

Non-Parametric Methods in Software Reliability

Simon Wilson

Dept of Statistics
Trinity College Dublin
Dublin 2, Ireland
simon.wilson@tcd.ie

Francisco Samaniego

Dept of Statistics
Univ of California Davis
Davis, California 95616, USA
fjsamaniego@ucdavis.edu

Abstract

In this paper we pursue a fully non-parametric Bayesian approach to software reliability. We propose a bug counting model, with an unknown number of bugs in the software, each with a distribution for time to discovery that is described by a Dirichlet process prior. We show that data can be considered as observations from the distribution function with considerable right censoring, thus the posterior of the distribution function is described by a beta-Stacy process. This model is applied to some data and its performance is briefly discussed.

1 Introduction

The number of software reliability models is impressive (Singpurwalla and Wilson 1999, Chapter 3). An immediate problem that arises from this is model choice. For many data sets it will not be clear, without fitting several models, which of them is appropriate.

One solution to the model choice problem is to bypass it by using a non-parametric approach. In this report we describe a Bayesian non-parametric model where a prior distribution is placed on the number of bugs in the code and on the space of distribution functions for discovery times. We show that the posterior distribution, while complicated, can be computed using Monte Carlo Markov chain (MCMC) methods. We illustrate with an example, concentrating on assessing the goodness of fit of the model. Our conclusion from the example is that the model fits our sample data reasonably well, but that there is a non-identifiability issue between the number of bugs and the distribution function that causes them to be badly estimated separately.

The paper is organised as follows. In Section 2 we describe the model. Section 3 briefly describes the MCMC computations of the posterior (necessarily brief since the implementation is complicated), and Section 4 describes how we can conduct model assessment. Section 5 concludes with the example and closing discussion.

2 Non-parametric Model for Software Failure

Assume that there are N bugs in the software, and that the discovery time for each bug has distribution function F , where both N and F are unknown. Bugs are assumed to be discovered independently. We assume that the software is used until a bug is discovered, whereupon the bug is fixed, and the software put back to use. At this point the process of discovery starts again but with one less bug in the code. This is the standard bug counting model with perfect repair; examples include the ubiquitous Jelinski and Moranda model, where $F(t) = 1 - \exp(-\beta t)$ for $\beta > 0$ (Jelinski and Moranda 1972).

This can also be thought of as a competing risks model, where the k th bug discovered is that with the minimum discovery time among the remaining $N - k + 1$ undiscovered bugs. We denote the k th discovery time by T_k , and the observed k th time by t_k . The survival function of T_k is $\bar{F}_k(t) = (1 - F(t))^{N-k+1}$. Alternatively, we can view the k th discovery time as one observed time at t_k and $N - k$ right censored times from F at t_k , the $N - k$ censored times being those of the remaining undiscovered bugs whose discovery times are greater than t_k . If we assume that K bugs are observed then our data consist of

observations t_1, \dots, t_K from F and $N - k$ right censored times at t_k , for $k = 1, \dots, K$. The likelihood for the data is then:

$$L(\text{data} | F, N) = \prod_{k=1}^K \frac{dF(t_k)}{dt} (1 - F(t_k))^{N-k}. \quad (1)$$

We specify a prior for the unknown N and F . We assume that N and F are independent a priori. This simplifies the specification and is reasonable since F pertains to the failure time for only a single bug. Thus our prior specification is to separately specify a prior for N and one for F .

Given the lack of information on N , it will be necessary to specify a proper prior for N ; several techniques have been proposed in the context of bug counting, for example the use of software metrics (Wiper and Rodríguez 2001) or the use of elicitation techniques (Campodónico and Singpurwalla 1995). One might also use non-informative proper priors, such as discrete uniform on a finite range $\{0, \dots, N_{\max}\}$, where N_{\max} is a suitably chosen upper bound (the number of characters in the code is one). In all cases, analysis of the sensitivity of the posterior to the prior on N will be essential.

As for F , a standard Bayesian non-parametric analysis assumes that F is in the class of all discrete distribution functions on $[0, \infty)$ and uses a Dirichlet process prior on this class. In the case of uncensored data, the posterior for F is still Dirichlet (Ferguson 1973). Our data can be viewed as mainly observations censored from the right, so this conjugacy is lost. However, it is still possible to compute the posterior distribution of F if we generalise the Dirichlet process prior, as we describe next.

3 Posterior Calculations by MCMC

Conjugacy for the prior of F is maintained in the case of right censoring if we generalise the Dirichlet to the class of so-called neutral to the right (NTTR) Lévy process priors, and a particular type of such a process, called a beta-Stacy process prior; for a definition of the process and its properties see (Walker, Damien, Laud, and Smith 1999).

Unfortunately, the details of implementing this posterior calculations are too lengthy to report in this short paper. We restrict ourselves to describing essential details and give references where full details are described. For the purposes of this report, it is sufficient to know that we write $F(t) = 1 - \exp(-Z(t))$ where $Z(t)$ is a beta-Stacy process, and that $Z(t)$ is defined in terms of two functions $\alpha(s)$ and $\beta(s)$, which we must specify. Here we assume a Dirichlet prior with mean as an exponential distribution with rate β , which implies that $\beta(s) = e^{-\beta s}$ and $d\alpha(s) = \beta e^{-\beta s} ds$. Further, such processes have countably many points of discontinuity x_1, x_2, \dots , and can be written as the sum of independent increments W_1, W_2, \dots at these points and a continuous component $Z_c(t)$:

$$Z(t) = Z_c(t) + \sum_i I(x_i < t) W_i.$$

Given N , we interpret our data as consisting of an observed failure time and $N - k$ right-censored failure times at t_k , for $k = 1, \dots, K$. It can be shown that the beta-Stacy process has the property of conjugacy when it is used as a prior on the class of distribution functions in the presence of right censored data (Walker and Muliere 1997). The posterior distribution for F can be written $F(t) = 1 - \exp(-Z^*(t))$, where $Z^*(t)$ is a beta-Stacy process with defining functions $\alpha(s)$ and $\beta^*(s) = \beta(s) + \sum_{k=1}^K I(t_k \geq s) (N - k + 1)$ and independent increments at the t_k with distributions specified in (Walker and Muliere 1997). By following the method described in that paper, one can generate a realisation of $Z^*(t)$ and hence of F given N and data.

We can rewrite the likelihood in Equation 1 as

$$\prod_{k=1}^K (F(t_k) - F(t_k-)) (1 - F(t_k))^{(N-k+1)-1}, \quad (2)$$

since under this model there is an increment in $Z^*(t)$ at t_k . This likelihood multiplied by the prior for N gives us the distribution of N given F and data, which is easy to sample by discrete inverse transform.

Thus we can sample from F given N and data, by sampling the relevant beta-Stacy process, and sample from N given F and data. Repeated sampling from these gives us a Gibbs sampler for sampling from the joint posterior of F and N . Note that $\alpha(s)$ and $\beta(s)$, that define the prior for F , must be specified in this scheme.

4 Model Assessment

In model assessment, we compare observed with predicted failure times. An obvious predicted value for T_k is the posterior median of F_k . Letting $(F^{(l)}, N^{(l)})$ be the l th MCMC sample from the posterior distribution of F and N , then $F_k^{(l)} = 1 - (1 - F^{(l)})^{N^{(l)} - k + 1}$, is the l th sample of the distribution of T_k . If L samples have been taken, then the posterior mean of the $100\alpha\%$ point of the distribution of T_k is computed by

$$F_{k,\alpha} \approx \frac{1}{L} \sum_{l=1}^L F_{k,\alpha}^{(l)}, \quad (3)$$

where $F_{k,\alpha}^{(l)}$ is the $100\alpha\%$ point of $F_k^{(l)}$. The median is calculated from Equation 3 with $\alpha = 0.5$. A prediction interval for T_k is then the 2.5% and 97.5% points of the posterior distribution of F_k , again calculated by Equation 3. A problem with computing the higher percentiles of F is that we do not estimate F well in the right tail, because of the large amount of right censoring. This is more serious for k near the estimated value of N , whose distribution has a much longer tail. In the example below, application of Equation 3 for k near K does not give good estimates for the higher percentiles.

Another possible predicted value for T_k is the posterior mean of F_k . We do not use this here because expected values require that all of F_k be estimated, and as we have said, F_k is not estimated well in the right tail.

5 Examples and Discussion

We run an example where $N = 250$, $K = 200$ and F is an exponential distribution with rate $\beta = 0.01$; this is the Jelinski Moranda model. A Dirichlet prior for F is assumed with mean equal to the true exponential distribution, and a Poisson prior with mean 400 was placed on N . The MCMC sampler was run for 15000 iterations, with the first 5000 ignored as burn in. To reduce sample autocorrelations, only every 10th sample was stored, giving us a sample of 1000 realisations of F and N . Standard convergence diagnostics showed that this appeared to be more than enough to produce independent samples from the equilibrium distribution.

The left of Figure 1 shows a summary of the posterior distribution of F along with the true $F(t) = 1 - \exp(-0.01t)$. It is immediately obvious that F has been greatly overestimated, that is we are predicting that individual bugs are discovered more quickly than was actually the case. If we look at the posterior distribution of N (not shown), we see that it puts probability 0.98 that $N = 200$ (that is, all bugs have been observed), and the largest of the 1000 sampled values of N is 202, compared to the true N of 250. Thus the model has fitted a situation in which there are fewer bugs, failing more rapidly, than is actually the case. Some light is shone on this disappointing state of affairs is by looking at the right graph in Figure 1. Here we have plotted each successive discovery time in sequence, and the posterior 2.5th, 50th and 97.5th percentiles for F_k , with the warning that the 50th and 97.5th are not estimated well for failure numbers near 200. Here we see that the predicted distributions for F_k are quite consistent with what was observed. We conclude that, in terms of fitting the data, the model has not done badly, although it has done so with values of F and N that are not those used to generate the data.

The reason why this has happened may be that there is some non-identifiability between F and N given only discovery time data, that is there are many combinations of F and N that are more or less equally supported by the data. Faced with this non-identifiability, the model has picked N to be as small as possible and F to be as large pointwise as possible. An explanation for this choice is that the likelihood in Equation 2 is strictly decreasing in N , for $N \geq K$. Therefore the data, regardless of F , will place most

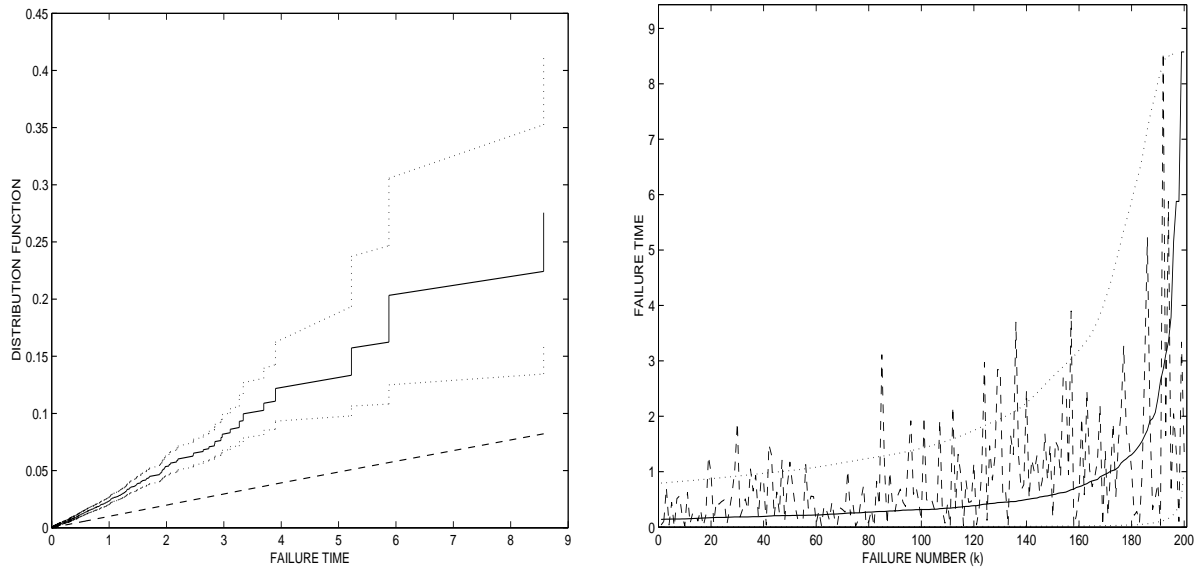


Figure 1: Analysis of simulated data. On the left, the posterior mean of F (solid line), with 2.5% and 97.5% pointwise posterior percentiles (dotted) and the true F (dashed). On the right, the simulated data (dashed line) with posterior predicted values: posterior median (solid line) with 2.5% and 97.5% posterior percentiles (dotted lines) for T_k , $k = 1, \dots, 200$.

weight on $N = K$. This conclusion is supported by several other examples on data sets of differing sizes with differing proportions of the bugs discovered.

The fact that the model fits in this manner has implications if we want to predict future discovery times and if we are interested in estimating the number of remaining bugs. Our conclusion is that, as the model stands, there is not enough information in discovery time data to both fit F and N satisfactorily. The next stage for this work is to investigate further the hypothesis on why this is happening, and to see what additional data would allow better estimation of N and F .

References

- Campodónico, S. and N. D. Singpurwalla (1995). Inference and predictions from Poisson point processes incorporating expert knowledge. *J. Amer. Statist. Assoc.* *90*, 220–226.
- Ferguson, T. S. (1973). A Bayesian analysis of some non-parametric problems. *Ann. Stat.* *1*, 209–230.
- Jelinski, Z. and P. Moranda (1972). Software reliability research. In W. Freiberger (Ed.), *Statistical computer performance evaluation*. Academic Press.
- Singpurwalla, N. D. and S. P. Wilson (1999). *Statistical methods in software reliability: reliability and risk*. New York: Springer.
- Walker, S. G., P. Damien, P. W. Laud, and A. F. M. Smith (1999). Bayesian nonparametric inference for random distributions and related functions. *J. Roy. Statist. Soc. B* *61*, 485–527.
- Walker, S. G. and P. Muliere (1997). Beta-stacy processes and a generalization of the Pólya-urn scheme. *Ann. Stat.* *25*, 1762–1780.
- Wiper, M. P. and M. T. Rodríguez (2001). Bayesian inference for a software reliability model using metrics information. Technical Report 01-20, Depto. de Estadística y Econometría, Universidad Carlos III de Madrid.