

Monte Carlo Methods for Bayesian Analysis of Survival Data Using Mixtures of Dirichlet Process Priors

Hani Doss

Department of Statistics
Ohio State University
Columbus, OH 43210-1247

Fred W. Huffer

Department of Statistics
Florida State University
Tallahassee, Florida 32306-4330

August 2002

Abstract

Consider the model in which the data consist of possibly censored lifetimes, and one puts a mixture of Dirichlet process priors on the common survival distribution. The exact computation of the posterior distribution of the survival function is in general impossible to obtain. This paper develops and compares the performance of several simulation techniques, based on Markov chain Monte Carlo and sequential importance sampling, for approximating this posterior distribution. One scheme, whose derivation is based on sequential importance sampling, gives an exactly iid sample from the posterior for the case of right censored data. A second contribution of this paper is a battery of programs that implement the various schemes discussed in this paper. The programs and methods are illustrated on a data set of interval-censored times arising from two treatments for breast cancer.

1 Introduction

Consider the model in which the data consist of possibly censored lifetimes. Specifically, there are iid random variables X_i from a distribution F , but these X_i 's may not be observed; rather, for each i , there is a set A_i within which X_i is known to lie. Right-censored data arises when the sets are either singletons or right-infinite intervals, and for this case the nonparametric maximum likelihood estimator is the well known Kaplan-Meier estimator, which is available in closed form. When the sets are arbitrary intervals, the EM algorithm can be used to obtain the nonparametric maximum likelihood estimator of F ; see Turnbull (1974, 1976).

Often we wish to hypothesize a low-dimensional parametric family of distributions H_θ ; $\theta \in \Theta \subset R^p$. For example, previous data of a similar type may suggest such a model, or one can infer a given parametric model from physical considerations. Parametric models, when they hold, are particularly useful when one wishes to make inference about F in a region where the data are sparse. In addition, they give rise to estimators that are more efficient than those based on a nonparametric model, although serious problems can arise if the parametric assumptions are not true.

A Bayesian nonparametric approach based on mixtures of Dirichlet processes (Ferguson 1973, 1974 and Antoniak 1974) offers a reasonable compromise between purely parametric and purely nonparametric models. Let ν be a prior distribution on Θ . For each $\theta \in \Theta$, define the measure $\alpha_\theta = M_\theta H_\theta$ on R , where $M_\theta > 0$ is a scalar. If θ is chosen from ν , and then F is chosen from $\mathcal{D}_{\alpha_\theta}$, the Dirichlet process with parameter measure α_θ , we say that the prior on F is a mixture of Dirichlet processes (with parameter $(\{\alpha_\theta\}_{\theta \in \Theta}, \nu)$). Often $M_\theta \equiv M$, i.e. the constants M_θ do not depend on θ . In this case, M can be interpreted as a precision parameter that indicates the degree of concentration of the prior on F around the parametric family $\{H_\theta; \theta \in \Theta\}$. For example, as $M \rightarrow \infty$ the distribution of F converges to $\int \delta_{H_\theta} \nu(d\theta)$, that is, $F = H_\theta$ and θ has prior ν , a standard Bayesian parametric model.

Doss (1994) reports that estimators based on mixtures of Dirichlet processes interpolate between the purely parametric and nonparametric models. For large values of M , the estimators are essentially equal to the Bayes estimator based on the parametric model. On the other hand, for small M , the estimators are essentially equal to the nonparametric maximum likelihood estimator except for time t in regions where the data are very sparse or non-existent. In those regions, the estimators make use of the parametric model.

For the case of right censoring, Susarla and Van Ryzin (1976) obtained a closed form expression for the mean and other moments of the posterior distribution of F when the prior on F is a single Dirichlet process. This was extended by Ferguson and Phadia (1979) to the case where the prior on F is a process neutral to the right (Doksum 1974). When the censoring pattern is more general, calculations become extremely complicated, and Monte Carlo appears to be the only feasible approach. Kuo and Smith (1992) considered the case of interval censored data and a Dirichlet process as the prior on F . They developed a Gibbs sampler that enables estimation of the posterior distribution of vectors of the form $(F(t_1), \dots, F(t_m))$, where for each j , t_j is an endpoint of one of the intervals A_i . We mention also the recursive algorithm of Newton and Zhang (1999). Although this method produces only an approximation, it can be computed very fast, and is useful in preliminary analyses prior to implementation of slower Monte Carlo methods.

In this paper we consider the problem of estimating the entire posterior distribution when the

prior on F is a mixture of Dirichlet processes. There exist a number of Monte Carlo methods, based on sequential importance sampling or Markov chain Monte Carlo, for approximating this posterior distribution. Some of these were originally developed for other models, but it is possible to obtain versions applicable to models involving censored data. Interestingly, for the case of right censored data, it is possible to adapt the sequential importance sampling method in a way that enables the generation of an exactly iid sample from the posterior. The main purpose of this paper is to describe the existing methods, propose some new ones, and compare the various methods both in terms of ease of implementation and speed of convergence.

Section 2 reviews the basic methods that are available. In Section 3 we show how some of the algorithms suggest new and more effective methods of carrying out Rao-Blackwellization. We have written functions in Splus that implement the algorithms that we have found most useful. In Section 4 we explain how to use these functions, and illustrate the results of Sections 2 and 3 and the use of our code on a data set that compares the time to cosmetic deterioration for breast cancer patients under two treatment regimens. The data consist of times that are “interval censored.” Section 5 discusses the various algorithms, compares them, and makes recommendations on their use.

For an overview of Bayesian nonparametric and semiparametric methods in survival analysis, see Sinha and Dey (1997) and Ibrahim, Chen, and Sinha (2001).

2 The Algorithms

Throughout this paper, \mathcal{L} will generically denote distribution or law. We will adopt the convention that subscripting a distribution indicates conditioning. Thus, if U and V are two random variables, $\mathcal{L}(U | V)$ and $\mathcal{L}_V(U)$ will both denote the conditional distribution of U given V . The prior on F is the mixture of Dirichlet processes

$$F \sim \int \mathcal{D}_{\alpha_\theta} \nu(d\theta). \quad (2.1)$$

Let **data** represent the event $\{X_i \in A_i; i = 1, \dots, n\}$. We are interested in various conditional distributions such as $\mathcal{L}_{\mathbf{data}}(F)$ and $\mathcal{L}_{\mathbf{data}}(X_1, \dots, X_n)$. Let X_{n+1} denote a future observation. We will also be interested in the predictive distribution $\mathcal{L}_{\mathbf{data}}(X_{n+1})$.

Before proceeding, it is useful to compare our setup with the following generic situation involving random effects. We have n different “centers” and at each one we gather data Y_i from the distribution P_{i,X_i} . The X_i ’s are thought of as iid from some distribution F , and uncertainty about this distribution is modelled by putting a mixture of Dirichlet processes prior on F . This hierarchical model is described as follows.

$$\text{Given } X_i, \quad Y_i \stackrel{\text{ind}}{\sim} P_{i,X_i}, \quad i = 1, \dots, n \quad (2.2a)$$

$$\text{Given } F, \quad X_i \stackrel{\text{iid}}{\sim} F, \quad i = 1, \dots, n \quad (2.2b)$$

$$\text{Given } \theta, \quad F \sim \mathcal{D}_{\alpha_\theta} \quad (2.2c)$$

$$\theta \sim \nu \quad (2.2d)$$

In (2.2a), the P_{i,X_i} ’s are known distributions, with densities p_{i,X_i} . An early version of this model and a computational algorithm for estimating posterior distributions were developed in

Escobar (1988, 1994). The model was developed further in many papers, with key contributions given in West, Müller and Escobar (1994), Escobar and West (1995, 1998), MacEachern (1994), Bush and MacEachern (1996), Liu (1996), and Neal (2000).

Our model differs from the standard hierarchical models involving (2.2) in two important respects. First, the likelihood (viewed as a function of X_1, \dots, X_n) arising at the top level in the hierarchy is a simple indicator function $\prod_i I(X_i \in A_i)$ in our model, whereas in standard hierarchical models it is the more complicated function $\prod_i p_{i, X_i}$. Secondly, in standard hierarchical models the support $\{x : p_{i, x}(Y_i) > 0\}$ is the same for all i , whereas in our model the supports (the sets A_i) differ.

The fact that the likelihood in our model is a simple indicator function gives us algorithmic possibilities which are unavailable (or very difficult to implement) in standard hierarchical models. As examples we note the approach reviewed in Section 2.2, the method in Section 2.1 that gives an exact sample from the posterior, and the method of Rao-Blackwellization in equation (3.2).

The fact that the supports A_i differ means that our model behaves in a radically different fashion than standard hierarchical models. In particular, as $M \rightarrow 0$, the posterior distribution in standard models leads to $X_1 = X_2 = \dots = X_n$ and F becoming degenerate (a point mass). Neither of these happens in our model. In fact, as $M \rightarrow 0$, our model produces sensible answers which are very similar to those given by the nonparametric MLE for this situation.

In all the methods we discuss, the random vector $\mathbf{X} = (X_1, \dots, X_n)$ and θ are viewed as missing data. The algorithms generate \mathbf{X} and θ . For some of the algorithms, the generation of θ is made possible by the lemma below, which gives the conditional distribution of θ given \mathbf{X} . The lemma is a special case of Lemma 1 of Antoniak (1974). For any vector \mathbf{u} we use $\#(\mathbf{u})$ to denote the number of distinct values in the vector. Throughout, we will assume that for each $\theta \in \Theta$, H_θ is absolutely continuous, with a density h_θ .

Lemma 1 *If the prior on F is given by (2.1), then the posterior distribution of F given \mathbf{X} is*

$$\int \mathcal{D}_{\alpha_\theta + \sum_{i=1}^n \delta_{X_i}} \nu_{\mathbf{X}}(d\theta), \quad (2.3)$$

where $\nu_{\mathbf{X}}$ is the measure which is absolutely continuous with respect to ν and is defined by

$$\nu_{\mathbf{X}}(d\theta) = c(\mathbf{X}) \left(\prod^{dist} h_\theta(X_i) \right) \left[\frac{(M_\theta)^{\#(\mathbf{X})} \Gamma(M_\theta)}{\Gamma(M_\theta + n)} \right] \nu(d\theta), \quad (2.4)$$

where the ‘dist’ in the product indicates that the product is taken over distinct values only, Γ is the gamma function, and $c(\mathbf{X})$ is a normalizing constant.

We note that if M_θ is constant in θ then (2.4) is very simple: the term in square brackets is absorbed into the normalizing constant, and $\nu_{\mathbf{X}}$ is just the posterior distribution of θ in the standard parametric Bayesian model in which $X_1, \dots, X_n \stackrel{iid}{\sim} H_\theta$ and θ has prior ν , except that only the distinct observations are used. In particular, if ν is conjugate to $\{H_\theta\}$, then $\nu_{\mathbf{X}}$ is available in closed form. Equation (2.4) gives the formula for $\nu_{\mathbf{X}}$ in the general case, and most of the results in this paper can be easily stated for the general case. For clarity of presentation, we henceforth will present the results in the special case where $M_\theta \equiv M$.

The remainder of this section presents the methods for estimating the posterior distribution of F . We discuss two algorithms based on sequential importance sampling, two Gibbs samplers, and two methods of modifying the Gibbs sampling algorithms designed to speed up convergence.

2.1 Sequential Importance Sampling

Sequential Importance Sampling (SIS) is a variant of importance sampling particularly well suited to missing data problems. See Kong, Liu, and Wong (1994) for a general introduction, and Liu (1996) and MacEachern, Clyde, and Liu (1999) for applications to models involving Dirichlet processes. The SIS scheme requires the data to have the format $(Y_1, Z_1), \dots, (Y_n, Z_n)$, where “ Y_i is the observed part” and “ Z_i is the unobserved part.” We obtain a replacement $\mathbf{Z}^* = (Z_1^*, \dots, Z_n^*)$ for the unobserved data by generating

$$\begin{aligned} Z_1^* &\sim \mathcal{L}(Z_1 | Y_1); \\ Z_2^* &\sim \mathcal{L}(Z_2 | Y_1, Z_1^*, Y_2); \\ &\vdots \\ Z_n^* &\sim \mathcal{L}(Z_n | Y_1, Z_1^*, \dots, Y_{n-1}, Z_{n-1}^*, Y_n). \end{aligned} \tag{2.5}$$

We repeat this, say J times, obtaining vectors $\mathbf{Z}^{*(j)}$; $j = 1, \dots, J$. These vectors are not drawn from $\mathcal{L}(\mathbf{Z} | \mathbf{Y})$ and must be appropriately re-weighted. For each vector, we form the predictive probabilities

$$\begin{aligned} v_1^{(j)} &= p(Y_1) \\ v_2^{(j)} &= p(Y_2 | Y_1, Z_1^*) \\ &\vdots \\ v_n^{(j)} &= p(Y_n | Y_1, Z_1^*, \dots, Y_{n-1}, Z_{n-1}^*), \end{aligned} \tag{2.6}$$

calculate

$$v^{(j)} = \prod_{i=1}^n v_i^{(j)}, \tag{2.7}$$

and to the vector $\mathbf{Z}^{*(j)}$ associate the weight

$$w_j = \frac{v^{(j)}}{\sum_{l=1}^J v^{(l)}}. \tag{2.8}$$

(In (2.6), p is used generically to denote joint distributions pertaining to the random vectors $(Y_1, Z_1), \dots, (Y_n, Z_n)$.) Kong, Liu, and Wong (1994) show that an expectation of the form $E(f(\mathbf{Y}, \mathbf{Z}) | \mathbf{Y})$ can be estimated by the weighted average

$$\sum_{j=1}^J w_j f(\mathbf{Y}, \mathbf{Z}^{*(j)}). \tag{2.9}$$

For notational convenience, in the SIS algorithms we describe below, we will drop the stars that we used in (2.5) and (2.6).

Let us now consider our censored data problem. In this case, we think of the data as consisting of the n pairs (A_i, X_i) ; $i = 1, \dots, n$, in which A_i is thought of as the “observed part” and X_i the “unobserved part,” and the pair (Θ, θ) , where θ is viewed as a latent variable on which we have no constraints. Our goal is to generate observations from $\mathcal{L}_{\text{data}}(\mathbf{X}, \theta)$, the conditional distribution of X_1, \dots, X_n and θ given that $X_l \in A_l$; $l = 1, \dots, n$.

There are two different strategies for carrying out SIS: we can put (Θ, θ) at the beginning of (2.5) or we can put it at the end. Putting (Θ, θ) at the beginning is more intuitive and leads to a more easily implemented SIS algorithm, but may produce highly variable importance sampling weights w_j in (2.8). The reason is that, once θ is chosen in the very first step (θ is chosen from the prior ν), it is used without change throughout the rest of the sequential generation. If the prior and posterior distributions of θ greatly differ, then this will result in highly nonuniform weights. (A general principle in SIS is to process the least informative observations as late as possible [see Section 2.3 of Kong, Liu, and Wong 1994]; in the first strategy, θ , on which we have no information, is processed first.) It turns out that we can effectively deal with this problem for the special case of right-censored data by slightly modifying the basic SIS algorithm given by (2.5)–(2.8); see the discussion later in this section. Putting (Θ, θ) at the end leads to an algorithm which is more difficult to implement, but has more uniform weights.

If we put (Θ, θ) in the beginning, Steps (2.5)–(2.8) consist of doing the following. First, generate

$$\theta \sim \nu. \tag{2.10}$$

Next, generate observations X_1, \dots, X_n , sequentially, as follows.

$$\begin{aligned} X_1 &\sim (\alpha_\theta)_{A_1} \\ X_2 &\sim (\alpha_\theta + \delta_{X_1})_{A_2} \\ &\vdots \\ X_n &\sim (\alpha_\theta + \sum_{i=1}^{n-1} \delta_{X_i})_{A_n} \end{aligned} \tag{2.11}$$

In (2.11) we have used the following notation. If π is any finite measure and B is a set, then π_B denotes the probability distribution obtained by restricting π to B and renormalizing it to be a probability measure; that is

$$\pi_B(A) = \pi(A \cap B) / \pi(B) \tag{2.12}$$

when $\pi(B) > 0$. We form the terms

$$v_0 = 1 \tag{2.13}$$

and

$$\begin{aligned}
v_1 &= P(X_1 \in A_1 \mid \theta) && \left(= \left(\frac{\alpha_\theta}{M} \right) (A_1) \right) \\
v_2 &= P(X_2 \in A_2 \mid \theta, X_1) && \left(= \left(\frac{\alpha_\theta + \delta_{X_1}}{M+1} \right) (A_2) \right) \\
&\vdots \\
v_n &= P(X_n \in A_n \mid \theta, X_1, \dots, X_{n-1}) && \left(= \left(\frac{\alpha_\theta + \sum_{i=1}^{n-1} \delta_{X_i}}{M+n-1} \right) (A_n) \right),
\end{aligned} \tag{2.14}$$

and calculate

$$v = \prod_{i=0}^n v_i. \tag{2.15}$$

We do this J times independently, obtaining vectors $(\theta^{(j)}, \mathbf{X}^{(j)})$; $j = 1, \dots, J$, to which we attach the weights w_j given by (2.8). Thus, for example, we estimate the posterior distribution of F given the data by

$$\sum_{j=1}^J w_j \mathcal{D}_{\alpha_{\theta^{(j)}} + \sum_{i=1}^n \delta_{X_i^{(j)}}}. \tag{2.16}$$

If we put (Θ, θ) at the end, then we generate the X_i 's using the unconditional distribution of the X_i 's, integrating over θ . The final step is to generate θ by using (2.4) of Lemma 1. Specifically, we generate

$$\begin{aligned}
X_1 &\sim \left(\int \alpha_\theta \nu(d\theta) \right)_{A_1} \\
X_2 &\sim \left(\int (\alpha_\theta + \delta_{X_1}) \nu(d\theta \mid X_1) \right)_{A_2} \\
&\vdots \\
X_n &\sim \left(\int (\alpha_\theta + \sum_{i=1}^{n-1} \delta_{X_i}) \nu(d\theta \mid X_1, \dots, X_{n-1}) \right)_{A_n} \\
\theta &\sim \nu_{\mathbf{X}}.
\end{aligned} \tag{2.17}$$

The predictive probabilities v_1, \dots, v_n in (2.6) are now given by

$$\begin{aligned}
v_1 &= \left(\int \frac{\alpha_\theta}{M} \nu(d\theta) \right) (A_1) \\
v_2 &= \left(\int \frac{\alpha_\theta + \delta_{X_1}}{M+1} \nu(d\theta \mid X_1) \right) (A_2) \\
&\vdots \\
v_n &= \left(\int \frac{\alpha_\theta + \sum_{i=1}^{n-1} \delta_{X_i}}{M+n-1} \nu(d\theta \mid X_1, \dots, X_{n-1}) \right) (A_n),
\end{aligned} \tag{2.18}$$

and the predictive probability corresponding to θ is $v_{n+1} = 1$. In (2.17) and (2.18), the measures $\nu(d\theta \mid X_1), \dots, \nu(d\theta \mid X_1, \dots, X_{n-1})$ are given by Lemma 1.

Right Censored Data

Although our methods of SIS apply to data that is arbitrarily censored, an interesting simplification occurs when the sets A_i satisfy the following nesting property: for all i and j , if $i < j$, then either $A_i \subset A_j$ or $A_i \cap A_j = \emptyset$. In this case, the values $\delta_{X_i}(A_j)$ occurring in (2.14) and (2.18) are not random, but deterministic. This implies, for instance, that the values v_i in (2.14) are functions of θ alone. This observation permits us to modify the first SIS algorithm so that it produces iid samples from the posterior. The second SIS scheme also benefits from the nesting property; see (2.21) below.

Consider the case where the data are right censored, that is, each of the sets A_i is either a singleton $\{T_i\}$ or a half-line (T_i, ∞) . Order the n observations so that $T_1 \geq T_2 \geq \dots \geq T_n$ and censored observations come before uncensored observations having the same value of T_i . With this ordering, the sets A_i satisfy the nesting property, so that the values v_i in (2.14) depend only on θ . If we take limits in (2.14), letting the interval A_i shrink down to a point, we find that each distinct uncensored value T_i gives a term $v_i \propto h_\theta(T_i)$. A censored observation leads to $v_i \propto M\bar{H}_\theta(T_i) + i - 1$ where $\bar{H}_\theta = 1 - H_\theta$.

Let τ_i be an indicator of censorship with $\tau_i = 1$ for uncensored observations and $\tau_i = 0$ for censored observations, and let ν' denote the density of ν . Define ξ to be the probability measure with density

$$\xi'(\theta) \propto \nu'(\theta) \prod_{i=1}^n v_i \propto \nu'(\theta) \prod_{i:\tau_i=1}^{\text{dist}} h_\theta(T_i) \prod_{i:\tau_i=0} (i - 1 + M\bar{H}_\theta(T_i)), \tag{2.19}$$

where the first product is over all distinct uncensored values. Suppose that in our first SIS algorithm, we replace (2.10) by $\theta \sim \xi$. Then we must replace (2.13) by the importance sampling factor $v_0 = \nu'(\theta)/\xi'(\theta)$. Now note that in the resulting modified SIS algorithm, the product v in (2.15) becomes constant. *Therefore, the weights (2.8) are all equal, and the pairs $(\theta^{(j)}, \mathbf{X}^{(j)})$ form an iid sample from $\mathcal{L}_{\text{data}}(\theta, \mathbf{X})$.* From this it follows that ξ is, in fact, the posterior distribution of θ . (We note that, in the general case where M_θ is not constant, a similar, but somewhat

more complicated, formula can be given for the posterior distribution of θ . See Doss and Huffer (2000).)

The program `ritcen` discussed in Section 4 implements the modified SIS scheme given above in the special case where $h_\theta(t) = \theta e^{-\theta t}$ and ν is a $\text{Gamma}(a, b)$ distribution. In this case, (2.19) becomes

$$\xi'(\theta) \propto p(\theta) \equiv f(\theta)g(\theta) \tag{2.20}$$

where $f(\theta)$ is the density of a $\text{Gamma}(a^*, b^*)$ distribution with $a^* = a + k$ and $b^* = b + s$ where k is the number of distinct uncensored times T_i and s is their sum, and

$$g(\theta) = \prod_{i: \tau_i=0} (i - 1 + M e^{-\theta T_i}).$$

For later use, note that if the first observation is censored, the first factor in this product is $M e^{-\theta T_1}$ which can be absorbed into the gamma density $f(\theta)$, so that we can assume without loss of generality that $g(\infty) \equiv \lim_{\theta \rightarrow \infty} g(\theta) > 0$.

We generate random variables with density ξ' by using a rejection scheme we now describe. Define $\ell(\theta) = \log g(\theta)$. It is easy to see that $\ell(\theta)$ is decreasing and convex. For fixed values $0 < \theta_1 < \dots < \theta_q < \infty$, define $\bar{\ell}(\theta)$ to be the function which agrees with $\ell(\theta)$ at $0, \theta_1, \dots, \theta_q$, is linear between these values, and takes the value $\ell(\theta_q)$ for $\theta > \theta_q$. Clearly $\ell(\theta) \leq \bar{\ell}(\theta)$ for all θ . Define $\bar{p}(\theta) = f(\theta) \exp(\bar{\ell}(\theta))$. Then $p(\theta) \leq \bar{p}(\theta)$ for all θ . Since $\bar{\ell}$ is piecewise linear, the function \bar{p} is proportional to a gamma density in each of the intervals (θ_{i-1}, θ_i) , where we take $\theta_0 = 0$ and $\theta_{q+1} = \infty$. Thus, it is straightforward to simulate a random variable Z from the density $b\bar{p}(\theta)$ where b is a normalizing constant. To do this, first compute $r_i = \int_{(\theta_{i-1}, \theta_i)} \bar{p}(\theta) d\theta$ and then $p_i = r_i / \sum_j r_j$ for $i = 1, \dots, q + 1$. Then, with probability p_i , generate a random variable from the gamma density in (θ_{i-1}, θ_i) conditional on it lying in this interval. Take this to be Z . Now that we can generate from $b\bar{p}$, we use a rejection scheme to generate from ξ' . Since $g(\infty) > 0$, for any $\epsilon > 0$ we can choose the values θ_i so that $p(\theta)/\bar{p}(\theta) > 1 - \epsilon$ for all θ . This allows us to make the probability of rejection arbitrarily small.

The routine `ritcen` uses the rejection algorithm described above. The quantities θ_i, p_i , and the parameters of the gamma distributions (proportional to $\bar{p}(\theta)$ in each of the intervals (θ_{i-1}, θ_i)) are all determined when the routine is started up, and then used throughout the sampling.

We note that the second SIS strategy (θ put at the end) also benefits when the sets satisfy the nesting property. For example, if the data are right censored and ordered in the way described earlier, then the predictive probability v_i in (2.18) of an uncensored observation ($\tau_i = 0$) satisfies

$$\frac{i - 1}{M + i - 1} \leq v_i \leq 1. \tag{2.21}$$

2.2 A Gibbs Sampler Based on a Constructive Representation of the Dirichlet Process Prior

The most natural way to implement the Gibbs sampler here is to proceed as is normally done in a Bayesian analysis of missing data problems under conjugacy. That is, consider the pair

(**missing data**, parameter ψ). In such a setup, if we knew the missing data, we would easily be able to find the conditional distribution of the parameter ψ , and if we knew the parameter ψ we would be able to generate the missing data. This is the approach taken by Doss (1994). He considers the pair $(\mathbf{X}, (\theta, F))$. If we knew (θ, F) , then generating \mathbf{X} subject to the constraint that $X_i \in A_i; i = 1, \dots, n$ would be trivial. And if we knew \mathbf{X} , then generating (θ, F) would be conceptually trivial from Lemma 1: we generate θ from (2.4) and then using that θ , we generate F from the integrand in (2.3), that is, generate $F \sim \mathcal{D}_\beta$ where $\beta = \alpha_\theta + \sum_{i=1}^n \delta_{X_i}$.

To fully describe this Gibbs sampling algorithm, we need to explain how we can generate F from \mathcal{D}_β . This is done by generating a sequence

$$B_1, B_2, \dots \stackrel{\text{iid}}{\sim} \text{Beta}(1, \beta(R)) \quad (2.22)$$

and an independent sequence $V_1, V_2, \dots \stackrel{\text{iid}}{\sim} \beta_0$ where $\beta_0 = \beta/\beta(R)$. Let

$$P_j = B_j \prod_{r=1}^{j-1} (1 - B_r),$$

and form the random distribution function

$$F = \sum_{j=1}^{\infty} P_j \delta_{V_j}. \quad (2.23)$$

This random distribution is distributed according to \mathcal{D}_β . (A rigorous proof of this fact is given in Sethuraman 1994.) To use the representation (2.23) to generate $X_1 \sim F$, we take a $U(0, 1)$ random variable U , and keep generating the beta random variables B_1, B_2, \dots (and the sequence V_1, V_2, \dots) until the first index J such that $\sum_{j=1}^J P_j \geq U$. If $V_J \in A_1$ we set $X_1 = V_J$. Otherwise, we repeat using an independent uniform variable, and continue until the corresponding “ V -value” is in the set A_1 . It is easy to see that the distribution of X_1 is F_{A_1} . Here, we are using the notation given in (2.12).

To generate X_2, \dots, X_n , we repeat this, using independent uniforms, but we use the same sequence B_1, B_2, \dots and V_1, V_2, \dots . (Note that we never need to generate all of F , but only the part of F that we need.)

This algorithm can be inefficient for two reasons. First, if the beta distributions in (2.22) have a mean that is close to 0 (which will happen if either the sample size is large or the hyperparameter M is large), a large number of betas will have to be generated before the terminating criterion $\sum_{j=1}^J P_j \geq U$ is met. Secondly, the conditioning on the event $X_i \in A_i$ is done through a rejection scheme, and this may have a low acceptance rate if the sets A_i have low probability.

2.3 A Gibbs Sampler Based on the Pólya Urn Scheme

This algorithm is based on the use of the “Pólya urn scheme” (PUS) of Blackwell and MacQueen (1973) introduced by Escobar (1988, 1994) as a tool for Gibbs sampling for models involving the Dirichlet process. In order to describe this algorithm, we briefly review the connection between the Dirichlet process and the PUS. Let α be a finite measure on R . A sequence

$\{S_1, S_2, \dots\}$ of random variables is defined to be a *Pólya sequence with parameter α* if for every $B \subset R$, we have $P(S_1 \in B) = \alpha(B)/\alpha(R)$, and for every n ,

$$P(S_{n+1} \in B \mid S_1, \dots, S_n) = \left(\alpha(B) + \sum_{i=1}^n \delta_{S_i}(B) \right) / (\alpha(R) + n).$$

From elementary properties of the Dirichlet process, it is easy to see that if $F \sim \mathcal{D}_\alpha$ and if $S_1, S_2, \dots \stackrel{\text{iid}}{\sim} F$, then $\{S_1, S_2, \dots\}$ is a Pólya sequence with parameter α , and for every n , S_1, \dots, S_n are exchangeable. (Blackwell and MacQueen (1973) proved the converse of this: If $\{S_1, S_2, \dots\}$ is a Pólya sequence with parameter α , then S_1, S_2, \dots are exchangeable, and the empirical distribution of $\{S_1, \dots, S_n\}$ converges a.s. to a limiting discrete measure H . Furthermore, $H \sim \mathcal{D}_\alpha$, and given H , we have $S_1, S_2, \dots \stackrel{\text{iid}}{\sim} H$.)

The PUS gives rise to a Gibbs sampler that runs over the vector $(X_1, \dots, X_n, \theta)$. Pick starting values $X_1^{(0)}, \dots, X_n^{(0)}, \theta^{(0)}$, with $X_i^{(0)} \in A_i$. Suppose that the current value of this vector is $(X_1^{(k-1)}, \dots, X_n^{(k-1)}, \theta^{(k-1)})$. Generate

$$X_i^{(k)} \sim \left(\alpha_{\theta^{(k-1)}} + \sum_{j < i} \delta_{X_j^{(k)}} + \sum_{j > i} \delta_{X_j^{(k-1)}} \right)_{A_i} \quad i = 1, \dots, n, \quad (2.24)$$

and then

$$\theta^{(k)} \sim \nu_{\mathbf{X}^{(k)}},$$

where $\nu_{\mathbf{X}^{(k)}}$ is given by (2.4). In this algorithm, the conditioning on the data is done by the restriction to A_i and subsequent renormalization in (2.24), which can be done in one step, i.e. without a rejection scheme.

We mention here the paper of Hanson and Johnson (2001), who develop extensions of the algorithms in Sections 2.2 and 2.3 to include covariates through an accelerated failure time model. Although as discussed earlier the algorithm in Section 2.2 can be inefficient, Hanson and Johnson (2001) report that, when covariates are included, the Markov chain it produces mixes much faster than does the chain in Section 2.3.

Consider now the two Gibbs sampling algorithms we have described. During the execution of either algorithm, for any k , the vector $\mathbf{X}^{(k-1)}$ is partitioned into a batch of clusters, with the X 's in the same cluster being equal. Each Gibbs sampler may mix very slowly when the posterior distribution gives high probability to vectors \mathbf{X} with large clusters (which can happen for instance, when M is small). The first algorithm generates $\mathbf{X}^{(k)}$ through the distribution F given in (2.23), which is based on the measure $\alpha_{\theta^{(k-1)}} + \sum_{i=1}^n \delta_{X_i^{(k-1)}}$, and the second uses (2.24). In either case, if a cluster is large, the probability that it will persist through the next iteration is high. The presence of persistent clusters results in a slowly mixing algorithm.

This problem has been encountered by other authors, in different contexts. There are two different approaches that have been proposed for dealing with it. One approach, proposed in MacEachern (1994), involves removing the locations of the clusters entirely by integration. The state space for the Gibbs sampler is collapsed to the finite space of possible configurations of clusters. The second approach, described in West, Müller, and Escobar (1994) and Bush and

MacEachern (1996), can be applied to either of the Gibbs samplers described in Sections 2.2 and 2.3, and involves appending to each cycle of the Gibbs sampler an extra step that moves the locations of the clusters of observations. We now describe how these ideas can be implemented in the context of our censored data problem.

2.4 A Gibbs Sampler on a Collapsed State Space

Let $\mathbf{c} = (c_1, c_2, \dots, c_n)$ be the cluster structure of the values X_1, X_2, \dots, X_n . The values c_i are integer labels assigned to the clusters of the X_i 's with X_i belonging to cluster c_i . Let \mathcal{C} denote the set of distinct values in (c_1, c_2, \dots, c_n) ; each of the clusters corresponds to a value in \mathcal{C} . (We are using the notation of Algorithm 3 in Neal (2000)).

Assume that $M_\theta \equiv M$. In this case, the prior distribution of (c_1, \dots, c_n) is independent of θ . Let $p(c_1, \dots, c_n)$ be the probability mass function of this distribution. (This distribution, the exact form of which we will not need, is just the obvious one given by the Pólya urn scheme, and can be written explicitly as a product of the conditional probabilities given in (2.7) of Neal (2000).) Noting that, conditional on θ , the distinct values among X_1, \dots, X_n are iid from H_θ , we may write the prior in our model as

$$\begin{aligned} \theta &\sim \nu \\ (c_1, \dots, c_n) &\sim p \\ Y_1, \dots, Y_{\#\mathbf{c}} &\stackrel{\text{iid}}{\sim} H_\theta \quad \text{and} \quad X_i = Y_{c_i} \text{ for } i = 1, \dots, n. \end{aligned}$$

Our data consists of the knowledge that $X_i \in A_i$ for $i = 1, \dots, n$, so that our “likelihood” function is the product of indicators

$$\prod_{i=1}^n I(X_i \in A_i) = \prod_{i=1}^n I(Y_{c_i} \in A_i) = \prod_{j \in \mathcal{C}} I(Y_j \in B_j) \quad \text{where } B_j = \bigcap_{i: c_i=j} A_i.$$

The product “prior \times likelihood” has the form

$$\nu(d\theta) p(c_1, \dots, c_n) \prod_{j \in \mathcal{C}} h_\theta(Y_j) \times \prod_{j \in \mathcal{C}} I(Y_j \in B_j). \quad (2.25)$$

Integrating out θ and the Y_j 's then gives the posterior distribution

$$p(c_1, \dots, c_n \mid \mathbf{data}) \propto p(c_1, \dots, c_n) g(c_1, \dots, c_n), \quad (2.26)$$

where

$$g(c_1, \dots, c_n) = \int \prod_{j \in \mathcal{C}} H_\theta(B_j) \nu(d\theta), \quad (2.27)$$

with the understanding that when B_j is a singleton, $H_\theta(B_j)$ is interpreted to be just the density h_θ at that point. Unfortunately, $g(c_1, \dots, c_n)$ generally has no simple closed form and must be obtained by numerical integration (an exception being the case of right-censored data, with the family of exponential distributions and a gamma prior).

We can sample from this posterior in many ways. The most obvious approach is to use a Gibbs sampler as in Algorithm 3 of Neal (2000) or MacEachern (1994). We visit the values c_1, \dots, c_n in sequence and reassign each to a new value, using the conditional posterior $p(c_i | \mathbf{c}_{-i}, \mathbf{data})$. Here we are using \mathbf{c}_{-i} to mean the vector (c_1, \dots, c_n) with the i^{th} component removed. It is easily seen from (2.26) and (2.27) that

$$p(c_i | \mathbf{c}_{-i}, \mathbf{data}) = bp(c_i | \mathbf{c}_{-i})g(c_1, \dots, c_n) \quad (2.28)$$

where b is a normalizing constant, and $p(c_i | \mathbf{c}_{-i})$ is the conditional prior given by

$$\begin{aligned} p(c_i = j | \mathbf{c}_{-i}) &= \frac{n_{-i,j}}{n-1+M} \quad \text{for } j \in \mathcal{C}_{-i} \\ p(c_i = j | \mathbf{c}_{-i}) &= \frac{M}{n-1+M} \quad \text{for } j \notin \mathcal{C}_{-i} \end{aligned} \quad (2.29)$$

Here \mathcal{C}_{-i} is the set of distinct values in \mathbf{c}_{-i} and $n_{-i,j}$ is the number of values $k \neq i$ for which $c_k = j$. Equation (2.29) is giving the probability that c_i will form a new cluster, which can be assigned an arbitrary integer label not in \mathcal{C}_{-i} . The normalizing constant b is obtained by requiring the sum of (2.28) over the possible values of c_i to be 1.

Once we have an observation from the stationary distribution of (c_1, \dots, c_n) , we can generate a point from $\mathcal{L}_{\mathbf{data}}(\mathbf{X}, \theta)$ by generating first θ and then \mathbf{X} , as follows:

$$\text{generate } \theta \sim a\nu(d\theta) \prod_{j \in \mathcal{C}} H_\theta(B_j) \quad (a \text{ is a normalizing constant}); \quad (2.30)$$

$$\text{generate } Y_j \sim (H_\theta)_{B_j} \text{ for } j \in \mathcal{C} \text{ and set } X_i = Y_{c_i} \text{ for } i = 1, \dots, n. \quad (2.31)$$

The distribution in (2.30) is the conditional distribution of θ given \mathbf{c} obtained from (2.25).

This algorithm appears promising in terms of mixing rate, but unfortunately, for general interval censored data, we have not found a way to carry out the numerical integrations required in (2.27) rapidly enough for the method to be competitive with the others.

2.5 Improving the Algorithms by Adding an Extra Step

In what follows, the expression ‘‘basic Gibbs sampler’’ refers to either the Gibbs sampler described in Section 2.2 or the one in Section 2.3; however, for the sake of concreteness, our explanation is in terms of the Gibbs sampler of Section 2.3. We can append to the basic Gibbs sampler an extra step which moves the clusters in such a way that the posterior distribution of (\mathbf{X}, θ) is still a stationary distribution for the new Markov chain. With the notation and facts established in the previous section this can now be described very quickly, as follows. Let (c_1, \dots, c_n) denote the cluster structure of \mathbf{X} as in the previous section. If the current value of the chain is (\mathbf{X}, θ) , the extra step consists of retaining from this only θ and the cluster structure (c_1, \dots, c_n) , and generating new values for X_1, \dots, X_n using (2.31).

The extra step moves the locations of the clusters. This increases the rate of mixing of the algorithm in two ways. There is an obvious direct benefit to moving the clusters, since otherwise the locations of the clusters can be highly persistent features of the Markov chain, particularly when M is small. Another benefit, which is more subtle and is indirect, is that this

can increase the rate of change of the cluster structure (c_1, \dots, c_n) . Each iteration of the basic Gibbs sampler visits each of the values X_1, \dots, X_n and re-assigns it to a new cluster. (This “new” cluster could actually be the same one it was previously assigned to, or an entirely new cluster which did not previously exist.) But the basic Gibbs sampler can only re-assign X_i to one of the clusters whose location is in A_i . This set of clusters may change fairly slowly if the clusters are persistent. By moving the locations of the clusters, performing the extra step can change the set of clusters whose locations are in A_i , and thus change the set of possible re-assignments in the next iteration of the basic Gibbs sampler.

The amount of this indirect benefit will depend on the extent and pattern of the overlaps among the data intervals A_1, \dots, A_n . For example, if the data intervals satisfy either $A_i = A_j$ or $A_i \cap A_j = \emptyset$ for all i and j , then the extra step can never change the set of clusters in any set A_i , and thus the extra step will produce absolutely no change in the mixing rate of the cluster structure. If there are sets A_i and A_j which overlap without being equal, then we can expect some increase in this mixing rate.

2.6 Uniform Ergodicity

As in MacEachern (1994), the collapsed state space algorithm of Section 2.4 produces an algorithm that is uniformly ergodic. This is because the Markov chain running on the vector (c_1, \dots, c_n) is uniformly ergodic (since the state space is finite), and it is not difficult to see that the vector $(X_1, \dots, X_n, \theta)$ inherits this uniform ergodicity. (For basic definitions regarding ergodicity, see Section 3.2 of Tierney (1994).)

On the other hand, none of the Markov chain algorithms of Sections 2.2, 2.3, or 2.5 need be uniformly ergodic. We do not provide a formal proof here, but rather discuss briefly the main issue, which has to do with the mixing parameter θ . In general, a Markov chain with transition probability function $P(x, A)$ is uniformly ergodic if and only if it satisfies a Doeblin condition, i.e. there exists a probability measure ρ and a constant ϵ , such that the probability measure $P(x, \cdot)$ is bounded below by $\epsilon\rho$ uniformly in x (see, e.g. Proposition 2 of Tierney (1994) or Theorem 3 of Athreya, Doss, and Sethuraman (1996)). Now, when all the sets A_i are bounded, a compactness argument gives a nonzero uniform lower bound on (2.4), the distribution from which θ is generated, and this is the crucial step in showing uniform ergodicity. However, when the sets A_i are not all bounded, as will happen when there are right censored observations, the simple compactness argument does not apply, and determining whether or not we have uniform ergodicity is a difficult problem which we have not solved.

It is possible to modify the algorithm which combines the Gibbs sampler of Section 2.3 with the extra step in Section 2.5 so that it becomes provably uniformly ergodic. We shall briefly describe the necessary modification using the notation of Section 2.4. Define $\mathbf{Y} = (Y_i : i \in \mathcal{C})$ so that \mathbf{Y} consists of the distinct values among X_1, \dots, X_n . Using (2.25), the conditional density of (θ, \mathbf{Y}) given $(\mathbf{data}, \mathbf{c})$ can be written as

$$f(\theta, \mathbf{y} \mid \mathbf{data}, \mathbf{c}) \propto \nu'(\theta) \prod_{i \in \mathcal{C}} h_\theta(y_i) I(y_i \in B_i), \quad (2.32)$$

where ν' is the density of ν . Divide the values in \mathbf{Y} into two groups according to whether the corresponding set B_i is bounded or unbounded, and let $\mathcal{B} = \{i \in \mathcal{C} : B_i \text{ is bounded}\}$,

$\mathcal{U} = \{i \in \mathcal{C} : B_i \text{ is unbounded}\}$. Define $\mathbf{Y}_b = (Y_i : i \in \mathcal{B})$ and $\mathbf{Y}_u = (Y_i : i \in \mathcal{U})$, and write $\mathbf{Y} = (\mathbf{Y}_b, \mathbf{Y}_u)$, and $\mathbf{y} = (\mathbf{y}_b, \mathbf{y}_u)$. Integrating over \mathbf{y}_u in (2.32) gives

$$f(\theta \mid \mathbf{y}_b, \mathbf{data}, \mathbf{c}) \propto \nu'(\theta) \prod_{i \in \mathcal{B}} h_\theta(y_i) \prod_{i \in \mathcal{U}} H_\theta(B_i). \quad (2.33)$$

Starting from this formula it is straightforward to show the validity of the following updating step: (i) generate θ from (2.33), and then (ii) generate \mathbf{X} as in (2.31). The ‘‘extra step’’ described in Section 2.5 consists only of part (ii) of the above. Part (i) is very easy to carry out in the exponential/gamma setup (where H_θ and ν are the exponential and gamma distributions, respectively) with interval censoring. Every unbounded set B_i must have the form $B_i = (u_i, \infty)$ so that $H_\theta(B_i) = e^{-\theta u_i}$. Thus, the distribution in (2.33) is just another gamma distribution. With a $\text{Gamma}(a, b)$ prior for θ , part (i) becomes simply $\theta \sim \text{gamma}(a^*, b^*)$ where $a^* = a + |\mathcal{B}|$, and $b^* = b + \sum_{i \in \mathcal{B}} Y_i + \sum_{i \in \mathcal{U}} u_i$ and $|\mathcal{B}|$ denotes the cardinality of \mathcal{B} . The values a^* and b^* are clearly bounded away from 0 and ∞ so that there is a uniform lower bound for these gamma densities. This leads to the uniform ergodicity of the Markov chain which incorporates this version of the extra step.

3 Rao-Blackwellization

In using the output of any of the samplers in Section 2 to estimate some quantity of interest, there are two methods of Rao-Blackwellization available. For ease of discussion, we shall describe these methods for the Gibbs samplers of Section 2.2 or Section 2.3 (with or without the extra step of Section 2.5).

Consider for instance the problem of estimating μ , the posterior mean of F , given by $\mu(t) = E(F(t) \mid \mathbf{data})$ for all t . (Note that μ coincides with the predictive distribution $\mathcal{L}_{\mathbf{data}}(X_{n+1})$.) If we use any of the Gibbs sampling algorithms for J cycles, we may form at the end of cycle j the distribution

$$\mu_j = E(F \mid \mathbf{X}, \theta) = \frac{\sum_{i=1}^n \delta_{X_i^{(j)}} + \alpha_{\theta^{(j)}}}{M + n},$$

so that the estimate formed using the simplest type of Rao-Blackwellization is

$$\tilde{\mu} = \frac{1}{J} \sum_{j=1}^J \mu_j. \quad (3.1)$$

Now μ will always have an absolutely continuous component, and in fact is entirely absolutely continuous if all observations are censored. But the estimate (3.1) will always have a discrete component, and so to estimate the density of the absolutely continuous part of μ , it is necessary to smooth $\tilde{\mu}$, for example by using a kernel density estimator.

To eliminate this problem we introduce a better method of Rao-Blackwellization. If we keep track of the cluster structure, we can in each cycle form the distribution μ_j^* defined by

$$\mu_j^* = E_{\mathbf{data}}(\mu_j \mid \mathbf{c}, \theta) = \frac{(\sum_{k \in \mathcal{C}} n_k (H_\theta)_{B_k}) + \alpha_\theta}{M + n}, \quad (3.2)$$

where $n_k = \#\{i : c_i = k\}$ is the number of elements in group k . We then take the average

$$\hat{\mu} = \frac{1}{J} \sum_{j=1}^J \mu_j^*.$$

In (3.2) we have suppressed the superscripts indexing the cycle to lighten the notation. The extra integration in (3.2) produces smoother estimates; in particular, if all observations are censored, (3.2) is absolutely continuous.

We note, however, that when estimating certain quantities such as $\mathcal{L}_{\text{data}}(F(t))$, the extra integration required by the approach in (3.2) may be too time-consuming, so that the resulting estimators are not practical.

The same methods of Rao-Blackwellization can be applied to the SIS algorithms, replacing the simple averages by weighted averages. For example, $\hat{\mu}$ becomes $\hat{\mu} = \sum_{j=1}^J w_j \mu_j^*$.

4 Splus Functions and Illustration on Breast Cancer Data

We have written programs `gibbs1`, `sis1`, `sis2` and `ritcen` which implement some of the algorithms discussed in this paper for the case in which we have the family of exponential distributions and a gamma prior, that is, when $\alpha_\theta = MH_\theta$ and $h_\theta(x) = \theta e^{-\theta x}$, and the prior ν for θ is a Gamma(a, b) distribution. In this situation our prior distribution is completely specified by the triple (a, b, M) . The purpose of these programs is to estimate aspects of the posterior distribution of the survival function $\bar{F}(t) = 1 - F(t)$. In particular, the programs supply estimates of the mean, variance and selected quantiles of the posterior distribution of \bar{F} . The program `gibbs1` uses the Gibbs sampling algorithm in Section 2.3 combined with the extra step described in Section 2.5. The program `sis1` uses the sequential importance sampling scheme described in equations (2.10)–(2.15) of Section 2.1, and `sis2` implements the scheme described in equations (2.17) and (2.18). Finally, `ritcen` implements the algorithm for generating iid observations from the posterior described at the end of Section 2.1. The `ritcen` program can be used only with right-censored data, but the other programs allow general interval censoring. In all our programs, the estimates of the posterior mean and variance of $\bar{F}(t)$ are computed using the type of Rao-Blackwellization in (3.2). The programs `gibbs1`, `sis1`, `sis2`, and `ritcen` are written as Splus functions which call dynamically loaded Fortran subroutines. These Splus functions are easy to use.

We illustrate on a data set involving time to cosmetic deterioration of the breast for women with Stage 1 breast cancer who have undergone a lumpectomy, for two treatments, these being radiation, and radiation coupled with chemotherapy. Radiation is known to cause retraction of the breast, and there is some evidence that chemotherapy worsens this effect. There is interest in the cosmetic impact of the treatments because both are considered very effective in preventing recurrence of this early stage cancer. The data come from a retrospective study of 46 patients who received radiation only and 48 who received radiation plus chemotherapy. Each woman made a series of visits to a clinician, who determined whether or not retraction had occurred. If it had, the time of retraction was known only to lie between the time of the present and last visits. Thus, the data consist of the interval, in months, in which deterioration occurred (interval censored observations), or the last time a patient was seen without having deterioration occurred

as yet (right censored observations). The data set is presented in Beadle et al. (1984a,b) (and also given in Klein and Moeschberger 1997, p. 18). Figure 1 gives a simple sample job using the `gibbs1` function.

```
rad <- gibbs1(bcdat.rad,c(2,88,10),10000,ci=.95)
summary(rad)
plot(rad)
```

Figure 1: Sample job.

All our functions have the same required arguments and use the same format for the data. Thus, in the sample job one can replace `gibbs1` by `sis1` or `sis2`. In the sample job, `bcdat.rad` is the data for the radiation-only group. This data set happens to involve interval censoring, but for right-censored data, one can also use `ritcen`. In all our functions, the three required arguments are (1) a two-column data matrix with each row describing one of the data intervals A_i , (2) a triple specifying the prior distribution, and (3) the simulation sample size. In the example, the estimates produced by `gibbs1` are based on 10,000 observations taken periodically from the output of a Gibbs sampler. The option `ci=.95` produces pointwise .95 posterior probability intervals for $\bar{F}(t)$. (The `ci` option is quite time consuming; on a machine doing about 16 specFP's, the first line in Figure 1 takes 11 seconds to execute without this option and 65 seconds with the option.)

The `summary` command in Figure 1 produces output like that in Table 1. The columns labeled `mean` and `sigma` contain estimates $\bar{\mu}(t)$ and $\sigma(t)$ of the posterior mean and standard deviation of $\bar{F}(t)$ at the time points t in the column labeled `time`. To be precise, $\bar{\mu}(t) \approx E(\bar{F}(t) \mid \mathbf{data})$ and $\sigma^2(t) \approx \text{Var}(\bar{F}(t) \mid \mathbf{data})$. These values are comparable (i.e. similar in format and interpretation) to the point estimates of $\bar{F}(t)$ and their standard deviations, which are supplied in frequentist analyses. The columns `se.mean` and `se.sigma` give estimated Monte Carlo standard errors for the values in `mean` and `sigma`. The columns `lower` and `upper` are estimates of the .025 and .975 quantiles of the posterior distribution of $\bar{F}(t)$. (In general, the option `ci=1- α` gives estimates of the $\alpha/2$ and $1-\alpha/2$ quantiles.)

We give a brief description of the way the estimates `upper` and `lower` are obtained. For any given value of t , the posterior distribution of $\bar{F}(t)$ is a mixture of beta distributions. For the SIS algorithms, a Monte Carlo approximation of this mixture is essentially (2.16). Fix a value of t and define $\psi(s) = P(\bar{F}(t) \leq s \mid \mathbf{data})$. The desired lower limit is the value s^* such that $\psi(s^*) = \alpha/2$. For any value of s , we can estimate $\psi(s)$ by

$$\hat{\psi}(s) = \sum_{j=1}^J w_j B(s \mid a^{(j)}(t), b^{(j)}(t)),$$

where

$$a^{(j)}(t) = \alpha_{\theta^{(j)}}((t, \infty)) + \sum_{i=1}^n \delta_{X_i^{(j)}}((t, \infty)), \quad b^{(j)}(t) = M + n - a^{(j)}(t),$$

and $B(s \mid a, b)$ denotes the Beta distribution with parameters a and b . (This formula is appropriate for the SIS algorithms; just replace the weighted average by a simple average to obtain the

time	mean	se.mean	lower	upper	sigma	se.sigma
0	1.000000	0.000000	1.000000	1.000000	0.000000	0.000000
2	0.977543	0.000091	0.906481	0.999982	0.026231	0.000078
4	0.956078	0.000177	0.876365	0.998735	0.032735	0.000064
22	0.701518	0.000187	0.558945	0.827112	0.068886	0.000062
24	0.676424	0.000229	0.535709	0.804416	0.069005	0.000081
25	0.661674	0.000270	0.524281	0.786416	0.067371	0.000080
46	0.437342	0.000485	0.276293	0.595088	0.081530	0.000127
48	0.404359	0.000893	0.188129	0.579200	0.098098	0.000463
60	0.315034	0.000869	0.066523	0.529565	0.119724	0.000286

Table 1: Output from the summary command.

corresponding formula for `gibbs1` and `ritcen`.) Estimates of $\psi'(s)$, and $\psi''(s)$ are given by similar formulas involving the derivatives $B'(s | a, b)$ and $B''(s | a, b)$. Since we can estimate ψ and its derivatives, we have implemented a Monte Carlo version of a Newton-Raphson type of procedure to estimate s^* . The values $\bar{\mu}(t)$ and $\sigma(t)$ are used to make an initial guess s_0 for s^* . Then a Monte Carlo sample is generated and used to estimate the values $\psi(s_0)$, $\psi'(s_0)$, and $\psi''(s_0)$. With these values we compute an improved guess s_1 . Successive Monte Carlo samples then produce further improvements s_2, s_3, \dots until a reasonable degree of convergence is achieved. The number and size of the Monte Carlo samples can be controlled by the user. The upper limits are obtained in a similar fashion.

The fact that our upper and lower probability limits for $\bar{F}(t)$ are exact (except for Monte Carlo errors) allows us to obtain corresponding probability limits for quantiles of \bar{F} . (In a frequentist nonparametric setting, establishing the validity of confidence intervals for quantiles is generally more difficult, as these are based on asymptotic approximations that require us to get a handle on the uniform behavior of the NPMLE near a given quantile. This is difficult even for the case of the Kaplan-Meier estimate.) Let $U(t)$ and $L(t)$ denote the upper and lower limits for $\bar{F}(t)$ as functions of t . For $0 < p < 1$, define $\bar{F}^{-1}(p) = \inf\{t : \bar{F}(t) \leq p\}$. Clearly $P(\bar{F}^{-1}(p) \leq t | \mathbf{data}) = P(\bar{F}(t) \leq p | \mathbf{data})$ for all p . Thus, the $\alpha/2$ point of the posterior distribution of $\bar{F}^{-1}(p)$ is just the value of t satisfying $L(t) = p$. Similarly, the $1 - \alpha/2$ point of the distribution is the solution of $U(t) = p$. Our software includes a program to compute posterior probability intervals for quantiles using this approach. This program also gives a ‘‘point estimate’’ of $\bar{F}^{-1}(p)$ obtained by solving $\bar{\mu}(t) = p$. As an example, applying our procedure to estimate $\bar{F}^{-1}(0.5)$ for the data and prior used in Figure 1 leads to a point estimate of 39.86 and a .95 probability interval of (27.6, 68.1). Our programs compute $\bar{\mu}(t)$, $L(t)$, and $U(t)$ at a grid of time points t specified by the user. In solving the equations $\bar{\mu}(t) = p$, $L(t) = p$, and $U(t) = p$, we use simple linear interpolation to approximate these functions for times between the grid points. This approach supplies reasonable accuracy so long as the grid of time points is sufficiently fine.

The standard errors `se.mean` and `se.sigma` in Table 1 give a rough indication of the accuracy of our estimates, allowing us to judge whether or not the simulation sample size is

sufficiently large. In this example, we are getting roughly 3 figure accuracy in our estimates. The method used to obtain the standard errors differs from function to function. For `gibbs1`, the method is a standard batching technique (see Ripley 1987, Section 6.2). In `sis1` and `sis2`, the independence of the SIS replicates $(v^{(j)}, \mathbf{Z}^{*(j)})$ described in (2.5)–(2.8) enables us to estimate the Monte Carlo accuracy as follows. The weighted average (2.9) is written as

$$\frac{\sum_{j=1}^J v^{(j)} f(\mathbf{Y}, \mathbf{Z}^{*(j)})}{\sum_{j=1}^J v^{(j)}}. \quad (4.1)$$

By the multivariate central limit theorem, the numerator and denominator of (4.1) are jointly asymptotically bivariate normal, so the delta method applied to the function $(u, v) \mapsto u/v$ gives asymptotic normality of (4.1), together with consistent estimates of the variance. (We note that the output supplied by `sis1` and `sis2` does not contain the column `se.sigma`.) With `ritcen`, we obtain iid observations directly from the posterior; no reweighting is required. Thus the Monte Carlo variability of any quantity can be estimated by using its sample variance.

In our functions, the default is to compute estimates only for the times t which occur in the data; for interval-censored data, estimates are supplied only for times which are end points of the censoring intervals. But all our functions accept the optional argument `adtim` which allows us to specify additional times at which estimates are to be computed. In fact, the output in Table 1 was actually computed using the option `adtim=c(2, 30, 42, 60)` in `gibbs1`.

The `plot` command in Figure 1 produces the plot given in Figure 2, which displays $\bar{\mu}(t)$ surrounded by a band given by the quantiles `lower` and `upper` described above. When the option `ci= β` is not used in the `gibbs1` function, `plot` will still display $\bar{\mu}(t)$ surrounded by a band, but this band's width is proportional to $\sigma(t)$. The width of the band is controlled by the option `mult` in the `plot` command; for a given value `mult=c`, the band displayed is $\bar{\mu}(t) \pm c\sigma(t)$. The default is `mult=1`.

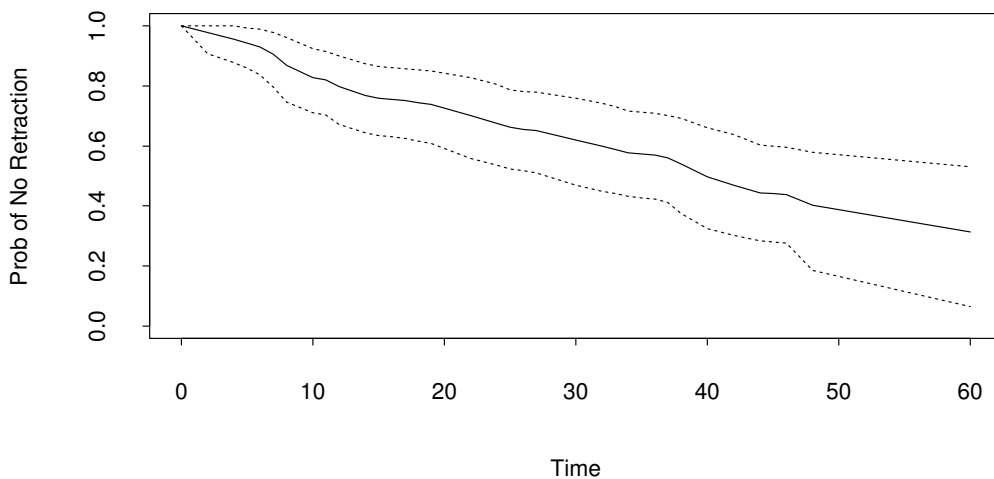


Figure 2: Output of `plot` command. The option `ci=.95` was used to generate the pointwise bands.

To illustrate the fact that estimators based on mixtures of Dirichlet processes tend to interpolate between the purely parametric and nonparametric models, we include Figure 3. This figure

displays the curves $\bar{\mu}(t)$ obtained from `gibbs1` applied to the breast cancer data `bc.dat.rad` for $M = .1, 10, \text{ and } 1000$. All three prior distributions take $(a, b) = (.1, .1)$. Superimposed on these curves is a plot of the nonparametric MLE computed using the `icfit` program in the package `interval` of Fay posted in the StatLib archive. The plot with $M = .1$ tracks the

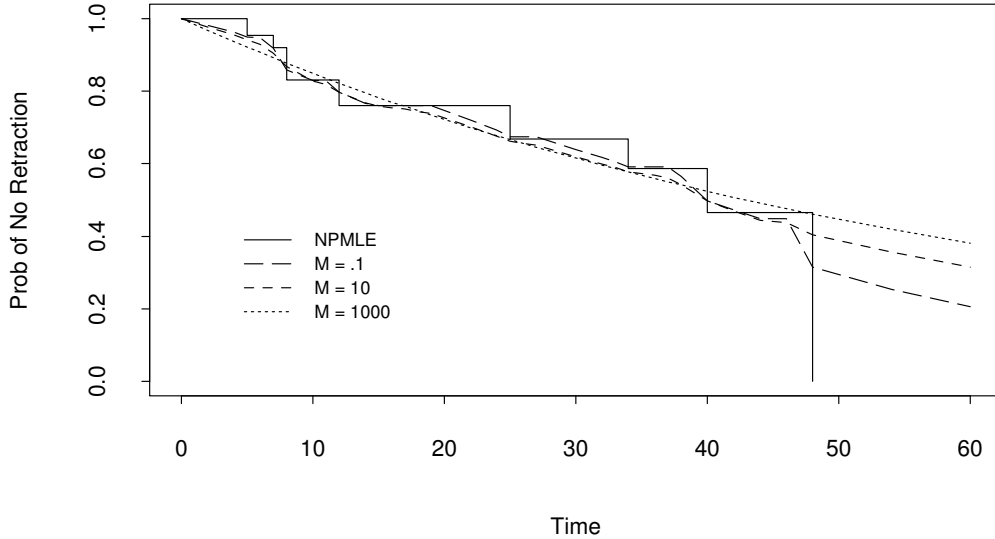


Figure 3: The NPMLE and three different Bayes estimates obtained using $M = .1$, $M = 10$, and $M = 1000$.

NPMLE fairly closely, and the plot with $M = 1000$ is a smooth function which is nearly indistinguishable from the parametric Bayesian estimate. The particular choice $(a, b) = (.1, .1)$ was motivated by the fact that, in the usual parametric Bayesian model without censoring, small values of a and b produce an approximation to the Jeffreys (“uninformative”) prior $d\theta/\theta$, and give the greatest agreement with the parametric MLE. However, we note that very similar results are obtained for a wide range of values (a, b) .

We may use our programs to compare the survival functions for the two treatment groups. We analyze the second group using the same prior $(a, b, M) = (2, 88, 10)$ used for the first group (Figure 2). These values are somewhat arbitrary, but do lead to a fairly dispersed prior on θ . (The conclusions we reach below are not very sensitive to the choice of a and b .) Figure 4 shows a plot of the resulting two probability bands, each of (pointwise) level .95.

This plot strongly suggests that retraction tends to occur earlier in the radiation plus chemotherapy group, a conclusion also reached by other authors (Finkelstein and Wolfe 1985, Klein and Moeschberger 1997, Fay 1999). (However, one must be cautious in interpreting this difference, since there may be confounding factors, as this is not a randomized trial. Beadle et al. (1984a) report that the two populations were dissimilar in the size of the primary tumor. We noted that data from Beadle et al. (1984b), which considers exclusively women treated with radiation only, shows that women with small tumors (0–2cm) have significantly better cosmetic effects than women with bigger tumors (2–5cm).)

This section is intended only as a brief introduction to our programs. A more complete description is given in a README file, distributed with the programs, all of which may be obtained

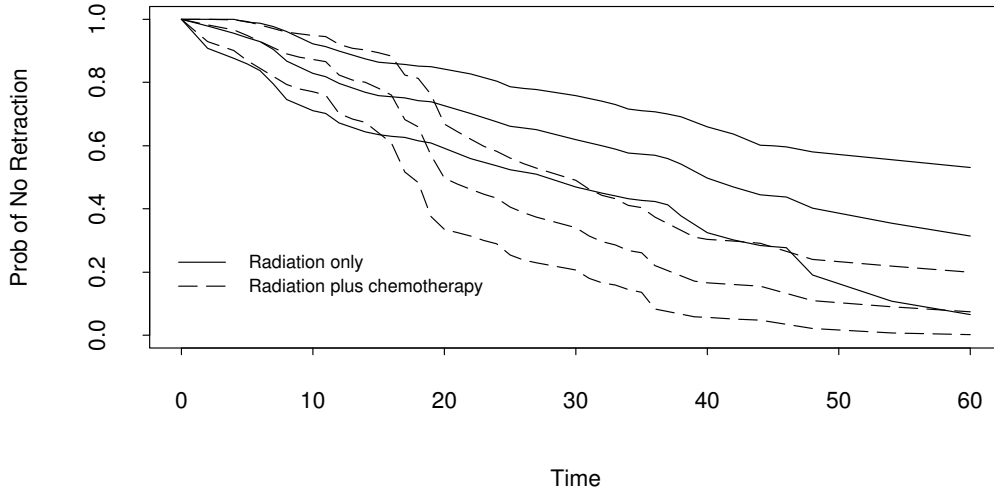


Figure 4: Comparison of distributions of retraction time for the two treatment groups.

electronically at <http://www.stat.ohio-state.edu/~huffer/npbayessurv>

5 Performance of the Algorithms

We conclude by comparing the performance of the algorithms `gibbs1`, `sis1`, `sis2` and `ritcen`. There are many comparisons we could make, but for simplicity, we shall compare the algorithms in terms of their Monte Carlo standard errors for $\bar{\mu}(t)$, the posterior mean of the survival function. We have carried out many simulation studies using both interval-censored and right-censored data. The studies reported below in Tables 2–4 are fairly representative. Tables 2 and 3 involve the two treatment groups of the breast cancer data discussed earlier. These data are interval censored. Table 2 presents the results for the “radiation only” group, and Table 3 for the “radiation plus chemotherapy” group. Table 4 involves the treatment group of the “Gehan data” (Gehan 1965). (To access this in Splus, we first invoke `library(MASS, first=T)` then follow with `gehan`.) These data are right censored.

For each data set, we display results for 10 different priors (a, b, M) which use two different choices for (a, b) and five values of M ranging from 0.1 to 1000. For each prior, we compare the various algorithms at three different time points t at which the posterior mean $\bar{\mu}(t)$ is approximately .90, .50, and .10, respectively. For convenience in the presentation, in each table we have selected one of the algorithms to serve as a reference. In Tables 2 and 3 this reference algorithm is `gibbs1`; in Table 4 it is `ritcen`. For each time point t , we report the standard error of $\bar{\mu}(t)$ for the reference algorithm, and for the other algorithms we give the ratio of the standard error with that of the reference algorithm. All the simulations use a sample size of 1,000,000. The standard errors reported for the reference algorithm have been multiplied by 10 to give values that would be attained with a sample size of 10,000, which might be a typical sample size used in applications. In the tables, the algorithm names `gibbs1`, `sis1`, `sis2`, and `ritcen` have been abbreviated as `g1`, `s1`, `s2`, and `rc`, respectively.

The `gibbs1` program has options which allow the user to specify how often to sample from

the chain, and how often the extra step in Section 2.5, and the associated Rao-Blackwellization in Section 3, are carried out. Determining the best possible values for these options may involve considerable experimentation. To keep things simple, in all our simulations we insert the extra step and the associated Rao-Blackwellization, and also sample the output of the chain after every repetition of the Gibbs sampler in Section 2.3. The Monte Carlo standard errors for `gibbs1` were estimated by dividing the sample of 1,000,000 into 500 batches.

M	$\bar{\mu}(t)$	$(a, b) = (2, 88)$			$(a, b) = (.1, .1)$		
		SE of g1	g1/s1	g1/s2	SE of g1	g1/s1	g1/s2
1000	.90	3.4e-4	1.5	0.094	3.8e-4	0.55	0.17
1000	.50	1.3e-3	1.5	0.085	1.4e-3	0.55	0.16
1000	.10	8.9e-4	1.5	0.065	9.6e-4	0.56	0.14
100	.90	3.3e-4	1.7	0.21	3.6e-4	0.61	0.14
100	.50	1.2e-3	1.7	0.19	1.3e-3	0.62	0.12
100	.10	9.9e-4	1.8	0.17	1.1e-3	0.65	0.098
10	.90	3.2e-4	1.1	0.51	3.3e-4	0.34	0.37
10	.50	6.6e-4	1.0	0.49	7.2e-4	0.36	0.35
10	.10	1.1e-3	1.5	0.37	1.5e-3	0.60	0.29
1	.90	6.1e-4	0.52	0.74	6.3e-4	0.16	0.57
1	.50	9.2e-4	0.35	0.49	9.1e-4	0.10	0.23
1	.10	1.4e-3	0.31	0.46	1.8e-3	0.11	0.29
0.1	.90	8.0e-4	0.19	0.25	7.6e-4	0.071	0.13
0.1	.50	1.0e-3	0.11	0.17	1.0e-3	0.02	0.044
0.1	.10	1.6e-3	0.095	0.14	1.8e-3	0.033	0.033

Table 2: Simulations on `bcdat.rad`, the “radiation only” group of the breast cancer data. The columns “SE of g1” report the Monte Carlo standard errors for `gibbs1`. The columns `g1/s1` and `g1/s2` give ratios of standard errors.

For interval-censored data, the results in Tables 2 and 3 show that none of the algorithms `gibbs1`, `sis1` or `sis2` uniformly dominates the others, and that the relative performance of the algorithms depends greatly on the data set and choice of prior. Although our tables do not address this point, we note that the performance of the SIS algorithms also depends greatly on the particular ordering of the data. The default in `sis1` and `sis2` (which was used in our simulations) is to order the data according to the value of $P(X_i \in A_i)$, the prior probability assigned to A_i , from smallest to largest. Based on simulation experiments reported in Tables 2 and 3 and others like them, we recommend `gibbs1` for general use with interval censored data. It is not uniformly superior to the other algorithms, but never performs much worse. On the other hand, the SIS algorithms sometimes do extremely badly relative to `gibbs1`. In particular, the relative performance of the SIS algorithms deteriorates as $M \rightarrow 0$.

Even though we recommend `gibbs1` for general use, we feel that users should not be reluctant to try out any of the algorithms. When an algorithm is doing badly, this will generally be clear from the estimated Monte Carlo standard errors, and the user can simply switch to a different algorithm. To aid in judging the performance of the SIS algorithms, our program output includes the “effective sample size” (ESS) described in Kong, Liu, and Wong (1994).

M	$\bar{\mu}(t)$	$(a, b) = (2, 88)$			$(a, b) = (.1, .1)$		
		SE of g1	g1/s1	g1/s2	SE of g1	g1/s1	g1/s2
1000	.90	2.1e-4	0.94	1.1	2.2e-4	0.36	1.1
1000	.50	7.4e-4	0.94	1.1	7.8e-4	0.36	1.1
1000	.10	5.0e-4	0.97	1.0	5.4e-4	0.37	1.0
100	.90	2.2e-4	0.97	1.1	2.3e-4	0.37	1.0
100	.50	6.2e-4	0.95	1.0	6.5e-4	0.37	1.0
100	.10	5.2e-4	1.0	0.97	5.3e-4	0.38	0.94
10	.90	1.3e-4	0.41	0.50	1.6e-4	0.19	0.60
10	.50	7.1e-4	0.49	0.70	6.8e-4	0.19	0.75
10	.10	4.7e-4	0.44	0.55	5.1e-4	0.20	0.62
1	.90	1.1e-4	0.06	0.085	1.2e-4	0.026	0.11
1	.50	1.2e-3	0.12	0.12	1.3e-3	0.047	0.17
1	.10	5.8e-4	0.079	0.089	5.9e-4	0.023	0.12
0.1	.90	1.1e-4	0.027	0.022	1.0e-4	0.019	0.033
0.1	.50	1.5e-3	0.071	0.079	1.4e-3	0.032	0.065
0.1	.10	6.8e-4	0.047	0.055	6.5e-4	0.033	0.057

Table 3: Simulations on the “radiation plus chemotherapy” group of the breast cancer data.

M	$\bar{\mu}(t)$	$(a, b) = (2, 80)$				$(a, b) = (0.1, 0.1)$			
		SE of rc	rc/g1	rc/s1	rc/s2	SE of rc	rc/g1	rc/s1	rc/s2
1000	.90	2.8e-4	0.56	1.0	0.53	3.1e-4	0.54	0.37	0.33
1000	.50	1.0e-3	0.56	0.99	0.48	1.1e-3	0.53	0.36	0.29
1000	.10	7.0e-4	0.55	1.0	0.38	7.7e-4	0.51	0.39	0.21
100	.90	2.9e-4	0.57	1.0	0.58	3.1e-4	0.55	0.37	0.39
100	.50	8.9e-4	0.57	0.99	0.53	9.8e-4	0.53	0.37	0.34
100	.10	6.7e-4	0.56	1.0	0.42	7.4e-4	0.52	0.39	0.26
10	.90	1.7e-4	0.65	0.94	0.83	2.1e-4	0.55	0.40	0.71
10	.50	5.2e-4	0.68	0.84	0.81	5.8e-4	0.61	0.34	0.67
10	.10	5.7e-4	0.65	0.99	0.69	6.3e-4	0.55	0.40	0.54
1	.90	2.9e-5	0.76	0.67	0.99	3.8e-5	0.81	0.39	0.99
1	.50	5.9e-4	0.85	0.58	1.0	6.0e-4	0.87	0.28	1.0
1	.10	4.9e-4	0.78	0.73	0.98	5.4e-4	0.80	0.38	0.97
0.1	.90	3.1e-6	0.91	0.61	1.0	4.0e-6	0.89	0.38	1.0
0.1	.50	6.4e-4	0.80	0.54	1.0	6.4e-4	0.81	0.27	1.0
0.1	.10	4.8e-4	0.90	0.67	1.0	5.2e-4	0.89	0.37	1.0

Table 4: Simulations on the treatment group of the “Gehan data”. The columns “SE of rc” report the Monte Carlo standard errors for `ritcen`. The columns `rc/g1`, `rc/s1` and `rc/s2` give ratios of standard errors.

The `ritcen` algorithm is the clear choice for use with right censored data since it produces an iid sample from the posterior. The superiority of `ritcen` is borne out in simulation experiments like those in Table 4. Another fact illustrated by Table 4 is that, if the right censored data is ordered appropriately, the performance of `sis2` improves as $M \rightarrow 0$. We have verified this

empirically, but it also follows from the bounds (2.21) for the predictive probabilities. For small M , because the v_i 's are close to 1, the weights w_j are nearly equal, and so `sis2` gives nearly an iid sample from the posterior.

Two questions are not addressed in the simulations described so far. First, how much is the `gibbs1` algorithm improved by incorporating the extra step of Section 2.5? Secondly, how well does `gibbs1` do relative to iid sampling? We have run various simulations to answer these questions, with some of the results summarized in Table 5. This table compares the performance of three variants of `gibbs1`. The first, denoted `g1` in Table 5, is just `gibbs1` with the options set as described earlier. This is the version of `gibbs1` studied in Tables 2–4. The second variant, denoted `nx`, omits the extra step. The third, denoted `id`, generates approximately iid observations from the posterior by sampling every 100th observation from the chain produced by `g1`. We ran these algorithms on the two groups of the breast cancer data and on the treatment group of the Gehan data using priors with $(a, b) = (0.1, 0.1)$ and $M = 1000, 10$, and 0.1 . For each data set and prior, we display ratios of standard errors at three different time points as in the earlier tables. For the Gehan data, we use the exact iid sampling method given by `ritcen` (denoted `rc`) instead of the approximately iid observations given by `id`. (Thus, the column `rc/g1` in Table 5 is just a subset of the corresponding column in Table 4.)

M	$\bar{\mu}(t)$	Rad Only		Rad + Chemo		Gehan Data	
		<code>g1/nx</code>	<code>id/g1</code>	<code>g1/nx</code>	<code>id/g1</code>	<code>g1/nx</code>	<code>rc/g1</code>
1000	.90	1.0	0.51	0.99	0.72	0.96	0.54
1000	.50	1.0	0.51	0.99	0.71	0.95	0.53
1000	.10	1.0	0.51	1.0	0.71	0.94	0.51
10	.90	0.79	0.61	0.96	0.72	0.86	0.55
10	.50	0.77	0.57	0.84	0.60	0.86	0.61
10	.10	0.80	0.41	0.94	0.68	0.83	0.55
0.1	.90	0.21	0.35	0.42	0.88	0.17	0.89
0.1	.50	0.12	0.60	0.11	0.42	0.92	0.81
0.1	.10	0.11	0.42	0.61	0.62	0.17	0.89

Table 5: Ratios of standard errors for three variants of `gibbs1` with the prior $(a, b) = (0.1, 0.1)$ and various values of M .

Table 5 illustrates the typical behavior of the `gibbs1` algorithm. For interval censored data, the extra step of Section 2.5 often dramatically improves the performance of the algorithm when M is small, but usually has little or no benefit for larger values of M . For right-censored data, there is very little gain from using the extra step when M is small (in sharp contrast to the situations described in West, Müller, and Escobar (1994) and in Bush and MacEachern (1996)). The reason is that then most of the values X_i are equal to one of the observed death times, and these are not moved by the extra step. For example, for the Gehan data with $M = 0.1$, the extra step helps only in the tails (such as when $\bar{\mu}(t) = .10$ or $.90$) at values of t which lie outside the observed data times. The extra step has little effect in the central part of the distribution.

Perhaps the most interesting piece of information to be gathered from Table 5 is that `g1` performs quite well relative to iid sampling for a broad range of values of M .

Finally, we comment briefly on the relative speed of the algorithms. The algorithms `gibbs1`,

`sis1`, and `sis2` have roughly equal running times per iteration. For right-censored data, `ritcen` is substantially faster than the other algorithms. To illustrate this, Table 6 gives approximate computation times (without the `ci` option) for a sample size of 100,000 in two of the situations studied in Tables 2 and 4. These timings were done on a machine doing about 16 specFP's.

bcdat.rad with $(a, b, M) = (2, 88, 10)$			Gehan Data with $(a, b, M) = (2, 80, 10)$			
g1	s1	s2	g1	s1	s2	rc
58.7	51.4	57.1	28.6	24.8	27.1	15.3

Table 6: Approximate computation times in seconds for a sample of 100,000.

Acknowledgments

We are grateful for constructive criticism from extremely conscientious referees.

References

- Antoniak, C. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics* **2** 1152–74.
- Athreya, K.B., Doss, H., and Sethuraman, J. (1996). On the convergence of the Markov chain simulation method. *The Annals of Statistics* **24** 69–100.
- Beadle, G., Come, S., Henderson, C., Silver, B., and Hellman, S. (1984a). The effect of adjuvant chemotherapy on the cosmetic results after primary radiation treatment for early stage breast cancer. *International Journal of Radiation Oncology, Biology and Physics* **10** 2131–2137.
- Beadle, G., Harris, J., Silver, B., Botnick, L., and Hellman, S. (1984b). Cosmetic results following primary radiation therapy for early breast cancer. *Cancer* **54** 2911–2918.
- Blackwell, D. and MacQueen J.B. (1973). Ferguson distributions via Pólya urn schemes. *The Annals of Statistics* **1** 353–355.
- Bush, C. and MacEachern, S.N. (1996). A semi-parametric Bayesian model for randomized block designs. *Biometrika* **83** 275–285.
- Doksum, K. (1974). Tailfree and neutral random probabilities and their posterior distributions. *The Annals of Probability* **2** 183–201.
- Doss, H. (1994). Bayesian nonparametric estimation for incomplete data via successive substitution sampling. *The Annals of Statistics* **22** 1763–1786.
- Doss, H. and Huffer, F. (2000). Monte Carlo methods for Bayesian analysis of survival data using mixtures of Dirichlet priors. Technical report, Department of Statistics, Florida State University.

- Escobar, M.D. (1988). Estimating the means of several normal populations by nonparametric estimation of the distribution of the means. Unpublished Ph.D. dissertation, Yale University, Department of Statistics.
- Escobar, M.D. (1994). Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association* **89** 268–277.
- Escobar, M.D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association* **90** 577–588.
- Escobar, M.D. and West, M. (1998). Computing Bayesian nonparametric hierarchical models. In *Practical Nonparametric and Semiparametric Bayesian Statistics* (D. Dey, P. Müller, and D. Sinha, eds.) 63–87, Springer-Verlag, New York.
- Gehan, E.A. (1965). A generalized Wilcoxon test for comparing arbitrarily single-censored samples. *Biometrika* **52** 203–233.
- Fay, M.P. (1999). Comparing several score tests for interval censored data. *Statistics in Medicine* **18** 273–285.
- Ferguson, T.S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics* **1** 209–230.
- Ferguson, T.S. (1974). Prior distributions on spaces of probability measures. *The Annals of Statistics* **2** 615–629.
- Ferguson, T.S. and Phadia (1979). Bayesian nonparametric estimation based on censored data. *The Annals of Statistics* **7** 163–186.
- Finkelstein, D.M. and Wolfe, R.A. (1985). A semiparametric model for regression analysis of interval-censored failure time data. *Biometrics* **41** 933–945.
- Hanson, T.E. and Johnson, W.O. (2001). A Bayesian semiparametric AFT model for interval censored data. Preprint.
- Ibrahim, J.G., Chen, M.-H., and Sinha, D. (2001). *Bayesian Survival Analysis*, Springer-Verlag, New York.
- Klein, J. and Moeschberger, M. (1997). *Survival Analysis*, Springer-Verlag, New York.
- Kong, A., Liu, J.S., and Wong, W.H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association* **89** 278–288.
- Kuo, L. and Smith, A.F.M. (1992). Bayesian computations in survival models via the Gibbs sampler. In *Survival Analysis: State of the Art* (J.P. Klein and P.K. Goel, eds.) 11–24, Kluwer Academic, Boston.
- Liu, J.S. (1996). Nonparametric hierarchical Bayes via sequential imputations. *The Annals of Statistics* **24** 911–930.
- MacEachern, S.N. (1994). Estimating normal means with a conjugate style Dirichlet process prior. *Communications in Statistics, Part B—Simulation and Computation* **23** 727–741.
- MacEachern, S.N., Clyde, M., and Liu, J.S. (1999). Sequential importance sampling for nonparametric Bayes models: The next generation. *Canadian Journal of Statistics* **27** 251–267.
- Neal, R.M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics* **9** 249–265.

- Newton, M.A. and Zhang, Y. (1999). A recursive algorithm for nonparametric analysis with missing data. *Biometrika* **86** 15–26.
- Ripley, B.D. (1987). *Stochastic Simulation*. Wiley, New York.
- Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica* **4** 639–650.
- Sinha, D. and Dey, D. (1997). Semiparametric Bayesian analysis of survival data. *Journal of the American Statistical Association* **92** 1195–1212.
- Susarla, V. and Van Ryzin, J. (1976). Nonparametric Bayesian estimation of survival curves from incomplete observations. *Journal of the American Statistical Association* **71** 897–902.
- Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics* **22** 1701–1727.
- Turnbull, B.W. (1974). Nonparametric estimation of a survivorship function with doubly censored data. *Journal of the American Statistical Association* **69** 169–173.
- Turnbull, B.W. (1976). The empirical distribution function with arbitrarily grouped, censored, and truncated data. *Journal of the Royal Statistical Society, Series B* **38** 290–295.
- West, M., Müller, P., and Escobar, M.D. (1994). Hierarchical priors and mixture models, with application in regression and density estimation. In *Aspects of Uncertainty: A Tribute to D.V. Lindley* (P. Freeman and A.F.M. Smith, eds.) 363–386, Wiley, New York.