$$

We still have that the unconditional mean $E(\rho_k \,|\, N) = 1/N$. In this case it would even be possible to use an improper prior distribution for ψ such as a uniform distribution or $f(\psi) \propto \frac{1}{\psi}$.

Given the observed test data and the prior distributions defined here we are able to calculate the posterior distributions as illustrated in Section 4.

4 Posterior distributions

Under all 3 models for $\boldsymbol{\rho}$, p and λ_0 are independent of the other model parameters *a posteriori* and have beta and gamma distributions respectively

$$\begin{aligned} p|\text{data} &\sim \text{Beta}(v + r - r_0, w + r_0) \\ \lambda_0|\text{data} &\sim \text{Gamma}(a + r, b + M_0 T_0) \end{aligned} \tag{3}$$

with mean $E[\lambda_0|\text{data}] = \frac{a+r}{b+M_0T_0}$.

The posterior distributions of N and $\boldsymbol{\rho}$, and $\boldsymbol{\phi}$ and ψ in the case of model 3, are described below.

4.1 Fixed Structure

Under this prior the ρ_k are fixed and the only unknown parameters are N and λ_0. We have $P(p, N, \lambda_0 \,|\, \mathcal{F}, \text{data}) = P(p|\text{data})\, P(N \,|\, \mathcal{F}, \text{data})\, P(\lambda_0 \,|\, \text{data})$, where $P(p|\text{data})$ and $P(\lambda_0 \,|\, \text{data})$ are given in (3) and

$$P(N \,|\, \mathcal{F}, \text{data}) \propto \left(\frac{1}{N}\right)^{r-r_0} \frac{\theta^N}{N!}$$

for $N \geq K$. The constant of proportionality is the sum of the terms on the right hand side over all valid values of N and may be easily approximated numerically.

4.2 Dirichlet Structure

The parameters are now $(p, N, \lambda_0, \boldsymbol{\rho})$ and the posterior has the form

$$P(p, N, \lambda_0, \boldsymbol{\rho} \,|\, \mathcal{D}, \text{data}) = P(p|\text{data})\, P(\lambda_0 \,|\, \text{data})\, P(N \,|\, \mathcal{D}, \text{data}) P(\boldsymbol{\rho} \,|\, \mathcal{D},\, N, \text{data}),$$

where

$$\begin{aligned} P(N \,|\, \mathcal{D}, \text{data}) &\propto \frac{\Gamma(N\phi)}{\Gamma(N\phi + r - r_0)} \frac{\theta^N}{N!}, \quad N \geq K, \\ \boldsymbol{\rho} \,|\, \mathcal{D},\, N, \text{data} &\sim \text{Dirichlet}(\phi + r_1, \ldots, \phi + r_N), \end{aligned}$$

with the constant of proportionality in $P(N \,|\, \mathcal{D}, \text{data})$ approximated easily by summation.

4.3 Hierarchical Prior

In this case, we cannot calculate the marginal posterior distribution of N (or the other model parameters) by straightforward methods. Instead, we can consider the use of a simulation scheme to sample from the posterior parameter distribution. One possibility is to use a reversible jump, Markov chain Monte Carlo (RJMCMC) sampler. See Green (1995) for details.

For a reversible jump sampler, conditional on N, a Gibbs sampler (see e.g. Smith and Gelfand 1990) is used to sample from the joint posterior distribution of the remaining model parameters ρ, ϕ, ψ. The Gibbs sampler proceeds by sequentially sampling from the conditional posterior distributions:

$$\begin{array}{rcl}
\boldsymbol{\rho}|\mathcal{H},\, N, \boldsymbol{\phi}, \psi, \text{data} & \sim & \text{Dirichlet}(\phi_1 + r_1, \ldots, \phi_N + r_N) \\
\psi|\mathcal{H},\, N, \boldsymbol{\rho}, \boldsymbol{\phi}, \text{data} & \sim & \text{Gamma}\left(\alpha + N, \beta + \sum_{i=1}^{N} \phi_i\right) \\
f(\boldsymbol{\phi}|\mathcal{H},\, N, \boldsymbol{\rho}, \psi, \text{data}) & \propto & \exp\left(-\psi \sum_{i=1}^{N} \phi_i\right) \Gamma\left(\sum_{i=1}^{N} \phi_i\right) \prod_{i=1}^{N} \frac{\rho_i^{\phi_i}}{\Gamma(\phi_i)}.
\end{array}$$

Note that the distributions of ψ and $\boldsymbol{\rho}$ are straightforward to sample directly. The distribution of $\boldsymbol{\phi}$ can be sampled using, for example, the Metropolis algorithm (Metropolis et al 1953).

In order to sample the posterior distribution of N, a reversible jump proposal is used. Thus, after each cycle of the sampler, we make a random selection between either a birth move generating an extra unobserved fault ($N \to N+1$) or a death move eliminating one of the unobserved faults ($N \to N-1$). In the case of a birth move, we generate the parameters ϕ_{N+1} and ρ_{N+1} from some pre-specified distributions and in the case of a death move, the parameters ϕ_N and ρ_N are "killed". In each case, $\boldsymbol{\rho}$ is rescaled so that its components sum to 1. The proposed move is accepted or rejected with a probability that can be calculated using the methods of Green (1995).

Thus, the algorithm proceeds as follows:

```
1. $j = 0$, $N = N^{(0)}$, $\boldsymbol{\phi} = \boldsymbol{\phi}^{(0)}, \psi = \psi^{(0)}$.

2. Sample $\boldsymbol{\rho}^{(j+1)} \sim$ Dirichlet$(\phi_1^{(j)} + r_1, \ldots, \phi_N^{(j)} + r_N^{(j)})$.

3. Sample $\psi^{(j+1)} \sim$ Gamma$\left(\alpha + N, \beta + \sum_{i=1}^{N^{(j)}} \phi_i^{(j)}\right)$.

4. Sample $\boldsymbol{\phi}^{(j+1)} \sim f\left(\boldsymbol{\phi}|\mathcal{H},\, N^{(j)}, \boldsymbol{\rho}^{(j+1)}, \psi^{(j+1)}\right)$.

5. Sample $N^{(j+1)}$ using the reversible jump sampler.  Redefine the
   values of $\boldsymbol{\rho}^{(j+1)}$ and $\boldsymbol{\phi}^{(j+1)}$.

6. $j = j + 1$.

7. Go to 2.
```

Under suitable conditions, it can be shown that the sampled data converge to a random sample from the joint posterior distribution. Thus, for example the posterior mean of N can simply be estimated by the sample mean;

$$E[N|\mathcal{H}, \text{data}] \approx \frac{1}{J}\sum_{j=1}^{J} N^{(j)}$$

where we have assumed that the sampler has been run for some time (to forget the dependence on the initial values) and then a sample of size J has been taken, for some sufficiently large value J.

For further details of MCMC and Gibbs sampling see e.g. Robert and Casella (2000).

4.4 Comparing the 3 models

So far we have not considered how to compare the adequacy of the three structures considered in the light of the observed data. In reliability problems with sequential failure times the usual technique is to use the prequential likelihood ratio, see e.g. Dawid (1984). Here, as the time order of the failures isirrelevant, this is equivalent to computing the Bayes factor, see e.g. Jeffreys (1961) or Kass and Raftery (1995).

The Bayes factor in favour of model $\mathcal{M}_1$ against model $\mathcal{M}_2$ is given by

$$B(\mathcal{M}_1, \mathcal{M}_2) = \frac{P(\text{data}|\mathcal{M}_1)}{P(\text{data}|\mathcal{M}_2)}$$

i.e. the ratio of integrated likelihoods under the two models. Both Jeffreys (1961) and Kass and Raftery (1995) give tables of values for the Bayes factor which may be interpreted as providing evidence in favour of one of the models.

In this case, it is possible to evaluate $P(\text{data}|\mathcal{F})$ and $P(\text{data}|\mathcal{D})$ directly and we can show that

$$B(\mathcal{F}, \mathcal{D}) = \frac{\sum_{N=k}^{\infty} \frac{\theta^N}{N!} \frac{1}{N^{r-r_0}}}{\sum_{N=k}^{\infty} \frac{\theta^N}{N!} \frac{\Gamma(N\phi)\prod_{j=1}^{k}\Gamma(\phi+r_j)}{\Gamma(N\phi+r-r_0)\Gamma(\phi)^{N-k}}}.$$

We can also estimate $P(\text{data}|\mathcal{H})$ from the data sampled from the reversible jump sampler by applying methods developed in Chib (1995) and Chib and Jeliazkov (2001). Thus, it is possible to estimate Bayes factors to compare all 3 models.

5 A cost function for further testing

In this section we apply the models to establish the optimal testing strategy given past data. The problem is to determine for how long software should be tested, and by how many testers, given data and information on the costs and benefits of such testing. The possibility that no more testing should be done is also a solution. The general goal is to achieve a balance between undertesting, when buggy software will be released and user confidence will be lost, and overtesting which will be expensive and overly time consuming. A reference to the strategies that might be used and the costs involved in testing is Singpurwalla and Wilson (1999, Chapter 6).

One informal approach would be to look at the predictive distribution of the time to next failure of the program after the faults detected in the first phase of testing have been removed. In our case, conditional on the model parameters, we have

$$P(T \geq t | N, \lambda_0, \boldsymbol{\rho}, \text{data}) = \exp\left(-\lambda_0 \sum_{i=K+1}^{N} \rho_i\right)$$

where T is the time to next failure. Given the different models, the predictive reliability function of T can be derived. For example, given the fixed, deterministic structure, we have

$$P(T \geq t | \mathcal{F}, \text{data}) = \sum_{N=K}^{\infty} P(N|\mathcal{F}, \text{data}) \left(\frac{b + M_0 T_0}{b + M_0 T_0 + (1 - K/N)t}\right)^{a+r} \quad (4)$$

The predictive reliability functions can similarly be estimated for both the Dirichlet and hierarchical models. Note also that there exists a finite probability that the program is fault free after the first phase of beta testing and thus the reliability function does not converge to zero as $t \to \infty$.

A more formal Bayesian approach is to specify a cost function that represents the costs and benefits of the testing process as a function of the decision variables and unknown quantities such as the number of bugs discovered and undiscovered. The optimal decision is those values of the decision variables that minimize the expected cost, expectation being taken with respect to the unknown quantities; see e.g. French (1988).

In our case, we consider the simple case of single stage testing; the software is to be tested for a further time T_1 by M_1 testers after which any faults found are to be corrected and the software is to be released immediately. We shall also assume that the probability that a failure is identified in this second test period is given by p as earlier.

The idea now is to optimize the values of T_1 and M_1 with respect to the costs involved. We shall assume the following costs.

1. a cost c_1 per tester per unit time. This reflects the cost of paying and supporting a tester. Typically, in the beta testing situation, this cost is likely to be small as testers are not usually paid directly, although the costs of supplying them with the software and responding to their comments and queries must be considered.

2. a cost c_2 for each new fault discovered during testing. This reflects the cost of correcting the discovered bugs at the end of the test period.

3. a cost c_3 per unit time. This reflects the lost opportunity cost of delaying release of the software.

4. a cost c_4 per failure per unit time after the software is released. We would generally set this cost to be much higher than the previous values as the damage caused by leaving high frequency faults in the program will be important.

This implies that the overall cost function is

$$\mathcal{C}(M_1, T_1) = c_1 M_1 T_1 + c_2 B + c_3 T_1 + c_4 \lambda_0 \left(\sum_{i=K+1}^{N} \rho_i (1 - I(i)) \right) \qquad (5)$$

where B is the number of distinct faults found in the testing phase and $I(i)$ is an indicator of whether bug i is found or not in the testing phase.

In the following subsection we note how to evaluate the expected cost function given the three different distributions for $\boldsymbol{\rho}$.

5.1 Evaluating the expected cost function

First we should note that the number of bugs found in the second test phase can be expressed as $B = \sum_{i=k+1}^{N} I(i)$, i.e. the sum of the indicators of whether or not bug i is found. Thus, the cost function in equation 5 can be written as

$$\mathcal{C}(M_1, T_1) = c_1 M_1 T_1 + c_3 T_1 + c_4 \lambda_0 \sum_{i=K+1}^{N} \rho_i + \sum_{i=K+1}^{N} (c_2 - c_4 \lambda_0 \rho_i) I(i).$$

Now suppose initially that we know the values of p, $N > K$, λ_0 and $\boldsymbol{\rho}$. Then taking expectations we have:

$$
\begin{aligned}
E[C(M_1,T_1)|p,N,\lambda_0,\boldsymbol{\rho},\text{data}] &= c_1M_1T_1+c_3T_1+c_4\lambda_0\sum_{i=K+1}^{N}\rho_i + \\
&\qquad \sum_{i=K+1}^{N}(c_2-c_4\lambda_0\rho_i)E[I(i)|p,N,\lambda_0,\boldsymbol{\rho},\text{data}]. \qquad (6)
\end{aligned}
$$

Now, let F be the number of failures that we observe in the second test period. Then,

$$
F|p,N,\lambda_0,\boldsymbol{\rho},\text{data} \sim \text{Poisson}\left(M_1T_1\lambda_0\sum_{j=K+1}^{N}\rho_j\right)
$$

and, we can condition on F to find $E[I(i)|p,N,\lambda_0,\boldsymbol{\rho},\text{data}]$.

$$
\begin{aligned}
E[I(i)|p,N,\lambda_0,\boldsymbol{\rho},\text{data}] &= E[E[I(i)|F,p,N,\lambda_0,\boldsymbol{\rho},\text{data}]] \\
&= \sum_{f=0}^{\infty}P(F=f|p,N,\lambda_0,\boldsymbol{\rho},\text{data})E[I(i)|F=f,p,N,\lambda_0,\boldsymbol{\rho},\text{data}]
\end{aligned}
$$

and now we have that

$$
E[I(i)|F=f,p,N,\lambda_0,\boldsymbol{\rho},\text{data}] = 1-\left(1-p+p\frac{\sum_{j=K+1,j\neq i}^{N}\rho_j}{\sum_{j=K+1}^{N}\rho_j}\right)^f \quad \text{for } N\geq K+1.
$$

Therefore we find that

$$
\begin{aligned}
E[I(i)|p,N,\lambda_0,\boldsymbol{\rho},\text{data}] &= \sum_{f=1}^{\infty}\frac{(M_1T_1\lambda_0\sum_{j=K+1}^{N}\rho_j)^f e^{-\left(M_1T_1\lambda_0\sum_{j=K+1}^{N}\rho_j\right)}}{f!} \\
&\quad \times\left[1-\left(1-p+p\frac{\sum_{j=K+1,j\neq i}^{N}\rho_j}{\sum_{j=K+1}^{N}\rho_j}\right)^f\right].
\end{aligned}
$$

Inserting this formula in equation (4) we have the expected cost given the model parameters. In order to calculate the unconditional expected cost $E[C(M_1,T_1)|\text{data}]$ we now need to integrate out (4) with respect to the model parameters p, N, λ_0, and $\boldsymbol{\rho}$. As λ_0 is independent of the remaining model parameters and has the same posterior distribution in all three structures considered, we can integrate it out immediately to give:

$$\begin{aligned}
E[C(M_1,T_1)|p,N,\boldsymbol{\rho},\text{data}] &= \int_0^\infty E[C(M_1,T_1)|p,N,\lambda_0,\boldsymbol{\rho},\text{data}]f(\lambda_0|\text{data})\ d\lambda_0 \\
&= c_1M_1T_1 + c_3T_1 + c_4\frac{a+r}{b+M_0T_0}\sum_{i=K+1}^{N}\rho_i + \\
&\quad \sum_{i=K+1}^{N}\sum_{f=1}^{\infty}\left[1-\left(1-p+p\frac{\sum_{j=K+1,j\neq i}^{N}\rho_j}{\sum_{j=K+1}^{N}\rho_j}\right)^f\right]\frac{\Gamma(a+r+f)}{f!\Gamma(a+r)} \\
&\quad \times\frac{(b+M_0T_0)^{a+r}(M_1T_1\sum_{j=K+1}^{N}\rho_j)^f}{(b+M_0T_0+M_1T_1\sum_{j=K+1}^{N}\rho_j)^{a+r+f}}\times \\
&\quad \left\{c_2-c_4\rho_i\frac{a+r+f}{b+M_0T_0+M_1T_1\sum_{j=K+1}^{N}\rho_j}\right\}
\end{aligned}$$

We can also integrate out p. Thus

$$\begin{aligned}
E[C(M_1,T_1)|N,\boldsymbol{\rho},\text{data}] &= \int_0^\infty E[C(M_1,T_1)|p,N,\boldsymbol{\rho},\text{data}]\ dp \\
&= c_1M_1T_1 + c_3T_1 + c_4\frac{a+r}{b+M_0T_0}\sum_{i=K+1}^{N}\rho_i + \\
&\quad \sum_{i=K+1}^{N}\sum_{f=1}^{\infty}\left[1-\sum_{s=0}^{f}\left(\begin{array}{c} f \\ s \end{array}\right)\frac{B(v+r-r_0+s,w+r_0+f-s)}{B(v+r-r_0,w+r_0)}\right. \\
&\quad \left.\left(\frac{\sum_{j=K+1,j\neq i}^{N}\rho_j}{\sum_{j=K+1}^{N}\rho_j}\right)^s\right]\frac{\Gamma(a+r+f)}{f!\Gamma(a+r)} \\
&\quad \times\frac{(b+M_0T_0)^{a+r}(M_1T_1\sum_{j=K+1}^{N}\rho_j)^f}{(b+M_0T_0+M_1T_1\sum_{j=K+1}^{N}\rho_j)^{a+r+f}}\times \\
&\quad \left\{c_2-c_4\rho_i\frac{a+r+f}{b+M_0T_0+M_1T_1\sum_{j=K+1}^{N}\rho_j}\right\} \qquad (7)
\end{aligned}$$

where $B(x,y)$ is the beta function. This formula can now be resolved for each of the three prior structures on N and $\boldsymbol{\rho}$ that have been considered.

5.1.1 Fixed, deterministic model

In this case we have $\rho_i = 1/N$ and thus, from formula 7 we have:

$$E[C(M_1,T_1)|\mathcal{F},\, \text{data}] = \sum_{N=K}^{\infty} P(N|\mathcal{F},\, \text{data}) E[C(M_1,T_1)|\mathcal{F},\, N, \text{data}]$$

where $P(N|\mathcal{F},\, \text{data})$ is the posterior density for N derived in Subsection 4.1. Thus, we have:

$$\begin{aligned} E[C(M_1,T_1)|\mathcal{F},\, \text{data}] &= c_1 M_1 T_1 + c_3 T_1 + c_4 \frac{a+r}{b+M_0T_0} \sum_{N=K+1}^{\infty} \frac{N-K}{N} P(N|\mathcal{F},\, \text{data}) \\ &\quad + \sum_{N=K+1}^{\infty} (N-K) P(N|\mathcal{F},\, \text{data}) \sum_{f=1}^{\infty} \Bigg[1 - \\ &\quad \sum_{s=0}^{f} \binom{f}{s} \frac{B(v+r-r_0+s, w+r_0+f-s)}{B(v+r-r_0, w+r_0)} \left(\frac{N-K-1}{N-K} \right)^s \Bigg] \\ &\quad \frac{\Gamma(a+r+f)}{f!\Gamma(a+r)} p_N^f (1-p_N)^{a+r} \left(c_2 - c_4 \frac{a+r+f}{N(b+M_0T_0+M_1T_1\frac{N-K}{N})} \right) \end{aligned}$$

where $p_N = \frac{M_1T_1\frac{N-K}{N}}{b+M_0T_0+M_1T_1\frac{N-K}{N}}$.

This function is now straightforward to approximate numerically by truncating the summations at sufficiently large values of N and f. Furthermore, conditional on M_1 it can be shown that the function is either strictly increasing or has a unique minimum in T_1 and this minimum can thus be found by well known methods such as finding using Newton-Raphson to find the zero of the derivative $\frac{dE[C(M_1,T_1)|\mathcal{F},\text{data}]}{dT_1}$.

5.1.2 Dirichlet distribution

The expected cost can be calculated in a similar manner to the previous case. In this case it can be shown that we have:

$$\begin{aligned} E[C(M_1, T_1)|\mathcal{D}, \text{data}] \quad = \quad & c_1 M_1 T_1 + c_3 T_1 + c_4 \frac{a+r}{b + M_0 T_0} \sum_{N=K+1}^{\infty} \frac{(N-K)\phi}{N\phi + r - r_0} P(N|\mathcal{D}, \text{data}) + \\ & \sum_{N=K+1}^{\infty} (N-K) P(N|\mathcal{D}, \text{data}) \sum_{f=1}^{\infty} \frac{\Gamma(a+r+f)}{f!\Gamma(a+r)} (M_1 T_1)^f (b + M_0 T_0)^{a+r} \times \\ & \left\{ c_2 g_1(f) - c_4 \frac{(a+r+f)\phi}{N\phi + r - r_0} g_2(f) \right\} \end{aligned}$$

where $P(N|\mathcal{D}, \text{data})$ is the posterior distribution for N given in Subsection 4.2 and

$$\begin{aligned} g_1(f) \quad = \quad & \frac{B((N-K)\phi + f, K\phi + r - r_0)}{B((N-K)\phi, k\phi + r - r_0)} \Bigg[1 - \sum_{s=0}^{f} \left(\begin{array}{c} f \\ s \end{array} \right) \frac{B(v + r - r_0 + s, w + r_0 + f - s)}{B(v + r - r_0, w + r_0)} \\ & \frac{B((N-K)\phi, s)}{B((N-K-1)\phi, s)} I_{s=0}^{N \geq K+2} \Bigg] \int_0^1 \frac{h(x|(N-K)\phi + f, K\phi + r - r_0)}{(b + M_0 T_0 + M_1 T_1 x)^{(a+r+f)}} \, dx \\ g_2(f) \quad = \quad & \frac{B((N-K)\phi + f + 1, K\phi + r - r_0)}{B((N-K)\phi + 1, K\phi + r - r_0)} \Bigg[1 - \sum_{s=0}^{f} \left(\begin{array}{c} f \\ s \end{array} \right) \frac{B(v + r - r_0 + s, w + r_0 + f - s)}{B(v + r - r_0, w + r_0)} \\ & \frac{B((N-K)\phi + 1, s)}{B(N-K-1)\phi, s)} I_{s=0}^{N \geq K+2} \Bigg] \int_0^1 \frac{h(x|(N-K)\phi + f + 1, K\phi + r - r_0)}{(b + M_0 T_0 + M_1 T_1 x)^{(a+r+f+1)}} \, dx \end{aligned}$$

Here, $I_{s=0}^{N \geq K+2}$ is an indicator function taking the value 1 if $N \geq K+2$ or $s = 0$ and zero otherwise. Also,

$$h(x|\psi_1, \psi_2) = \frac{1}{B(\psi_1, \psi_2)} x^{\psi_1 - 1} (1-x)^{\psi_2 - 1} \quad \text{is a beta density function.}$$

Although this expression appears somewhat daunting, its numerical evaluation is straightforward as only one dimensional integrals are needed and, as earlier, the function has a unique minimum.

5.1.3 Hierarchical distribution

In this case we can estimate the expected cost function for given values of M_1 and T_1 by averaging the formula (7) over the data sampled from the posterior parameter distribution of p, N, $\boldsymbol{\rho}$ and $\boldsymbol{\phi}$.

As earlier, this function has a unique minimum for given M_1 and the optimum values of M_1 and T_1 can be encountered by, for example, a brute search method over a range of possible values.

6 Examples

In this section, we illustrate our procedure with two examples: one simulated and one real.

6.1 Simulated Example

Here we first generated assumed that a program contained a total of $N = 20$ faults with overall failure rate $\lambda_0 = 10$. Then, the vector of relative sizes of each fault $\boldsymbol{\rho}$ was generated from a Dirichlet distribution with parameter vector $(1, \ldots, 1)^T$. Thus, the true model here is the Dirichlet structure with $\phi = 1$.

The software was then assumed to be tested (by a single tester) for 20 time units. For every observed failure, it was assumed that the tester had a 90% probability of identifying the cause.

During the test period, failures identified as caused by 14 distinct faults were observed. The true fault sizes λ_i and number of times fault i was detected in testing r_i are given in Table 1. Note that, corresponding to $i = 0$, there were 15 unidentified failures observed during testing out of a total of $r = 199$ failures.

Table 1: True fault sizes and numbers of detections in testing.

i	λ_i	r_i	i	λ_i	r_i
0	10.000	15	11	.416	10
1	1.600	26	12	.241	1
2	1.149	20	13	.208	5
3	1.030	23	14	.204	3
4	.985	23	15	.208	0
5	.901	16	16	.053	0
6	.664	12	17	.052	0
7	.604	12	18	.051	0
8	.551	14	19	.043	0
9	.541	10	20	.035	0
10	.464	9			

Under all three model structures, the prior expected number of faults was set to $\theta = 20$, i.e. the correct value, and Jeffreys priors were used for p

Table 2: Posterior distributions of the numbers of faults in the software under the three models.

	Model		
N	$\mathcal{F}$	$\mathcal{D}$	$\mathcal{H}$
14	1.0000	.2836	.8794
15	0	.2241	.1096
16	0	.1676	.0101
17	0	.1189	.0008
18	0	.0804	.0001
19	0	.0518	.0000
20	0	.0320	.0000
> 20	0	.0416	0
$E[N\|\text{data}]$	14.0000	16.0256	14.1784
$E\left[\sum_{i\geq 15}\lambda_i\|\text{data}\right]$	0.0000	0.110	0.064

(i.e. Beta(0.5,0.5)) and λ_0 (i.e. $f(\lambda_0) \propto 1/\lambda_0$). This last corresponds to the limit of the Gamma prior distribution with a, $b \to 0$.

Under the Dirichlet model, we set $\phi = 0.1$ which is slightly different from the true value and under the hierarchical model, we set used a Gamma($\alpha = 1, \beta = 0.1$) prior for ψ. In this case, the sampler was run for 10000 iterations to burn in and 100000 in equilibrium.

Under all three models, the posterior distribution for λ_0 is Gamma(199,20) with mean $E[\lambda_0|\text{data}] = 9.95$ and similarly, the posterior mean estimate of p is $E[p|\text{data}] = 0.9225$. In both cases, as we should expect, these values are close to the true values.

In Table 2 we illustrate the posterior probabilities of different numbers of faults N and the posterior mean estimates of N for the three models. Also, we indicate the total expected rate of the unobserved faults $E\left[\sum_{i\geq 15}\lambda_i|\text{data}\right]$.

We can see here that the fixed model puts a probability of almost 1 on all faults having been detected in testing. Under the remaining models there is more uncertainty although in all cases the true number of faults remaining has been underestimated. (This is to be expected as we can see that fault number 15 with a relatively large rate (.208) was not observed in the testing period. Note that a 95% highest posterior density interval for N under the

Dirichlet structure is $[14, 20]$ which does include the true value.

Bayes factors were also calculated in order to compare the models as outlined in subsection 4.4. Perhaps surprisingly these showed found slight evidence in favour of the fixed model ($\log B(\mathcal{F}, D) \approx 3$) with the hierarchical model being much less probable ($\log B(\mathcal{F}, D) > 10$). We note however that the integrated likelihood for the Dirichlet model is sensitive to the election of the parameter ϕ and although altering this parameter slightly does not much change the predictions of the model, it does strongly alter the integrated likelihood. Thus, setting $\phi = 0.2$ we find $\log B(\mathcal{F}, D) \approx -2$ giving slight evidence in favour of the Dirichlet model and setting $\phi = 1$, i.e. the true value, we have $\log B(\mathcal{F}, D) \approx -9$ which is overwhelming evidence.

We also carried out some further sensitivity analysis on the prior distribution of N by varying the prior mean between 15 and 30. The results for the fixed model were essentially unchanged but there was some sensitivity for the other models. For example, in the case of the Dirichlet model, the posterior mean value of N varied between 15.2 and 19.5 and the 95% highest posterior density interval varied between $[14, 18]$ given a prior mean of 15 and $[14, 27]$ given a prior mean of 30.

We now assume that the observed 14 faults were corrected and we consider the problem of whether or not to undertake further testing. Firstly, in Figure 1 we plot the reliability functions for the three models as in equation 4.

We note some differences in the three reliability functions. Firstly, the predicted reliability for the fixed model is virtually equal to 1, because it is predicted that there are no faults remaining in the software with probability almost 1. Under the hierarchical model, the reliability function converges to approximately 0.9 by time 100 and under the Dirichlet model, the reliability function is somewhat lower as time increases.

Now consider the cost function for further testing. The maximum duration of the further testing period is 40 time units and we are able to use up to three testers. The loss function parameters are $c_1 = 1$, $c_2 = 0.1$, $c_3 = 0.1$ and $c_4 = 100000$. In Figure 1, we illustrate the expected cost functions for each of the three models and the different possible numbers of testers.

Firstly, under the fixed model, as the probability that there are no faults left in the software is approximately equal to 1, the expected cost function becomes an approximately linear function of test time for any number of testers. Thus, given this model it is optimal not to test. For both Dirichlet and hierarchical models, the structure of the expected cost function is somewhat different. The predicted expected cost at a given time and for a given number of testers is always higher under the Dirichlet model than

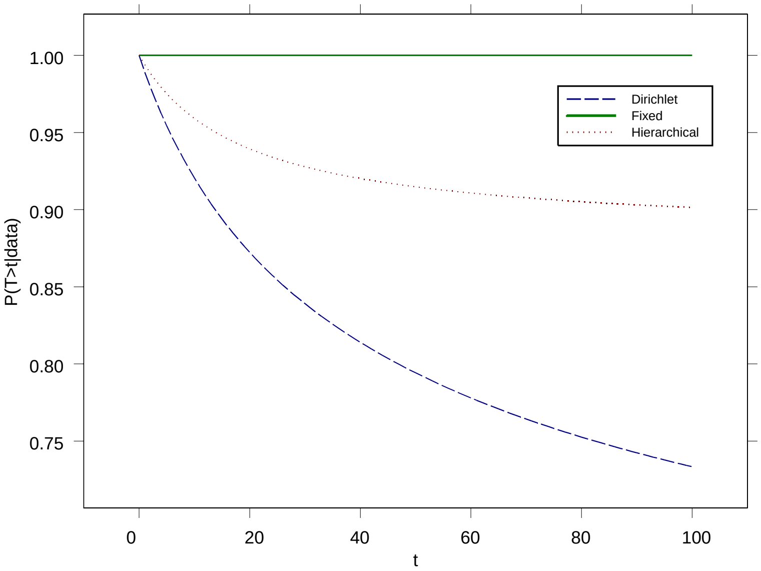

Figure 1: Predicted reliability functions.

under the fixed model. This is to be expected as under the first model it is predicted that more bugs are left in the software. In both cases it is optimal to use 3 testers with an optimal test time of around 19.5 units assuming the hierarchical model and 39.6 units assuming the Dirichlet model.

6.2 Web Log Data Example

In this case, we consider the analysis of web-log errors from a web server. In accessing a web server, various types of "failures" are possible, the most typical being the *404 error* when a user tries to access a file that does not exist; usually via a broken link. Clearly it is in a servers interest to try to get any faults within the web site and external broken links corrected.

The usage of a web site is recorded on a web log file and failures are included in an error log file. These results can be summarized using a web analysis tool such as *Analog* (http://www.analog.cx/) or *AwStats* (http://awstats.sourceforge.net/).

We considered a period of one week during which, on average, there were

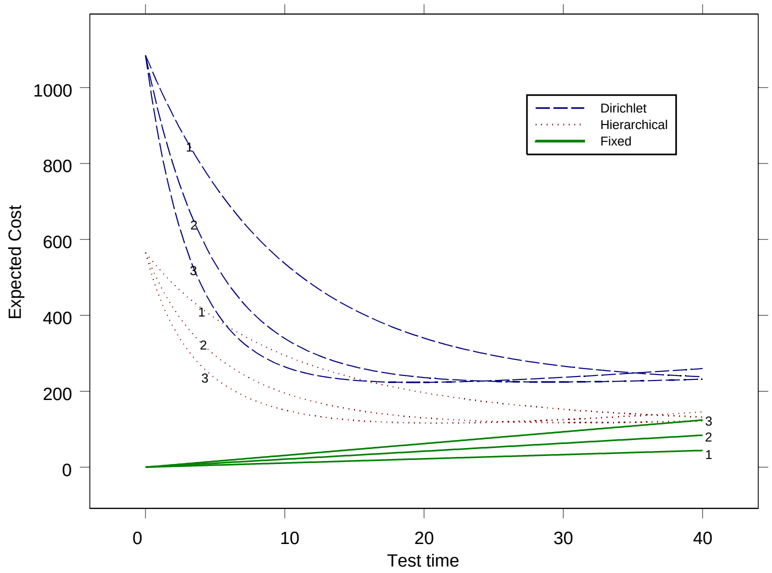

Figure 2: Expected cost functions for further testing.

324 users accessing the site. Failure data was recorded using a web analysis tool. During the period, 40 faults were identified, with the most prevalent being accessed just over 5000 times and the least prevalent being accessed 51 times. In other words, there were over 5000 attempted accesses to a non existent web file, from a number of broken links. The total number of failures was over twenty thousand and there were around 5700 unidentified failures. A graph showing number of failures associated with each fault is given in Figure 3. We can see that there were a few very large faults and many more smaller faults.

Here the prior distribution for N was set to be Poisson with mean 60, and the remaining prior distributions were set to be as in the previous example. In this case, both the fixed and hierarchical models predict that there are no faults remaining in the software with probabilities greater than 0.999. Furthermore, calculation of Bayes factors gives very strong evidence that the Dirichlet model best fits the data (log Bayes factors greater than 10). Therefore, we now consider only this model.

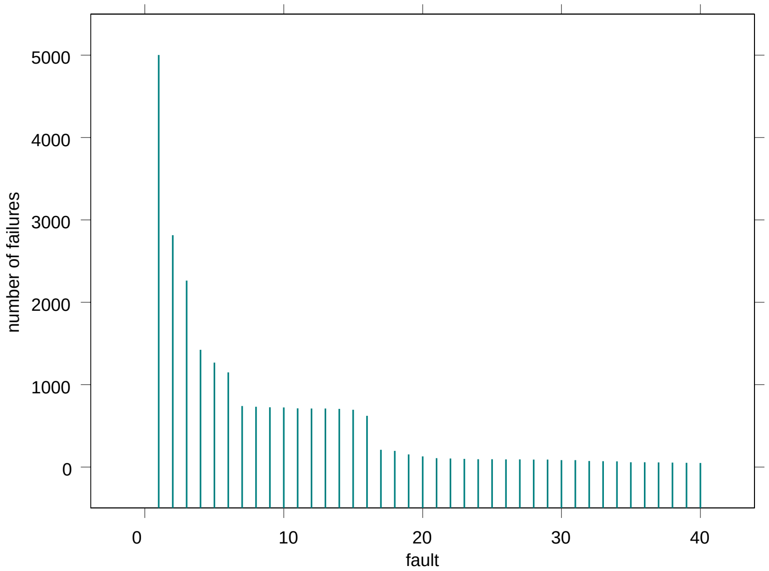

Figure 3: Numbers of observed failures associated with each fault.

Firstly, in Figure 4, we illustrate the posterior distribution of the numbers of faults in the program under the Dirichlet prior structure.

We can see that there is some uncertainty about the initial number of faults in the program. In fact, a 95% maximum posterior density interval is given by $[40, 45)$. Thus, it seems reasonable to consider the possibility of further testing.

Firstly, we should note although the time for further testing, T_1 can be controlled, the number of testers M_1 should here be considered as an exogeneous variable, as the number of users accessing the web page cannot be controlled a priori. Furthermore, we will typically not have to pay the testers so we should assume $c_1 = 0$ in the cost function.

One possibility would be, before the first testing phase, to place a prior distribution on the number of testers accessing the site per hour and then update this distribution given the information of the number of users accessing the site during the first week, assuming that the usage pattern will be the same during later periods. However, in doing this, the evaluation of

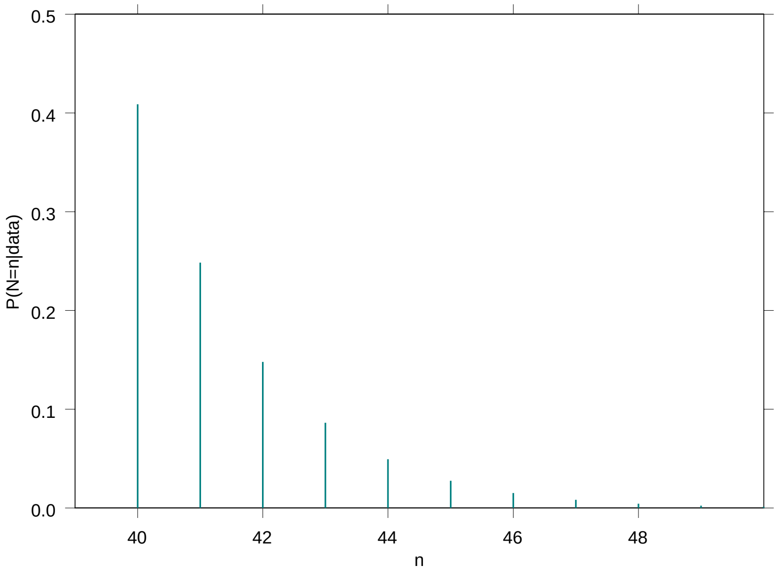

Figure 4: Posterior probabilities of different numbers of faults.

the expected cost function would be somewhat more complicated. A simpler approach is to use the observed mean value of 324 users in the first week to give us an empirical estimate for M_1 and then to look at how the expected cost function varies when we alter this value slightly.

Here we consider a possible further test period of up to 168 hours (1 week) with cost function parameters $c_1 = 0$, $c_2 = 0.1$, $c_3 = 0.01$ and $c_4 = 1000000$.

In Figure 5 we examine the expected cost function for values of M_1 between 300 and 350.

Given this cost function, the results appear to be fairly insensitive to small changes in the number of testers. Thus, the optimal test time varies between 57 hours (expected cost 2.94 units) assuming $M_1 = 300$ and 65.5 hours (cost 2.84) if $M_1 = 350$. The optimal testing time assuming $M_1 = 324$ is 62 hours with expected cost 2.89 units.

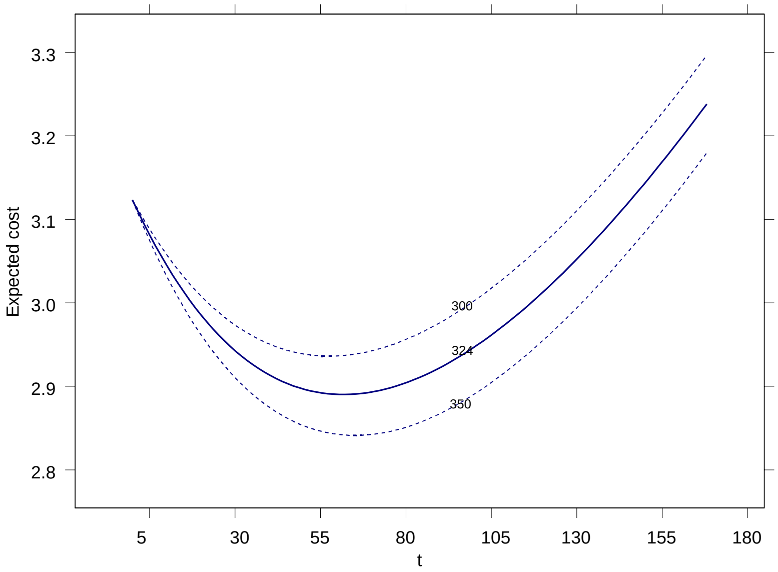

Figure 5: Expected cost function for $M_1 = 300$, 324 and 350.

7 Discussion

In this article, we have illustrated Bayesian inference for a simple model for the beta testing procedure given three different prior distribution structures and have illustrated how to decide whether or not testing should be continued using by estimating the optimal test time given a certain cost function.

There are various possible modifications and extensions that could be considered. Firstly, as we have noted, it is quite important obtain reasonable prior information about the true number of faults in the program N. Firstly, the Poisson form we have used here has been chosen for convenience and other parametric prior distributions could be considered, e.g. a negative binomial model. More importantly, in real problems expert or covariate information is often available and it would be interesting to develop methods to incorporate this information via methods developed in e.g. Campodónico and Singpurwalla (1994) and Rodríguez Bernal and Wiper (2001).

Secondly, as we have three possible model structures, rather than using Bayes factors to choose a model, we could also consider the use of model averaging. It would be possible to implement such an approach via reversible jump MCMC methods (Green 1995).

A restriction of our approach is that we have assumed that the different testers behave in the same way, so that the failure rate of a fault is independent of the tester using the software. It is possible in practice that different testers will show different patterns of usage or *operational profiles.* Thus, the failure rate of a fault may vary from tester to tester. It would be possible to extend our approach to this case, adding further hierarchical structure to our models although we would then need to use MCMC methods to implement the modeling.

Acknowledgements

Michael Wiper acknowledges assistance from the Spanish Ministry of Science and Technology via the project BEC2000-0167.

References

[1] Campodónico, S. and Singpurwalla, N.D. A Bayesian Analysis of the Logarithmic–Poisson Execution Time Model Based on Expert Opinion and Failure Data. *IEEE Transactions on Software Engineering 20*, (1994), 677–683.

[2] Chib, S. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association 90*, (1995), 1313–1321.

[3] Chib, S. and Jeliazkov I. Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association 96*, (2001), 270–281.

[4] Dawid, A.P. The Prequential Approach. *Journal of the Royal Statistical Society, A147*, (1984), 278–292.

[5] French, S. *Decision Theory: An Introduction to the Mathematics of Rationality.* Wiley, Chichester, 1988.

[6] Gelfand, A.E. and Smith, A.F.M. Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association 85*, (1990), 398–409.

[7] Green, P. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika 82*, (1995), 711–732.

[8] Jeffreys, H. *Theory of Probability*, (3rd edition). Clarendon Press, Oxford, 1961.

[9] Jelinski, Z. and Moranda, P. Software Reliability Research. In *Statistical Computer Performance Evaluation*, W. Freiburger, Ed. Academic Press, New York, 1972, pp. 465–484.

[10] Kass, R. and Raftery, A.E. Bayes factors. *Journal of the American Statistical Association 90*, (1995), 773–795.

[11] Little, R. and Rubin, D. *Statistical Analysis with Missing Data.* John Wiley, New York, 1987.

[12] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. Equations of state calculations by fast computing machines. *Journal of Chemical Physics 21*, (1953), 1087–1091.

[13] Robert, C. and Casella, G. *Monte Carlo Statistical Methods.* Springer Verlag, Berlin, 2000.

[14] Rodríguez Bernal, M.T. and Wiper, M.P. Bayesian Inference for a Software Reliability Model Using Metrics Information. In *Safety and Reliability: Towards a Safer World* (2001), E. Zio, M. Demichela and N. Piccinini, Eds., Politecnico de Torino, pp. 1999–2006.

[15] Singpurwalla, N.D. and Wilson, S. *Statistical Methods in Software Engineering: Reliability and Risk.* Springer, New York, 1999.