$\mathcal{I}^2\mathcal{SDS}$

The Institute for Integrating Statistics in Decision Sciences

Technical Report TR-2010-10

May 8, 2010

A Bayesian Hidden Markov Model for Imperfect Debugging

Antonio Pievatolo
CNR IMATI, Italy

Fabrizio Ruggeri
CNR IMATI, Italy

Refik Soyer
Department of Decision Sciences
The George Washington University

A Bayesian Hidden Markov Model for Imperfect Debugging

Antonio Pievatolo
CNR IMATI, I-20133, Milano, Italy

Fabrizio Ruggeri
CNR IMATI, I-20133, Milano, Italy

Refik Soyer
Department of Decision Sciences
The George Washington University, Washington, DC, 20052

September 20, 2010

Abstract

In this paper we present a new model to describe software failures from a debugging process. Our model allows for the imperfect debugging scenario by considering potential introduction of new bugs to the software during the development phase. Since the introduction of bugs is an unobservable process, latent variables are introduced to incorporate this property via a hidden Markov model. We develop a Bayesian analysis of the model and discuss its extensions. We also consider how to infer the unknown number of states of the hidden Markov model. The model and the Bayesian analysis are implemented to actual software failure data.

Keywords: Software reliability; failure times; Bayes factor; model selection

1 Introduction

Many papers have been published on software reliability since the original works of Jelinski and Moranda (1972) and Musa and Okumoto (1984); see Singpurwalla and Wilson (1999). Bayesian methods have been widely used in this field as discussed in the recent review by Wiper (2007).

Possibility of imperfect debugging and introduction of new bugs during software testing have been considered in earlier papers starting with Kremer (1983) who proposed a birth-death process for the number of bugs in the software at a given time. Gaudoin, Lavergne and Soler

(1994) considered failures at times $T_1 < \ldots < T_n$ and modelled the interfailure times with independent exponential distributions. In particular, they took

$$T_i - T_{i-1} \sim \mathcal{E}(\lambda_i), i = 1, \ldots, n.$$

with

$$\lambda_{i+1} = \lambda_i e^{-\theta_i}, \tag{1}$$

where λ_i and θ_i, $i = 1, \ldots, n$, are nonnegative. From (1), it is clear that the parameter θ_i plays a relevant role in describing the effect of the intervention during software testing. If $\theta_i = 0$, then there is no debugging effect on software reliability, which increases (decreases) if $\theta_i > 0$ ($\theta_i < 0$). The latter case is due to introduction of new bugs to the software.

A slightly modified version of this model was proposed by Gaudoin (1999), who considered

$$\lambda_{i+1} = (1 - \alpha_i - \beta_i)\lambda_i + \mu\beta_i,$$

for modelling the more realistic case where intervention at each stage may introduce new bugs while fixing the existing ones at the same time. The effect of the positive intervention is modelled by α, whereas β is used for the negative one.

In a Bayesian framework, Basu and Ebrahimi (2003) proposed exponential interfailure times with a martingale process Gamma prior such that $E(\lambda_{i+1}|\lambda_i) = \lambda_i$, $i = 1, \ldots, n$.

More recently, Durand and Gaudoin (2005) considered a hidden Markov model (HMM) similar to the one we introduce in Section 2, but they considered a non Bayesian approach and used an EM algorithm to obtain maximum likelihood estimates. They motivated the choice of a HMM since, under suitable conditions, a process of general interarrival times can be described by exponential interarrivals conditional upon a HMM. They applied the Bayesian information criterion (BIC) to choose among models with different number of states of the hidden process. As mentioned by the authors, the choice of the starting values of the EM algorithm could have an influence on both parameter estimation and model selection. As indicated by the sensitivity studies we performed, estimation from our MCMC approach, based on very simple posterior conditional distributions, was not sensitive to starting points. Another advantage of the proposed Bayesian approach is that it allows us to incorporate available information about the debugging process to specify prior distributions. Furthermore, predictive distributions of future failure times as well as posterior distributions of the states of the hidden Markov chain at any point can be obtained in a straightforward manner via the calculus of probability.

Finally, Ravishanker, Liu and Ray(2008) defined a hidden Markov model governing the parameters of a nonhomogeneous Poisson process where failures are observed as counts in nonoverlapping time intervals. The counts are independently Poisson distributed with mean value given by

$$\theta \left\{ \prod_{j=1}^{i-1} (1 - p_{t_j}) \right\} p_{t_i}$$

for interval $[t_{i-1}, t_i)$ as $i = 1, \dots, T$, where $p_{t_i} = 1 - \exp\{\beta_{S_{t_i}}(t_i - t_{i-1})^\alpha\}$. The random quantity S_{t_i} follows a hidden Markov chain with a finite state space, such that $\beta_{S_{t_i}} = \beta_j$ if $S_{t_i} = j$. This model can be viewed as a nonhomogeneous Poisson Process with Markov switching mean value function, where the possible switching occurs at a sequence of pre-specified time points. The paper by Ravishanker *et al.* (2008) differs from ours in several aspects. First of all, they consider modeling of failure counts in fixed intervals due to *iterative development* whereas in our setup we model failure times by taking into account potential changes in the failure characteristics due to removal or introduction of bugs. Secondly, our model, unlike Ravishanker *et al.*, makes no assumption on the number of bugs initially in the software. Furthermore, the MCMC methods used in the two papers are significantly different especially for model selection. Our approach for assessing the unknown dimension of the hidden Markov chain is based on the marginal likelihood and thus is based on Bayes factors whereas Ravishanker *et al.* use the BIC for this purpose.

In this paper we present a new model motivated by potential introduction of new bugs to the software during the debugging process. The proposed model, based on a hidden Markov chain, assumes that times between failures are exponentially distributed with parameters depending on an unknown latent state variable which, in turn, evolves as a Markov chain. The model, implicitly, takes into account the possibility of not knowing if a new bug has been added at each stage. Thus, it can be used not only to model the failure process but also to infer if new bugs are introduced at different stages of testing. We introduce an extension of the model by assuming ordering of the failure rates associated with the latent states. We present Bayesian analysis of both models using Markov chain Monte Carlo methods and discuss implementation issues. We also develop inference about the unknown dimension of the hidden Markov chain using marginal likelihoods. The proposed models and their Bayesian analysis as well as the marginal likelihood based approach for inferring the dimension of the Markov chain represent a contribution to the state of the art in software reliability analysis.

In Section 2 we present the hidden Markov model and its Bayesian analysis. We consider estimation of the dimension of the state space of the HMM in Section 3. An alternative choice of prior distribution, based on ordered failure rates, is discussed in Section 4. The model is applied to the Jelinski and Moranda's Naval Tactical data and Musa's System 1 data in Section 5. Discussion on current research and concluding remarks are presented in Section 6.

2 A hidden Markov model for software failures

During the development phase software goes through stages of testing. After each stage modifications are made to the software with the hope of removing bugs that are the causes of software failures. This process which is referred to as debugging is not perfect since it is possible to introduce new bugs during the process and unintentionally cause an increase in the failure rate. Since introduction of bugs is not observable, one can only infer it by modeling its effect on the failure rate over stages of testing.

We assume that, during the testing stages, the failure rate of the software is governed by a latent process Y. Let Y_t denote the state of the latent process at time t and, given the state at time t is i, assume that X_t, the failure time for period t, follows an exponential model given by

$$X_t|Y_t = i \sim \mathcal{E}(\lambda_i).$$

The states of the latent process reflect the effectiveness of the interventions, i.e. the design changes, to the software prior to the t-th stage of testing. The failure rate of the software depends on this latent random variable.

We assume that the latent process $Y = \{Y_t : t \geq 1\}$ is a Markov chain with a transition matrix P on a finite state space $E = \{1, \ldots, k\}$. The initial state Y_1 is given a uniform distribution on $\{1, \ldots, k\}$. Given the latent process, we assume that X_t's are conditionally independent, that is,

$$\pi(X_1, X_2, \ldots, X_n|Y) = \prod_{t=1}^{n} \pi(X_t|Y).$$

In the Bayesian setup we assume that the transition matrix P and the failure rate λ_i, for $i = 1, \ldots, k$, are all unknown quantities. For the components of the transition matrix, it is assumed that $P_i = (P_{i1}, \ldots, P_{ik})$, $i = 1, \ldots, k$, i.e. the i-th row of P, follows a Dirichlet distribution $\mathcal{D}ir(\alpha_{i1}, \ldots, \alpha_{ik})$, as

$$\pi(P_i) \propto \prod_{j=1}^{k} P_{ij}^{\alpha_{ij}-1} \tag{2}$$

with parameters α_{ij}, $i, j = 1, \ldots, k$, and such that the P_i's are independent of each other. For a given state $i = 1, \ldots, k$, we assume a Gamma prior

$$\lambda_i \sim \mathcal{G}(a_i, b_i),$$

with independent λ_i's.

If software failures are observed for n testing stages, then, given the observed data $x^{(n)} = (x_1, x_2, \ldots, x_n)$, we are interested in the joint posterior distribution of all unknown quantities $\Theta =(\lambda^{(k)}, P, Y^{(n)})$, where $\lambda^{(k)} = (\lambda_1, \ldots, \lambda_k)$,and $Y^{(n)} = (Y_1, \ldots Y_n)$. It is not computationally feasible to evaluate the joint posterior distribution of Θ in closed form. However, we can use a Gibbs sampler to draw samples from the joint posterior distribution.

The likelihood function is

$$\mathcal{L}(\Theta; x^{(n)}) = \prod_{t=1}^{n} \lambda_{Y_t} e^{-\lambda_{Y_t} x_t}$$

and the posterior distribution is given by

$$\pi(\Theta|x^{(n)}) \propto \lambda_{Y_1} e^{-\lambda_{Y_1} x_1} \Big[\prod_{t=2}^{n} P_{Y_{t-1},Y_t} \, \lambda_{Y_t} e^{-\lambda_{Y_t} x_t}\Big] \Big[\prod_{i=1}^{k} \pi(P_i) \, [\lambda_i]^{a_i-1} \, e^{-b_i \lambda_i}\Big],$$

where $\pi(P_i)$ is given by (2). The implementation of the Gibbs sampler requires draws from the full conditional distributions of the unknown quantities, that is, the components of Θ. We first note that, given $Y^{(n)}$, the full conditional distribution of the elements of P can be obtained as

$$P_i|Y^{(n)} \sim Dir\{\alpha_{ij} + \sum_{t=1}^{n} \mathbf{1}(Y_t = i, Y_{t+1} = j); j \in E\} \tag{3}$$

where $\mathbf{1}(\cdot)$ is the indicator function and, given $Y^{(n)}$, P_i's are obtained as independent Dirichlet vectors. Given $Y^{(n)}$, they are also independent of other components of Θ.

The full conditional posterior distribution of λ_i's can be obtained as

$$\lambda_i|Y^{(n)},\, x^{(n)} \sim \mathcal{G}(a_i^*, b_i^*) \tag{4}$$

where

$$a_i^* = a_i + \sum_{t=1}^{n} \mathbf{1}(Y_t = i)$$

and

$$b_i^* = b_i + \sum_{t=1}^{n} \mathbf{1}(Y_t = i)\, x_t.$$

Finally, we can show that the full conditional posterior distributions of Y_t's, as $t = 2, \ldots, n-1$ are given by

$$\pi(Y_t|Y^{(-t)},\, \lambda_{Y_t},\, x^{(n)},\, P) \propto P_{Y_{t-1},Y_t}\, \lambda_{Y_t} e^{-\lambda_{Y_t}\, x_t} P_{Y_t,Y_{t+1}} \tag{5}$$

where $Y^{(-t)} = \{Y_s; s \neq t\}$. The full conditional posterior distributions of Y_1 and Y_n are proportional to $\lambda_{Y_1} \exp\{-\lambda_{Y_1} x_1\} P_{Y_1,Y_2}$ and $P_{Y_{n-1},Y_n} \lambda_{Y_n} \exp\{-\lambda_{Y_n} x_n\}$, respectively. Note that the above is a discrete distribution with constant of proportionality given by

$$\sum_{j \in E} P_{Y_{t-1},\, j}\, \lambda_j\, e^{-\lambda_j\, x_t} P_{j,\, Y_{t+1}}$$

as $t = 2, \ldots, n-1$, with obvious adjustments for $t = 1$ and $t = n$.

Thus, we can draw a posterior sample from $\pi(\Theta|x^{(n)})$ by iteratively drawing from the given full conditional posterior distributions. If we start with an initial value of the states, say, $Y_0^{(n)}$, then we can update the probability transition matrix via (3). Then, given the data and $Y_0^{(n)}$, we can draw the failure rates independently using (4). Given these values, we can use (5) to draw a new sample for the states. We can repeat these iterations many times to obtain a joint posterior sample.

Posterior predictive distribution of X_{n+1}, after observing $x^{(n)}$, is given by

$$\pi(X_{n+1}|x^{(n)}) = \sum_{j \in E} \int \pi(X_{n+1}|\lambda_j)\, P_{Y_n,j}\ \pi(\Theta|\, x^{(n)})\ d\Theta,$$

which can be approximated as a Monte Carlo integral via

$$\pi(X_{n+1}|x^{(n)}) \approx \frac{1}{G} \sum_{g=1}^{G} \sum_{j \in E} P_{Y_n^g,j} \pi(X_{n+1}|\lambda_j^g),$$

where Y_n^g and λ_j^g are draws from the Gibbs sampler.

We note that as a result of the posterior analysis the hidden states may be ranked according to their associated failure rates, where smaller failure rates correspond to a more desirable environment. However, the subscripts of the λ_i's are not constrained to match the same ranking, so that label switching (hence multimodality) may occur while running the Gibbs sampler. But this is not a concern, given that we focus on summary statistics that are labelling-invariant. An alternate model where the failure rates are ordered is presented in Section 4.

3 Estimating the dimension of the state space in HMMs

Our development in the previous sections assumed that k, the dimension of the state space of the Markov chain Y, was known. An important issue in the Bayesian analysis of the HMMs is the estimation of the dimension of the state space. One approach to dimension selection is to consider this as a model selection problem in the Bayesian framework.

As pointed out by Kass and Raftery (1995), Bayesian model comparison/selection is made using Bayes factors which are obtained as the ratio of marginal likelihoods $p(D|i)$ under two competing models $i = 1, 2$ where D denotes the observed data. Note that in our case we have $D = x^{(n)} = (x_1, x_2, \ldots, x_n)$. In many problems $p(D|i)$ is not available in an analytical form and its evaluation using posterior Monte Carlo samples is not a trivial task. Thus, various alternatives to marginal likelihoods have been suggested for model selection using Monte Carlo samples; see for example Gelfand (1996).

However, in certain problems where a Gibbs sampler is used and all the full conditional distributions are known, it is possible to approximate the marginal likelihoods from the posterior samples using a method introduced by Chib (1995). In what follows we will illustrate how the approach by Chib (1995) can be extended for the hidden Markov models of the type discussed here. Our development follows Hock and Soyer (2006) who used the Chib's approach for hidden Markov models in signal processing. An alternative approach is that of Green (1995) which is based on reversible jump Markov chain Monte Carlo methods and provides posterior probabilities for candidate models. Another approach that provides posterior model probabilities using

Markov chain Monte Carlo is presented in Carlin and Chib (1995). A comprehensive review of these and other approaches is given in Han and Carlin (2001).

Note that, suppressing dependence on model i, the marginal likelihood for a particular model is expressed as

$$p(D) = \frac{p(D|\Theta)p(\Theta)}{p(\Theta|D)}, \tag{6}$$

where Θ is a vector of parameters. As pointed out by Chib (1995) the above holds for any value of Θ, say Θ^*, and the value of posterior density $p(\Theta^*|D)$ can be estimated by $\hat{p}(\Theta^*|D)$ using Monte Carlo samples. Since $p(D|\Theta^*)$ and $p(\Theta^*)$ can be evaluated at Θ^*, the log marginal likelihood can be estimated as

$$ln\, \hat{p}(D) = ln\, p(D|\Theta^*)\ + ln\, p(\Theta^*) -\ ln\, \hat{p}(\Theta^*|D). \tag{7}$$

In evaluating the above, the only term which is not readily available is $\hat{p}(\Theta^*|D)$, but as shown in Chib (1995) this can be obtained using the outputs from the Gibbs sampler. In our case, we also have the latent variables $\boldsymbol{Y}^{(n)}$ as a part of the unknown parameter vector. Thus, we can write $p(D) = p(x^{(n)})$ as

$$p(x^{(n)}) = \frac{p(x^{(n)}|\boldsymbol{\lambda}^{(k)})p(\boldsymbol{\lambda}^{(k)}|\boldsymbol{Y}^{(n)})p(\boldsymbol{Y}^{(n)}|\boldsymbol{P})\, p(\boldsymbol{P})}{p(\boldsymbol{\lambda}^{(k)},\, \boldsymbol{P},\, \boldsymbol{Y}^{(n)}|x^{(n)})}, \tag{8}$$

where all the terms in the numerator of (8) can be evaluated at $(\boldsymbol{\lambda}^{(k)},\, \boldsymbol{P},\, \boldsymbol{Y}^{(n)})$ =$(\boldsymbol{\lambda}^{*(k)},\, \boldsymbol{P}^*,\, \boldsymbol{Y}^{*(n)})$. We note that (8) holds for any value of $(\boldsymbol{\lambda}^{(k)},\, \boldsymbol{P},\, \boldsymbol{Y}^{(n)})$, but, as pointed out in Chib (1995), it can be more accurately approximated by evaluating it at a high density point. Thus, the posterior modes, that can be easily approximated from the Gibbs output, will be used for $(\boldsymbol{\lambda}^{*(k)},\, \boldsymbol{P}^*,\, \boldsymbol{Y}^{*(n)})$. In Section 5 we give details on which modes are actually needed and on how we overcome the label switching problem. To approximate $p(x^{(n)})$ we need to obtain $p(\boldsymbol{\lambda}^{*(k)},\, \boldsymbol{P}^*,\, \boldsymbol{Y}^{*(n)}|x^{(n)})$ which is not immediately available. Using the multiplication rule we can write

$$p(\boldsymbol{\lambda}^{*(k)},\, \boldsymbol{P}^*,\, \boldsymbol{Y}^{*(n)}|x^{(n)}) = p(\boldsymbol{\lambda}^{*(k)}|\boldsymbol{Y}^{*(n)}, x^{(n)})p(\boldsymbol{P}^*|\boldsymbol{Y}^{*(n)})p(\boldsymbol{Y}^{*(n)}|x^{(n)}) \tag{9}$$

where $p(\boldsymbol{\lambda}^{*(k)}|\boldsymbol{Y}^{(n)}, x^{(n)})$ is the product of independent gamma densities and $p(\boldsymbol{P}^*|\boldsymbol{Y}^{*(n)})$ is the product of independent Dirichlet densities. Thus, the only term we need to evaluate is $p(\boldsymbol{Y}^{*(n)}|x^{(n)})$.

Again using the multiplication rule we write

$$p(\boldsymbol{Y}^{*(n)}|x^{(n)}) = p(Y_1^*|x^{(n)})\, p(Y_2^*|Y_1^*, x^{(n)}) \cdots p(Y_t^*|\boldsymbol{Y}^{*(t-1)}, x^{(n)}) \cdots p(Y_n^*|\boldsymbol{Y}^{*(n-1)}, x^{(n)})$$

where $\boldsymbol{Y}^{(t-1)} = (Y_1, \ldots, Y_{t-1})$. Note that the first term $p(Y_1^*|x^{(n)})$ can be estimated from the draws available from the Gibbs sampler as

$$p(Y_1^*|x^{(n)}) \approx \frac{1}{G}\sum_{g=1}^{G} p(Y_1^*|(\boldsymbol{\lambda}^{(k)})^{(g)}, (\boldsymbol{Y}^{-1})^{(g)}, \boldsymbol{P}^{(g)}, x^{(n)}). \tag{10}$$

Evaluation of the remaining densities requires additional sampling. For a general term $p(Y_t^*|\boldsymbol{Y}^{*(t-1)}, x^{(n)})$ which is given by

$$p(Y_t^*|\boldsymbol{Y}^{*(t-1)}, x^{(n)}) = \int p(Y_t^*|\boldsymbol{\lambda}^{(k)}, \boldsymbol{P}, \boldsymbol{Y}^{(s>t)}, \boldsymbol{Y}^{*(t-1)}, x^{(n)})$$

$$p(\boldsymbol{\lambda}^{(k)}, \boldsymbol{P}, \boldsymbol{Y}^{(s>t)} | \, \boldsymbol{Y}^{*(t-1)}, x^{(n)}) d\boldsymbol{\lambda}^{(k)} \, d\boldsymbol{P} \, d\boldsymbol{Y}^{(s>t)}, \tag{11}$$

where $\boldsymbol{Y}^{(s>t)} = (Y_{t+1}, \ldots, Y_n)$, we need to continue sampling from full conditionals of $(\boldsymbol{\lambda}^{(k)}, \boldsymbol{P}, \boldsymbol{Y}^{(s>t)})$ given $(\boldsymbol{Y}^{*(t-1)}, x^{(n)})$. In other words, additional sampling will use the full conditional distributions: $p(\boldsymbol{\lambda}^{(k)}|Y_t, \ \boldsymbol{Y}^{(s>t)}, \ \boldsymbol{Y}^{*(t-1)}, x^{(n)})$, $p(\boldsymbol{P}|Y_t, \ \boldsymbol{Y}^{(s>t)}, \ \boldsymbol{Y}^{*(t-1)})$, $p(Y_t|\boldsymbol{\lambda}^{(k)}, \ \boldsymbol{P}, \boldsymbol{Y}^{(s>t)}, \boldsymbol{Y}^{*(t-1)}, x^{(n)})$ and $p(Y_h|\boldsymbol{\lambda}^{(k)}, \boldsymbol{P}, Y_t, \boldsymbol{Y}^{(-h,\, s>t)}, \boldsymbol{Y}^{*(t-1)}, x^{(n)})$ for $h = t+1, \ldots, n$ where $\boldsymbol{Y}^{(-h,\, s>t)} = \{Y_s; s > t \text{ and } s \neq h\}$.

Thus, (11) can be evaluated as

$$p(Y_t^*|\boldsymbol{Y}^{*(t-1)}, x^{(n)}) \approx \frac{1}{G'}\sum_{g=1}^{G'} p(Y_t^*| \, (\boldsymbol{\lambda}^{(k)})^{(g)}, \boldsymbol{P}^{(g)}, (\boldsymbol{Y}^{(s>t)})^{(g)}, \boldsymbol{Y}^{*(t-1)}, x^{(n)})$$

where $((\boldsymbol{\lambda}^{(k)})^{(g)}, \boldsymbol{P}^{(g)}, (\boldsymbol{Y}^{(s>t)})^{(g)})$ represents samples from $p(\boldsymbol{\lambda}^{(k)}, \boldsymbol{P}, \ \boldsymbol{Y}^{(s>t)}|\boldsymbol{Y}^{*(t-1)}, x^{(n)})$. This completes all the terms needed to compute (9) and to approximate the marginal likelihood (8). We note that for each given model, which may represent a specific dimension k , the marginal likelihood (8) can be approximated and these are compared to find the model which is the most supported by the data.

In a recent paper, Früwirth-Schnatter (2004) showed that the marginal likelihood for Markov switching models obtained through Chib's method was biased due to potential multimodality induced by relabelling. For example, if the modes are so well separated that the Gibbs sampler visits only one mode, then one actually is sampling from a constrained space and therefore the full conditionals that are being averaged should be the full conditional on this space and not on the full space. However, this does not apply to our case, because $p(Y_t^*| \, (\boldsymbol{\lambda}^{(k)})^{(g)}, \boldsymbol{P}^{(g)}, (\boldsymbol{Y}^{(s>t)})^{(g)}, \boldsymbol{Y}^{*(t-1)}, x^{(n)})$ does not change if we impose an ordering constraint on the rates.

We will see in Section 5 that label switching never occurs with $k = 2$, but it may happen for larger values. In the former case, we are precisely in the situation just described. In the latter case, one cannot anticipate how many modes will be visited, but relative distribution values are represented correctly by the empirical distribution derived from the sampler. In this case the full conditional being averaged is also invariant under relabelling and thus no bias is induced by Chib's scheme in our model.

4 An alternative model for failure rates

Our development in Section 3 assumed the a priori independence of failure rates given the latent states. This conditional independence was also preserved a posteriori. Often, it is reasonable to expect that failure rates under different environments will be ordered implying that certain environments will be less failure prone than others. Such an expectation and thus, the implied dependence can be incorporated into the prior of the failure rates.

Such a prior can be obtained as an extension of McKay's bivariate gamma distribution (see Kotz et al. (2000)). More specifically, in the McKay's bivariate distribution we have $\lambda_1 \sim Gamma(a_1, \beta)$ and $\lambda_2|\lambda_1 \sim Gamma(a_2, \beta)$ over (λ_1, ∞). Thus,

$$\pi(\lambda_1, \lambda_2) = \frac{\beta^{a_1}}{\Gamma(a_1)} \lambda_1^{a_1 - 1} e^{-\beta\lambda_1} \, \frac{\beta^{a_2}}{\Gamma(a_2)} \, (\lambda_2 - \lambda_1)^{a_2 - 1} \, e^{-\beta(\lambda_2 - \lambda_1)}$$

where $0 < \lambda_1 < \lambda_2 < \infty$. The above can be written as

$$\pi(\lambda_1, \lambda_2) = \frac{\beta^{a_1 + a_2}}{\Gamma(a_1)\Gamma(a_2)} \lambda_1^{a_1 - 1} (\lambda_2 - \lambda_1)^{a_2 - 1} \, e^{-\beta\lambda_2}.$$

We can obtain the marginal of λ_2 as

$$\lambda_2 \sim Gamma(a_1 + a_2, \beta) \text{ and } \lambda_1|\lambda_2 \sim Beta(a_1, a_2; 0, \lambda_2).$$

The above can be easily extended to k random variables $\lambda_1, \lambda_2, \ldots, \lambda_k$ such that

$$\lambda_1 < \lambda_2 < \cdots < \lambda_k$$

by assuming that $\lambda_1 \sim Gamma(a_1, \beta)$, $\lambda_2|\lambda_1 \sim Gamma(a_2, \beta)$ over $(\lambda_1, \infty), \ldots,$ $\lambda_k|\lambda_{k-1} \sim Gamma(a_k, \beta)$ over over (λ_{k-1}, ∞). Thus, the joint distribution of $\lambda_1, \lambda_2, \ldots, \lambda_k$ is given by

$$\prod_{i=1}^{k} \frac{\beta^{a_i}}{\Gamma(a_i)} \, (\lambda_i - \lambda_{i-1})^{a_i - 1} \, e^{-\beta(\lambda_i - \lambda_{i-1})}, \tag{6}$$

where $\lambda_0 = 0$. We can write (6) as

$$\pi(\lambda_1, \ldots, \lambda_k) = \beta^{\sum_{j=1}^{k} a_j} \, e^{-\beta\lambda_k} \prod_{i=1}^{k} \frac{(\lambda_i - \lambda_{i-1})^{a_i - 1}}{\Gamma(a_i)} \; .$$

The above prior implies gamma marginals for λ_i's as

$$\lambda_i \sim Gamma(\sum_{j=1}^{i} a_j, \beta)$$

and conditional for λ_i, $i < k$ is a truncated beta such as

$$\lambda_i | \lambda_{(-i)} \sim Beta(a_i, a_{i+1}; \lambda_{i-1}, \lambda_{i+1})$$

where $\lambda_{(-i)} = \{\lambda_j | j \neq i\}$.

Except for the update step of rates, the MCMC sampling scheme with this different prior remains the same. A straightforward strategy (see Erkanli, Mazzuchi and Soyer, 1998) is to run rejection sampling to obtain a random variate from the full conditional distribution of λ_i, for each i, where the candidate value λ_i is repeatedly drawn from a $Beta(a_i, a_{i+1}; \lambda_{i-1}, \lambda_{i+1})$ until a uniform random variate is lower than

$$\left(\frac{\lambda_i}{z_i}\right)^{n_i} e^{-s_i(\lambda_i - z_i)}$$

where $n_i = \sum_{t=1}^{n} \mathbf{1}(Y_t = i)$ and $s_i = \sum_{t=1}^{n} x_i \mathbf{1}(Y_t = i)$ and

$$z_i = \begin{cases} \lambda_{i-1} & \text{if} \quad \frac{n_i}{s_i} \leq \lambda_{i-1} \\ \frac{n_i}{s_i} & \text{if} \quad \lambda_{i-1} < \frac{n_i}{s_i} < \lambda_{i+1} \\ \lambda_{i+1} & \text{if} \quad \frac{n_i}{s_i} \geq \lambda_{i+1} \end{cases} .$$

When $i = k$, $\lambda_{k+1} = +\infty$, and therefore only the first two conditions are applicable.

We note that in the above scheme, the candidate value is drawn from the prior, whereas the acceptance ratio is a likelihood ratio. In general it is not a good idea to use the prior distribution as a proposal distribution to get Monte Carlo samples from the posterior, but in our case the support of our proposal conveys information from the observed data.

5 Analysis of software reliability data

We next illustrate the use of the HMM by applying it to two well known datasets, the Jelinski and Moranda's Naval Tactical data and Musa's System 1 data. We also present analysis of some simulated data to discuss implementation issues.

5.1 Jelinski-Moranda data

The data, presented in Jelinski and Moranda (1972), consists of 34 failure times (in days) of a large military system, and is referred to as the Naval Tactical Data System (NTDS). In the analysis of the NTDS data, we consider two possible states for Y_t initially, i.e. $E = \{1, 2\}$ and assume uniform distributions for the rows P_i, $i = 1, 2$, of the transition matrix. We describe uncertainty about the λ's, by considering diffuse priors $\lambda_i \sim \mathcal{G}(0.01, 0.01)$, $i = 1, 2$. Gibbs sampler was run for 5000 iterations and no convergence problems were observed. In what

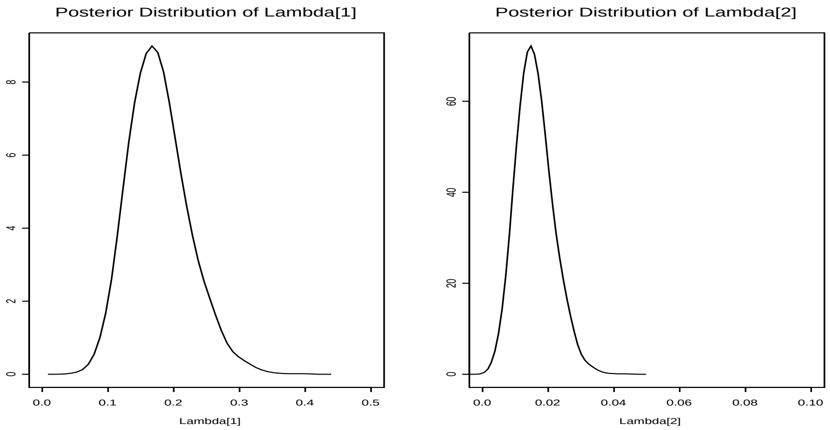


Figure 1: Posterior distributions of λ_1 and λ_2.

follows we present the posterior results for major quantities of interest as illustrated by plots and tables.

In Figure 1 we present the posterior distributions of λ_1 and λ_2. As can be seen from Figure 1, the posterior distribution of λ_1 is concentrated at higher values than that of λ_2 implying that environment 1 is the less desirable of the two environments. In other words, it represents the environment with higher failure rates and smaller expected time to failures.

Posterior distributions of transition probabilities are presented in Figure 2. We can see from Figure 2 that the process Y_t tends to stay in environment 1 (compared to environment 2) from one testing stage to the next one. This is implied by the posterior distribution of P_{11} which is concentrated around values that are higher than 0.6, whereas the posterior distribution of P_{22} is more dispersed.

Both Figures 1 and 2 are label-dependent, but they are valid, because label switching never occurred during our implementation of the Gibbs sampler, given that λ_2 is one order of magnitude smaller than λ_1.

Posterior predictive distribution of the next time to failure, that is, the distribution of X_{35} is shown in Figure 3. As we can see from the predictive density, the next time to failure is expected within few days. Table 1 presents the posterior distributions of the environment 1 for time periods, $t = 1, \ldots, 34$ as well as the observed time to failures for the periods. As we can see from the table the posterior probability of the "bad" environment (i.e. environment 1) decreases as we observe longer failure times.

The specified number of environments, that is $k = 2$, is supported by the the marginal likelihood, computed via Chib's method as illustrated in Section 3. As k ranges from 1 to 4,

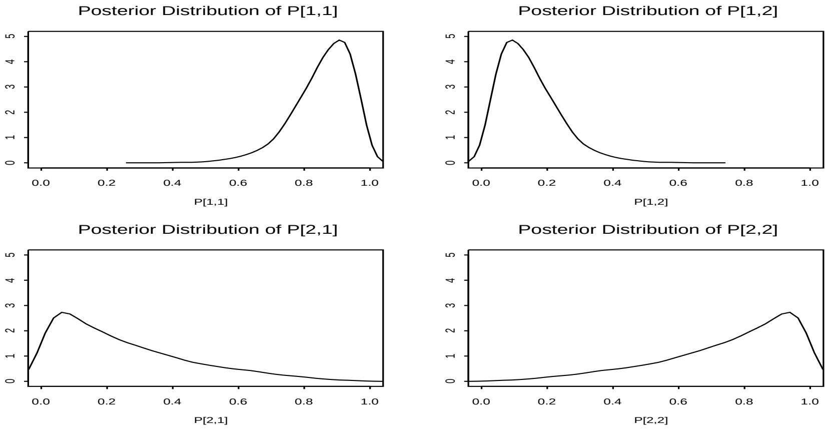


Figure 2: Posterior distributions of transition probabilities.

the logarithm of the marginal likelihood $p(D|k)$ takes the following values: -148.92, -139.81, -142.43, -144.63. With the selected hyperparameters, expression (8) has quite a simple form:

$$p(D|k) = \prod_{i=1}^{k} \frac{\Gamma(a_i^*)}{\Gamma(a_i)} \frac{b_i^{a_i}}{b_i^{*a_i^*}} \times \frac{\Gamma(k)^k}{k} \prod_{i=1}^{k} \frac{\prod_{j=1}^{k} \Gamma(1+m_{ij})}{\Gamma(k+m_i)} \times \frac{1}{p(\boldsymbol{Y}^{*(n)}|D)} \qquad (12)$$

where $m_{ij} = \sum_t \mathbf{1}(Y_t = i, Y_{t+1} = j)$ and $m_i = \sum_j m_{ij}$. We observe that factors depending on $\boldsymbol{\lambda}^{*(k)}$ and on $\boldsymbol{P}^*$ cancel out and we only need to provide $\boldsymbol{Y}^{*(n)}$.

For $k = 2$ we may let Y_t^* be the modal value in the sequence $\{Y_t^g\}_{g\geq 1}$ for every t, because label switching never occurs during the Gibbs sampling. When $k > 2$, label switching does occur and such an operation is meaningless, but we can retain information which is consistent throughout iterations, that is, the rank of $\lambda^g_{Y_t^g}$ within the vector of sorted rates $(\lambda^g_{(1)}, \ldots, \lambda^g_{(k)})$. Then we build a table of frequencies for the sequence of ranks of $\{\lambda^g_{Y_t^g}\}_{g\geq 1}$, where every rank ranges from 1 to k. The relative frequency of, say, rank 2 is the sample average estimate of the posterior probability that the environment is the second best at epoch t. Finally, we let Y_t^* be the rank with the highest frequency.

5.2 Musa's System 1 data

We next consider the System 1 data of Musa (1979) which consists of 136 software failure times. As in the case of the Jelinski-Moranda data, we consider only two states for Y_t, and assume uniform distributions for the row vectors P_i of the transition matrix, and the same

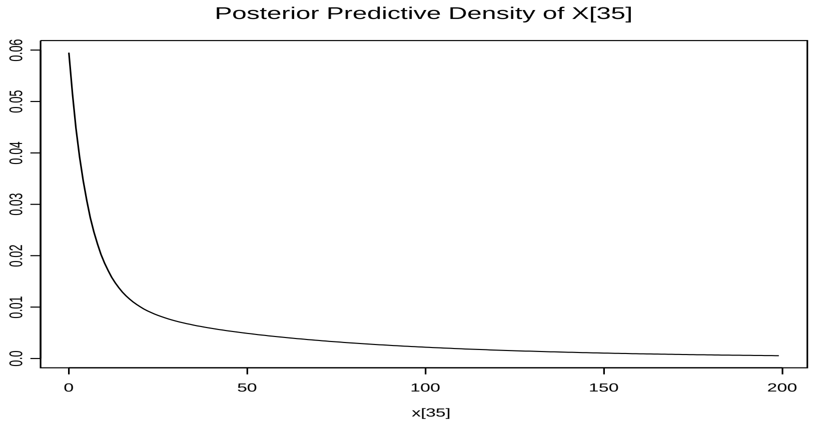


Figure 3: Predictive distribution of 35-th observation.

diffuse gamma distributions for the λ's. As before 5000 iterations of the Gibbs sampler was run and this led to convergence for all the quantities, with no occurrences of label switching. The posterior analysis for the major quantities of interest will be presented in the sequel using few plots.

From Figure 4, we can see that the times between failures tend to increase over time implying an overall reliability growth. The posterior distributions of λ_1 and λ_2 are presented in Figure 5. We can see from Figure 5 that the posterior distribution of λ_1 is concentrated around lower values than that of λ_2. Thus environment 1 is the more desirable of the two environments, that is, it represents the environment with smaller failure rates and larger expected time to failures. In Figure 6 we present the posterior distributions of transition probabilities. We can see from the figure that the process Y_t tends to stay in the same state from one testing stage to the next one. Posterior predictive distribution of the next time to failure, that is, the distribution of X_{137} is shown in Figure 7. As can be seen from the figure, the time to the next failure in this case has more variability than the one in the Jelinski-Moranda data shown in Figure 3.

In Figure 8 we present the posterior probabilities $P(Y_t = 1|D)$ for the "good" environment, that is, for environment 1, for time periods $t = 1, \ldots, 136$. As we can see from the figure, the posterior probability is rather low for most of the first 80 testing stages implying that modifications which are made to the software during these stages have not improved the reliability from one period to the next. On the other hand, the posterior probabilities for environment 1 wander around values higher than 0.85 for most of the stages implying the improvement in the reliability achieved during the later stages. We note that as in the case of the Jelinski-Moranda data, the higher posterior probabilities in Figure 8 are associated with longer failure times shown

Table 1: Posterior probabilities of state 1 over time.

t	X_t	$P(Y_t = 1 \mid D)$	t	X_t	$P(Y_t = 1 \mid D)$	t	X_t	$P(Y_t = 1 \mid D)$
1	9	0.8486	2	12	0.8846	3	11	0.9272
4	4	0.9740	5	7	0.9792	6	2	0.9874
7	5	0.9810	8	8	0.9706	9	5	0.9790
10	7	0.9790	11	1	0.9868	12	6	0.9812
13	1	0.9872	14	9	0.9696	15	4	0.9850
16	1	0.9900	17	3	0.9886	18	3	0.9858
19	6	0.9714	20	1	0.9584	21	11	0.7100
22	33	0.2036	23	7	0.3318	24	91	0.0018
25	2	0.6012	26	1	0.6104	27	87	0.0020
28	47	0.0202	29	12	0.2788	30	9	0.2994
31	135	0.0006	32	258	0.0002	33	16	0.1464
34	35	0.0794						

in Figure 4.

The analysis with two environments is again justified by the values of the marginal likelihood. The estimators of the reduced conditional ordinates found by Chib's method are less well separated than those of the Jelinski-Moranda data, thus we did five independent repeated runs for each value of k. In Figure 9 we plotted the marginal likelihoods along with Student's confidence intervals. For comparison, we plotted also the marginals of Jelinski-Moranda data.

The marginal likelihood is highest at $k = 3$. However a run of the Gibbs sampler with three hidden states produces identical posterior distributions for the two smallest rates, whereas a third hidden state is included only to capture three null failure times at epochs 33, 61 and 104, with an associated rate averaging at 267.80. The time series plot of posterior probability that $Y(t) = 1$ is substituted by the time series plot of posterior probability that $Y(t)$ occupies a hidden state associated with the lowest failure rate, and this is also unchanged except for deeper valleys at the epochs of the null failure times.

5.3 Simulated data

By the analysis done so far, one may get the impression that few states are often enough to describe software reliability improvement. As a matter of fact we are assigning diffuse priors to rates, thus any hidden state allows for a substantial variability of failure times. However, if different environments have well-separated rates, then one can end up with more hidden states. Table 2 shows summaries from repeated runs for a simulated dataset of 60 failure times, where they are divided into three groups: twenty failure times have rate 0.01, the second twenty failure

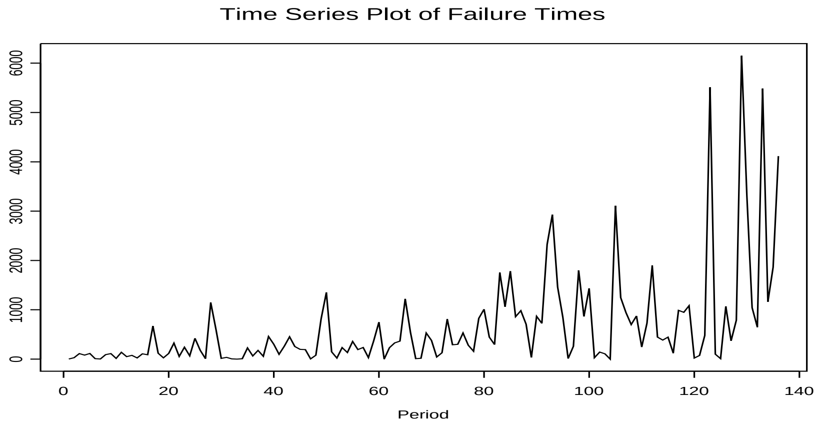


Figure 4: Failure times.

times have rate 0.001, and the remaining ones have rate 0.0001. The marginal likelihood is highest with three hidden states, as expected.

Table 2: Marginal likelihoods for a simulated dataset

k	avg. of $ln\hat{p}(D\|k)$	no. of runs	st. dev.
1	-562.921	0 (exact)	0
2	-516.6426	5	0.09
3	-514.5183	5	0.04
4	-518.2889	5	0.20

Another undesirable situation is overfitting, but with diffuse priors on rates this is unlikely. An analysis of a simulated dataset of 38 failure times with rate 0.0001, gives log-marginal likelihoods -391.8237, -394.0936, -396.2610 for 1, 2 and 3 hidden states respectively.

5.4 Constrained model

We repeated the above data analyses with prior (6), which required running the Gibbs sampler where the updating of the rates were done through rejection sampling. The hyperparameters were set to $a_i = 1$ for all i's and $\beta = k/10$, so that the largest rate has mean 10 and variance $100/k$ and λ_i has mean $(i/k)10$ and variance $(i/k^2)100$. To calculate also the marginal likelihood, one

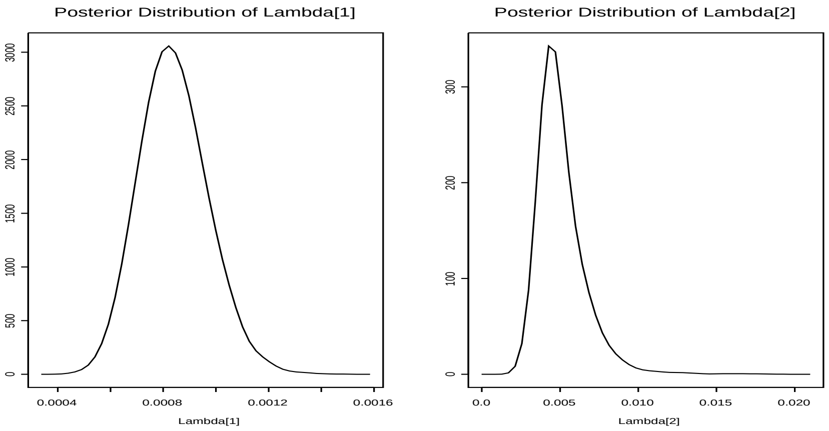


Figure 5: Posterior distributions of λ_1 and λ_2.

should implement the more general method of Chib and Jeliazkov (2001), because the model is nonconjugate and not all the full conditional distributions are available in closed form. However this proved unnecessary, because, as expected, the summary plots match the old ones not only in the obvious way for $k = 2$ (since label switching never occurs), but also for $k = 3$. As far as the rates are concerned, the posterior density of, say, λ_2 for the unordered case is derived from the sample sequence $\{\lambda^g_{(2)}\}$; the posterior probability that $Y(t) = 1$ is estimated by the relative frequency of $\lambda^g_{Y(t)}$ being the smallest rate in the unordered case, and no differences are observed with respect to the ordered case. One could also produce a meaningful plot of the density of a transition probability in the unordered case, by just collecting, say, the number of transitions from the state with the second smaller rate to the state with the third smaller rate to get an equivalent of P_{23} in the ordered case.

A visual confirmation of this equivalence comes from the comparison, in Figure 10, of the traces of the rates for the Jelinski-Moranda dataset with $k = 3$, for which we set $a_i = i$ and $b_i = 3/10$, to have the same marginal means and variances of the ordered case. Actually, the rates for the unordered case appear also to be less autocorrelated.

There are also some disadvantages in using the constrained model. The first one lies in the less straightforward sampling scheme: the rejection sampling used for the rates can be inefficient for some hyperparameter choices or some datasets. For example, the vectors of mean and standard deviation of the number of replicates needed to obtain a valid value of $\lambda^{(k)}$, with $k = 3$, are $(13, 8, 24)$ and $(15, 26, 42)$, respectively, for the Jelinski-Moranda dataset. These values are acceptable, but the same statistics for the Musa dataset are $(19, 7036, 269)$ and $(19, 10163, 318)$. The efficiency of the sampler for the two datasets compares in a similar way also for $k = 2$.

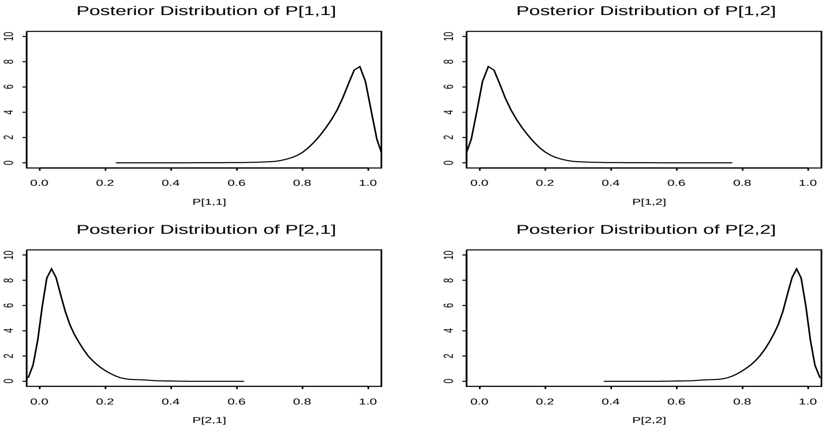


Figure 6: Posterior distributions of transition probabilities.

The second disadvantage is specific to the Musa dataset, which has three failure times that are exactly zero. Any sampling scheme that depends on the maximum likelihood estimate of the rate, for a group of observations associated with the same hidden state, fails when the three zero failure times form a separate group. In fact the maximum likelihood estimate would be $+\infty$, and this is precisely what happens for $k = 3$ with s_3 being occasionally zero, whereby the rejection sampling scheme fails.

6 Concluding remarks

This paper presents a Bayesian approach to describe possible introduction of bugs during software debugging. The choice of a hidden Markov model is justified not only by the nature of the problem, (that is, the introduction of bugs is not observed directly but only through its effects on reliability) but also by the need to provide a flexible model for the dependency between subsequent interfailure times. Implementation of the MCMC algorithm, estimates and forecasts based on samples from the posteriors are quite straightforward. The unknown states of the hidden Markov chain are treated as parameters within the MCMC. Their posterior distributions are helpful to describe the evolution of the reliability status of the software over time. The more challenging part of the MCMC implementation is for inferring the unknown dimension (number of hidden states) of the hidden Markov chain. Through a careful factorization of the involved probabilities it has been possible to perform successful model selection for both simulated and actual data.

There are other issues that can be addressed by considering extensions of our models. As

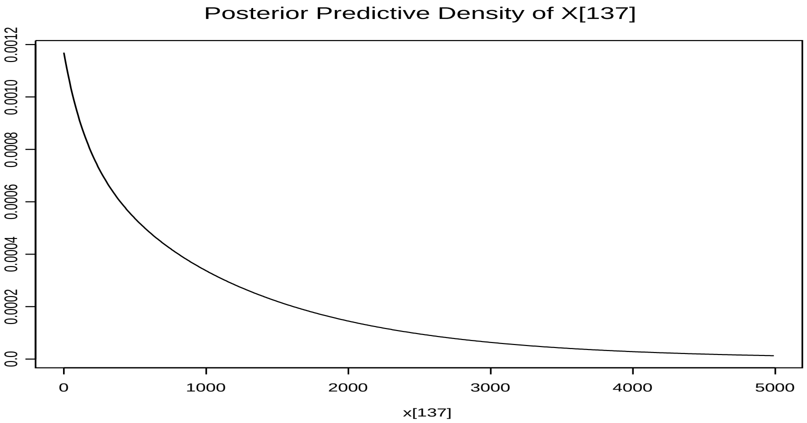


Figure 7: Predictive distribution of 137-th observation.

discussed both in Durand and Gaudoin (2005) and Ravishanker *et al.* (2008), transitions between states at each epoch could be constrained, e.g. allowing only for transitions to the closest, more and less reliable, states from the current one. Software metrics, e.g. number of code lines, as discussed in Wiper and Rodriguez Bernal (2001), could be used as covariates, possibly in the prior distributions of either the λ's parameters or the transition probabilities. The number of bugs is somehow related to the number of code lines; in fact, according to the Software Engineering Institute, even experienced programmers inject about one defect into every 10 lines of code. The current paper deals with interfailure times modelled through exponential distributions. In a forthcoming paper, we are considering a self-exciting point process to describe imperfect debugging phase.

References

[1] Basu, S., and Ebrahimi, N. (2003), "Bayesian Software Reliability Models Based on Martingale Processes", *Technometrics*, 45, 150–158.

[2] Carlin, B., and Chib, S. (1995), "Bayesian Model Choice by Markov Chain Monte Carlo", *Journal the Royal Statistical Society*, Ser. B, 57, 473–884.

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

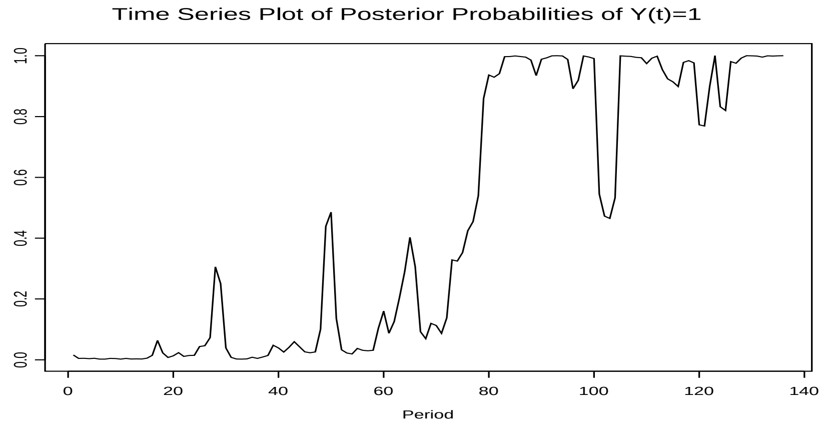


Figure 8: Posterior probability of $Y_t = 1$.

[4] Chib, S. (1995), "Marginal Likelihood from the Gibbs Output", *Journal of the American Statistical Association*, 90, 1313–1321.

[5] Durand, J.B., and Gaudoin, O. (2005), "Software Reliability Modelling and Prediction with Hidden Markov Chains", *Statistical Modelling*, 5, 75–93.

[6] Erkanli, A., Mazzuchi, T.A., and Soyer, R. (1998), "Bayesian Computation for a Class of Reliability Growth Models", *Technometrics*, 40, 14–23.

[7] Früwirth-Schnatter, S. (2004), "Estimating Marginal Likelihoods for Mixture and Markov Switching Models Using Bridge Sampling Techniques, *Econometrics Journal*, 7, 143–167.

[8] Gaudoin, O. (1999) "Software Reliability Models with Two Debugging Rates", *International Journal of Reliability, Quality and Safety*,6, 31–42.

[9] Gaudoin, O., Lavergne, C., and Soler, J.L (1994), "A Generalized Geometric De-eutrophication Software-Reliability Model, *IEEE Transactions on Reliability*, R-44, 536–541.

[10] Gelfand, A.E. (1996), "Model Determination Using Sampling-based Methods", in *Markov Chain Monte Carlo in Practice*, eds. W.R. Gilks, S. Richardson and D.J. Spiegelhalter, London: Chapman and Hall, pp.145–161.

[11] Green, P.J. (1995), "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination", *Biometrika*, 82, 711–732.

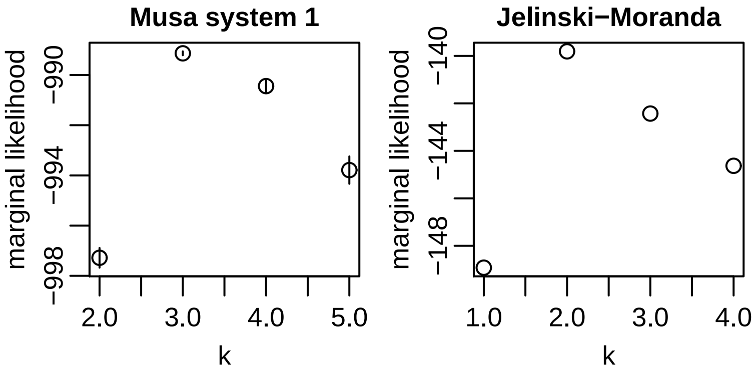


Figure 9: Plot of marginal likelihood against the number of hidden states.

[12] Han, C., and Carlin, B. (2001), "MCMC Methods for Computing Bayes Factors: A Comparative Review", *Journal of the American Statistical Association*, 96, 1122–1132.

[13] Hock, M., and Soyer, R. (2006), "A Bayesian Approach to Signal Analysis of Pulse Trains", in *Bayesian Monitoring, Control and Optimization*, eds. B.M. Colosimo and E. del Castillo, London: Chapman and Hall, pp. 215–243.

[14] Jelinski, Z., and Moranda, P. (1972), "Software Reliability Research", in *Statistical Computer Performance Evaluation*, ed. W. Freiberger, New York: Academy Press, pp. 465–497.

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

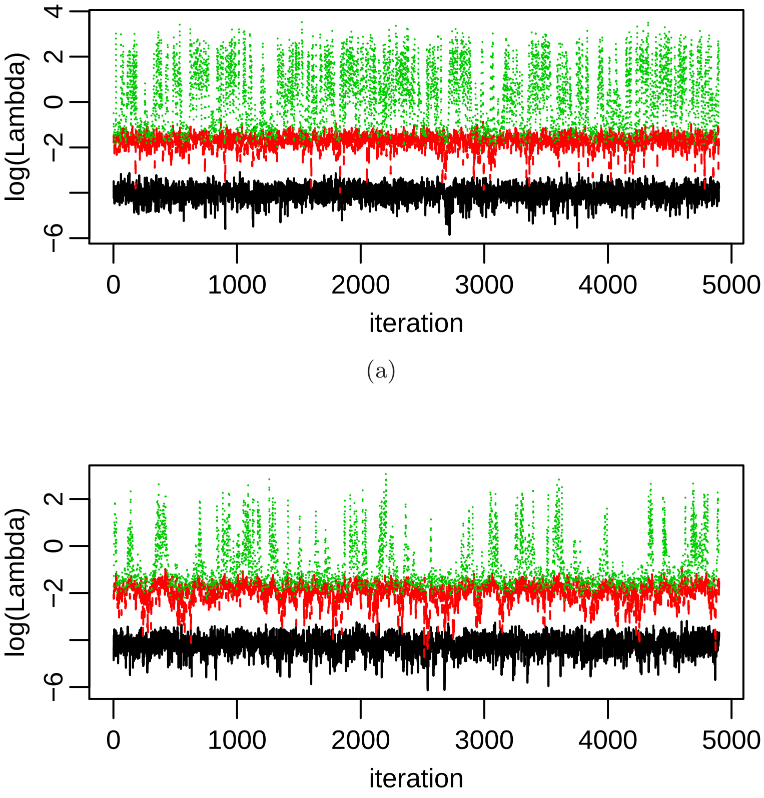
[16] Kremer, W. (1983), "Birth-Death and Bug Counting", *IEEE Transactions on Reliability*, R-32, 37–46.

[17] Kotz, S., Johnson, N.L, and Balakrishnan, N. (2000), *Continuous Multivariate Distributions: Models and Applications*, Chichester: John Wiley and Sons.

[18] Musa, J.D. (1979), "Software Reliability Data", Technical Report, Rome Air Development Center.

[19] Musa, J.D., and Okumoto, K. (1984), "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement", *Proceedings of the seventh International Conference on Software Engineering*, pp. 230–237.

[20] Ravishanker, N., Liu, Z., and Ray, B.K. (2008), "NHPP Models with Markov Switching for Software Reliability, *Computational Statistics and Data Analysis*, 52, 3988–3999.



(a)

(b)

Figure 10: Trace plots of the log-rates for the Jelinski-Moranda dataset: unconstrained model (a) and constrained model (b)

[21] Singpurwalla, N.D., and Wilson, S. (1999), *Statistical Methods in Software Engineering*, New York: Springer Verlag.

[22] Wiper, M.P. (2007), ”Software Reliability: Bayesian Analysis”, in *The Encyclopedia of Statistics in Quality and Reliability*, eds. F. Ruggeri, R.S. Kenett and F.W. Faltin, Chichester: John Wiley and Sons, vol. 4, pp. 1859–1863.

[23] Wiper, M.P., and Rodriguez Bernal, M.T. (2001), ”Bayesian Inference for a Software Reliability Model Using Metrics Information, in *Safety and Reliability: Towards a Safer World,*, eds. E. Zio, M. Demichela, and N. Piccinini), Torino: Politecnico di Torino, pp. 1999-2006.