Supplement to "Modelling county level breast cancer survival data using a covariate-adjusted frailty proportional hazards model"

Haiming Zhou[1], Timothy Hanson[‡1], Alejandro Jara[2], and Jiajia Zhang[3]

[1]*Department of Statistics, University of South Carolina, Columbia SC 29208, USA*

[2]*Department of Statistics, Pontificia Universidad Católica de Chile, Santiago, Chile*

[3]*Department of Epidemiology and Biostatistics, University of South Carolina, Columbia SC 29208, USA*

[‡]Electronic address: `hansont@stat.sc.edu`; Corresponding author

Appendix A: Mixtures of linear dependent Tailfree processes

A tailfree process is a stochastic process with trajectories on a given space of probability distributions. Tailfree processes generalize Polya tree processes, which further generalize Dirichlet processes. [4] and [2] introduced and developed tailfree processes. [3] summarized, extended, and made connections among these developments.

Let G_θ be a family of distribution functions on $\mathbb{R}$, indexed by $\theta \in \Theta$, serving to center the random probability measure G. A tailfree random distribution G with support on the real line is defined by allocations of conditional probabilities to increasingly refined partitions of $\mathbb{R}$. Let $E^m = \{\epsilon_1 \cdots \epsilon_m : \epsilon_i = 0, 1; i = 1 \ldots, m\}$ be the m-fold Cartesian product for all $m \in \mathbb{N}$ except $E^0 = \{\emptyset\}$, and set $E_j = \bigcup_{m=0}^{j} E^m$ for every $j = 1, 2, \ldots, J$, where J is a positive integer. For convenience, we use the convention that $\boldsymbol{\epsilon}0 = 0$ and $\boldsymbol{\epsilon}1 = 1$ for $\boldsymbol{\epsilon} = \emptyset$. At each level $j \in \{1, 2, \ldots, J\}$, we partition $\mathbb{R}$ into 2^j sets $\pi_\theta^j = \{B_\theta(\boldsymbol{\epsilon}) : \boldsymbol{\epsilon} \in E^j\}$, where $B_\theta(\boldsymbol{\epsilon}) = (G_\theta^{-1}(k/2^j), G_\theta^{-1}([k+1]/2^j)]$, with G_θ^{-1} being the quantile function of G_θ, and k being the decimal representation of $\boldsymbol{\epsilon} = \epsilon_1 \cdots \epsilon_j \in E^j$. For $\boldsymbol{\epsilon} = \emptyset$, define $B_\theta(\boldsymbol{\epsilon}) = \mathbb{R}$. These partition sets produce dyadic splits: $B_\theta(\boldsymbol{\epsilon}) = B_\theta(\boldsymbol{\epsilon}0) \cup B_\theta(\boldsymbol{\epsilon}1)$, for $\boldsymbol{\epsilon} \in E_{J-1}$. For each partition set at level j, indexed by the binary number $\boldsymbol{\epsilon} \in E^j$, a tailfree construction assigns two random conditional probabilities, $Y_{\boldsymbol{\epsilon}0} = G\{B_\theta(\boldsymbol{\epsilon}0)|B_\theta(\boldsymbol{\epsilon})\}$ and $Y_{\boldsymbol{\epsilon}1} = 1 - Y_{\boldsymbol{\epsilon}0} = G\{B_\theta(\boldsymbol{\epsilon}1)|B_\theta(\boldsymbol{\epsilon})\}$, to the offspring sets of $B_\theta(\boldsymbol{\epsilon})$. The collections $\{Y_0\}$, $\{Y_{00}, Y_{10}\}$,$\{Y_{000}, Y_{010}, Y_{100}, Y_{110}\}, \ldots,$ are mutually independent. Assume that G follows G_θ on sets in the finest partition π_θ^J and note that $G(B_\theta(\boldsymbol{\epsilon})) = \prod_{i=1}^{j} Y_{\epsilon_1 \cdots \epsilon_i}$ for every $\boldsymbol{\epsilon} \in E_J$. Given the J partitions $\Pi^{J,\theta} = \bigcup_{j=1}^{J} \pi_\theta^j$, a particular finite tailfree process is defined in Table 1 for $j = 1, 2, 3$.

Table 1: Principle scheme to define a tailfree process centered around G_θ

<table>
<tr><td colspan="8">$\mathbb{R}$</td></tr>
<tr><td colspan="4">$B_\theta(0)$
$Y_0 = G\{B_\theta(0)|\mathbb{R}\}$</td><td colspan="4">$B_\theta(1)$
$Y_1 = G\{B_\theta(1)|\mathbb{R}\}$</td></tr>
<tr><td colspan="2">$B_\theta(00)$
$Y_{00} = G\{B_\theta(0)|B_\theta(0)\}$</td><td colspan="2">$B_\theta(01)$
$Y_{01} = G\{B_\theta(01)|B_\theta(0)\}$</td><td colspan="2">$B_\theta(10)$
$Y_{10} = G\{B_\theta(10)|B_\theta(1)\}$</td><td colspan="2">$B_\theta(11)$
$Y_{11} = G\{B_\theta(11)|B_\theta(1)\}$</td></tr>
<tr><td>$B_\theta(000)$
Y_{000}</td><td>$B_\theta(001)$
Y_{001}</td><td>$B_\theta(010)$
Y_{010}</td><td>$B_\theta(011)$
Y_{011}</td><td>$B_\theta(100)$
Y_{100}</td><td>$B_\theta(101)$
Y_{101}</td><td>$B_\theta(110)$
Y_{110}</td><td>$B_\theta(111)$
Y_{111}</td></tr>
</table>

A Polya tree prior is defined by assigning to the conditional probabilities $\{Y_{\boldsymbol{\epsilon}0}\}_{\boldsymbol{\epsilon} \in E_{J-1}}$ independent beta distributions, that is, $Y_{\boldsymbol{\epsilon}0} \overset{ind.}{\sim} \text{Beta}(cd_{\boldsymbol{\epsilon}0}^2, cd_{\boldsymbol{\epsilon}0}^2)$, where $c > 0$ is the precision parameter and $d_{\boldsymbol{\epsilon}0}$ is the length of $\boldsymbol{\epsilon}0$. To allow the entire shape of the frailty distribution to vary smoothly with covariates, we instead consider logistic-normal conditional probabilities that closely follow beta distributions yielding a particular class of dependent tailfree processes [8]. Let $\mathbf{X} = [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n]'$ be the $n \times (q+1)$ design matrix, where each $\tilde{\mathbf{x}}_i = (1, \mathbf{x}_i')'$. Given

the cluster-level covariate values $\mathbf{x}$ in cluster, a tailfree random distribution $G_{\mathbf{x}}$ is defined by replacing the above independent beta random probabilities $\{Y_{\boldsymbol{\epsilon}0}\}_{\boldsymbol{\epsilon}\in E_{J-1}}$ with the following logistic-normal random variables

$$Y_{\boldsymbol{\epsilon}0}(\mathbf{x},\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}) = h\left(\tilde{\mathbf{x}}'\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}\right) = \frac{\exp(\tilde{\mathbf{x}}'\boldsymbol{\beta}_{\boldsymbol{\epsilon}0})}{1+\exp(\tilde{\mathbf{x}}'\boldsymbol{\beta}_{\boldsymbol{\epsilon}0})}, \quad Y_{\boldsymbol{\epsilon}1}(\mathbf{x},\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}) = 1-h\left(\tilde{\mathbf{x}}'\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}\right) = \frac{1}{1+\exp(\tilde{\mathbf{x}}'\boldsymbol{\beta}_{\boldsymbol{\epsilon}0})},$$

where $\boldsymbol{\beta}_{\boldsymbol{\epsilon}0} \overset{ind}{\sim} N_{q+1}\left(\mathbf{0}, 2n(\mathbf{X}'\mathbf{X})^{-1}/(c\rho(d_{\boldsymbol{\epsilon}0}))\right)$, with $\rho(d_{\boldsymbol{\epsilon}0}) = d_{\boldsymbol{\epsilon}0}^2$, following so called g-priors [16]. The resulting process $\{G_{\mathbf{x}} : \mathbf{x}\in\mathcal{X}\}$ is referred to as a *p*artially specified linear dependent tailfree process with parameters $(h, \Pi^{J,\theta}, \mathcal{A}^{J,c,\rho})$. It is easy to see that $E\{Y_{\boldsymbol{\epsilon}0}(\mathbf{x},\boldsymbol{\beta}_{\boldsymbol{\epsilon}0})|\mathbf{x}\} = 0.5$, so $E(G_{\mathbf{x}}) = G_\theta$, namely the random distribution $G_{\mathbf{x}}$ is centered around G_θ for all $\mathbf{x}\in\mathcal{X}$.

For notational simplicity, set $Y_{\boldsymbol{\epsilon}0}(i) = Y_{\boldsymbol{\epsilon}0}(\mathbf{x}_i,\boldsymbol{\beta}_{\boldsymbol{\epsilon}0})$ and $\eta_{\boldsymbol{\epsilon}0}(i) = \tilde{\mathbf{x}}_i'\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}$. Define $\boldsymbol{\eta}_{\boldsymbol{\epsilon}0} = (\eta_{\boldsymbol{\epsilon}0}(1),\ldots,\eta_{\boldsymbol{\epsilon}0}(n))'$. Based on the g-priors of $\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}$, we have $\boldsymbol{\eta}_{\boldsymbol{\epsilon}0}|c \sim N_n(\mathbf{0}, g\mathbf{M})$ where $g = 2/(c\rho(d_{\boldsymbol{\epsilon}0}))$, with $\rho(d_{\boldsymbol{\epsilon}0}) = d_{\boldsymbol{\epsilon}0}^2$, and $\mathbf{M} = n\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$. This specification implies the logit conditional probabilities are positively correlated for the clusters with covariates that are close to each other. Let $\bar{\mathbf{x}} = \sum_{i=1}^n \mathbf{x}_i$ and $(n-1)\mathbf{S} = \sum_{i=1}^n(\mathbf{x}_i-\bar{\mathbf{x}})(\mathbf{x}_i-\bar{\mathbf{x}})'$. [1] notes that the ith diagonal element of $\mathbf{M}$ is given by $m_{ii} = 1 + nd_{ii}/(n-1)$, where $d_{ii} = (\mathbf{x}_i-\bar{\mathbf{x}})'\mathbf{S}^{-1}(\mathbf{x}_i-\bar{\mathbf{x}})$ is the sample squared Mahalanobis distance between $\mathbf{x}_i$ and $\bar{\mathbf{x}}$. For cluster-level covariate values $\mathbf{x}_{i_1}$, $\mathbf{x}_{i_2}$ reasonably close to the sample mean $\bar{\mathbf{x}}$ (in terms of the Mahalanobis distance), a first order approximation gives that $(Y_{\boldsymbol{\epsilon}0}(i_1), Y_{\boldsymbol{\epsilon}0}(i_2))'$ approximately follows $N_2(\mathbf{0}, g\mathbf{X}_0(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_0)$, where $\mathbf{X}_0 = (\mathbf{x}_{i_1}, \mathbf{x}_{i_2})'$. The larger c or $d_{\boldsymbol{\epsilon}0}$ are, the better this approximation becomes. This leads to

$$\mathrm{corr}\{Y_{\boldsymbol{\epsilon}0}(i_1), Y_{\boldsymbol{\epsilon}0}(i_2)\} \approx \frac{1+d_{i_1,i_2}}{\sqrt{1+d_{i_1,i_1}}\sqrt{1+d_{i_2,i_2}}},$$

where $d_{i,j} = (\mathbf{x}_i-\bar{\mathbf{x}})'\mathbf{S}^{-1}(\mathbf{x}_j-\bar{\mathbf{x}})$. Note that the locations $\mathbf{x}_{i_1}$ and $\mathbf{x}_{i_2}$ relative to the mean covariate vector $\bar{\mathbf{x}}$ play a role in the correlation, and not just the Mahalanobis distance between them. When the covariate vector for a cluster i is fixed at $\mathbf{x}_i$, [8] show that assuming $\eta_{\boldsymbol{\epsilon}0}(i) \sim N(0, 2m_{ii}/cd_{\boldsymbol{\epsilon}0}^2)$ approximates $Y_{\boldsymbol{\epsilon}0}(i) \sim \mathrm{Beta}(c_i d_{\boldsymbol{\epsilon}0}^2, c_i d_{\boldsymbol{\epsilon}0}^2)$, where $c_i = cm_{ii}^{-1}$. This approximation explains the similarity between a marginal realization $G_{\mathbf{x}_i}$ of the linear-dependent tailfree process and a "traditional" Polya tree prior.

The precision parameter $c\in\mathbb{R}^+$ controls how closely the random distribution $G_{\mathbf{x}}$ follows G_θ in terms of L_1 distance [6]. Large values of c indicate a strong belief that the frailties are closely *iid* from G_θ, since as c tends to ∞, the random distribution $G_{\mathbf{x}}$ is G_θ with probability 1. Smaller values of c, on the other hand, allow more pronounced deviations of G from G_θ. The choice $c=1$ has been advocated by many authors, e.g. recently [13]. Alternatively, we can also put a gamma prior on c, say, $c\sim\Gamma(a_c, b_c)$, as suggested in [8]; if that is the case,

the full conditional distribution for c is also a gamma distribution. As for the choice of J, we typically consider $J \approx \log_2(n/n_0)$, where n is the sample size and n_0 (usually from 5 to 10) is a "typical" number of observations falling into each set at level J [7]. In addition, the linear dependent tailfree process depends on the partition $\Pi^{J,\theta}$ which is further determined by the centering distribution G_θ. If one simply fixes θ at a specific value, the posterior inferences may be affected unduly due to the partition dependence. In practice, one common strategy to mitigate this problem is to specify a mixture of LDTFP by allowing parameter θ of the centering distribution to be random, that is,

$$\{G_{\mathbf{x}} : \mathbf{x} \in \mathcal{X}\} \sim \text{LDTFP}(h, \Pi^{J,\theta}, \mathcal{A}^{J,c,\rho}), \quad \theta \sim P(d\theta),$$

where $P(d\theta)$ represents a prior for θ; in this article, we consider $\theta^{-2} \sim \Gamma(\tau_1, \tau_2)$, where Γ refers to the gamma distribution with shape τ_1 and rate τ_2.

Given θ and $\boldsymbol{\beta} = \{\boldsymbol{\beta}_{\boldsymbol{\epsilon}0} : \boldsymbol{\epsilon} \in E_{J-1}\}$, the conditional density of e_i, given cluster-level predictors $\mathbf{x}_i$, is given by

$$g(e_i|\theta, \boldsymbol{\beta}) = 2^J \phi_\theta(e_i) \prod_{j=1}^{J} Y_{\boldsymbol{\epsilon}_\theta(e_i,j)}(\mathbf{x}_i, \boldsymbol{\beta}_{\boldsymbol{\epsilon}_\theta(e_i,j-1)0}), \tag{A.1.1}$$

where $\phi_\theta(\cdot)$ is the density of a $N(0, \theta^2)$ variate and $\boldsymbol{\epsilon}_\theta(e_i, j) = \epsilon_1\epsilon_2 \cdots \epsilon_j$ is the set in π_θ^j that e_i is in. Therefore, the joint density of the frailty terms is given by

$$p(e_1, \ldots, e_n|\theta, \boldsymbol{\beta}) = \left[\prod_{i=1}^{n} 2^J \phi_\theta(e_i)\right] \left[\prod_{\boldsymbol{\epsilon} \in E_{J-1}} \prod_{i=1}^{n} \frac{\exp(\tilde{\mathbf{x}}_i' \boldsymbol{\beta}_{\boldsymbol{\epsilon}0})^{I\{e_i \in B_\theta(\boldsymbol{\epsilon}0)\}}}{[1 + \exp(\tilde{\mathbf{x}}_i' \boldsymbol{\beta}_{\boldsymbol{\epsilon}0})]^{I\{e_i \in B_\theta(\boldsymbol{\epsilon})\}}}\right], \tag{A.1.2}$$

where $I\{A\}$ is the indicator function for A. This expression has the form of $2^J - 1$ logistic regression kernels, one for each $\boldsymbol{\epsilon}0$, times the likelihood for θ obtained from fitting the standard parametric family $N(0, \theta^2)$ to data $\mathbf{e} = (e_1, \ldots, e_n)$. This forms the basis of an efficient MCMC scheme for obtaining posterior inference.

Appendix B: MCMC details

In this appendix, we describe an efficient MCMC algorithm for obtaining a posterior sample $\{(\boldsymbol{\gamma}^{(s)}, \mathbf{e}^{(s)}, \theta^{(s)}, \boldsymbol{\beta}^{(s)})\}_{s=1}^{S}$. Based on this random sample, we can obtain any posterior target of inference of interest. For instance, the covariate effect $\boldsymbol{\xi}$ is estimated by the posterior mean $\hat{\boldsymbol{\xi}} = \mathcal{S}^{-1}\sum_{s=1}^{\mathcal{S}} \boldsymbol{\xi}^{(s)}$; the centering parameter θ^2 is estimated by the $\hat{\theta}^2 = \mathcal{S}^{-1}\sum_{s=1}^{\mathcal{S}} \theta^{2(s)}$; the predictive frailty density given some specific cluster-level covariate $\mathbf{x}$ is estimated by $\hat{g}(e|\mathbf{x}) = \mathcal{S}^{-1}\sum_{s=1}^{\mathcal{S}} g(e|\theta^{(s)}, \boldsymbol{\beta}^{(s)})$ with g given in (A.1.1); the predictive survival function given covariate vector $\mathbf{w}$ is estimated by

$$\hat{S}(t|\mathbf{w}) = \mathcal{S}^{-1}\sum_{s=1}^{\mathcal{S}}\sum_{k=1}^{\mathcal{K}} \exp\left\{-\sum_{k=1}^{K(t)} \lambda_k^{(s)}\Delta_k(t)\exp\left\{\mathbf{w}'\boldsymbol{\xi}^{(s)} + e_k^{(s)}\right\}\right\}, \tag{B.1}$$

where $\{e_k^{(s)}\}_{k=1}^{\mathcal{K}}$ is a random sample from the frailty density $g(\cdot|\theta^{(s)}, \boldsymbol{\beta}^{(s)})$.

Updating the frailties

Modifying [9], the random effects e_i can be updated as follows. For $i = 1, \ldots, n$, let $V_i(\boldsymbol{\gamma}, e_i) = [\theta^{-2} + \mathbf{1}'_{N_i}\mathbf{W}_i(\boldsymbol{\gamma}, e_i)\mathbf{1}_{N_i}]^{-1}$, where $\mathbf{1}_{N_i}$ is an $N_i \times 1$ vector of 1s and $\mathbf{W}_i(\boldsymbol{\gamma}, e_i) = \text{diag}(\mu_{ijk})$ is an $N_i \times N_i$ diagonal matrix. Then for the $(s+1)$th scan of the posterior distribution, the candidate $e_i^* \sim N(e_i^{(s)}, V_i(\boldsymbol{\gamma}, e_i^{(s)}))$ is accepted with probability

$$\alpha(e_i^*|e_i^{(s)}) = 1 \wedge \frac{L_i(\boldsymbol{\gamma}, e_i^*)g(e_i^*|\theta, \boldsymbol{\beta})\phi(e_i^{(s)}|e_i^*, V_i(\boldsymbol{\gamma}, e_i^*))}{L_i(\boldsymbol{\gamma}, e_i^{(s)})g(e_i^{(s)}|\theta, \boldsymbol{\beta})\phi(e_i^*|e_i^{(s)}, V_i(\boldsymbol{\gamma}, e_i^{(s)}))}, \tag{B.2}$$

where $L_i(\boldsymbol{\gamma}, e_i) = \prod_{j=1}^{n_i}\prod_{k=1}^{K(t_{ij})} p(y_{ijk}|\boldsymbol{\gamma}, e_i)$ and $\phi(\cdot|\mu, \sigma^2)$ represents the density function of a $N(\mu, \sigma^2)$ random variable.

Updating fixed effects parameters

Set $N = \sum_{i=1}^{n} N_i$, $\mathbf{y} = (\mathbf{y}'_1, \ldots, \mathbf{y}'_n)'$, $\boldsymbol{\mu}(\boldsymbol{\gamma}, \mathbf{e}) = E[\mathbf{y}|\boldsymbol{\gamma}, \mathbf{e}]$. Let $\mathbf{Z} = (\mathbf{z}'_{ijk})$ be an $N \times (K + p)$ design matrix and $\mathbf{W}(\boldsymbol{\gamma}, \mathbf{e}) = \text{diag}(\mu_{ijk})$ be an $N \times N$ diagonal matrix, where all subscripts ijk are in lexicographical order. Assume a normal prior for the fixed effects $\boldsymbol{\gamma} \sim N_{K+p}(\boldsymbol{\gamma}_0, \mathbf{S}_0)$. Let $V(\boldsymbol{\gamma}, \mathbf{e}) = [\mathbf{S}_0^{-1} + \mathbf{Z}'\mathbf{W}(\boldsymbol{\gamma}, \mathbf{e})\mathbf{Z}]^{-1}$ and

$$\mathbf{m}(\boldsymbol{\gamma}, \mathbf{e}) = \mathbf{V}(\boldsymbol{\gamma}, \mathbf{e})[\mathbf{S}_0^{-1}\boldsymbol{\gamma}_0 + \mathbf{Z}'\mathbf{W}(\boldsymbol{\gamma}, \mathbf{e})\mathbf{Z}\boldsymbol{\gamma} + \mathbf{Z}'(\mathbf{y} - \boldsymbol{\mu}(\boldsymbol{\gamma}, \mathbf{e}))].$$

Then for the $(s+1)$th scan of the posterior distribution, the candidate $\boldsymbol{\gamma}^*$ is generated from an $N_{K+p}(\mathbf{m}(\boldsymbol{\gamma}^{(s)}, \mathbf{e}), \mathbf{V}(\boldsymbol{\gamma}^{(s)}, \mathbf{e}))$ distribution with acceptance proability

$$\alpha(\boldsymbol{\gamma}^*|\boldsymbol{\gamma}^{(s)}) = 1 \wedge \frac{L(\boldsymbol{\gamma}^*, \mathbf{e})\phi_{K+p}(\boldsymbol{\gamma}^*|\boldsymbol{\gamma}_0, \mathbf{S}_0)\phi_{K+p}(\boldsymbol{\gamma}^{(s)}|\mathbf{m}(\boldsymbol{\gamma}^*, \mathbf{e}), \mathbf{V}(\boldsymbol{\gamma}^*, \mathbf{e}))}{L(\boldsymbol{\gamma}^{(s)}, \mathbf{e})\phi_{K+p}(\boldsymbol{\gamma}^{(s)}|\boldsymbol{\gamma}_0, \mathbf{S}_0)\phi_{K+p}(\boldsymbol{\gamma}^*|\mathbf{m}(\boldsymbol{\gamma}^{(s)}, \mathbf{e}), \mathbf{V}(\boldsymbol{\gamma}^{(s)}, \mathbf{e}))}, \tag{B.3}$$

where $L(\boldsymbol{\gamma}, \mathbf{e}) = \prod_{i=1}^n L_i(\boldsymbol{\gamma}, e_i)$ and $\phi_k(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents the density function of a $N_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ random vector.

Updating the Tailfree process coefficients

A proposal based on one step of the Newton-Raphson algorithm [15, 5] efficiently updates the parameters $\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}$. To maintain identifiability of the model we set the first level conditional probability $Y_0(\mathbf{x}_i, \boldsymbol{\beta}_0) = 0.5$, i.e. $\boldsymbol{\beta}_0 = \mathbf{0}$. It follows that we need to update $\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}$ for all $\boldsymbol{\epsilon} \in E_{J-1}\backslash\emptyset$ yielding $2^J - 2$ in total. Define "pseudodata"

$$\widetilde{e}_i(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}) = \tilde{\mathbf{x}}_i'\boldsymbol{\beta}_{\boldsymbol{\epsilon}0} + \frac{I\{e_i \in B_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}0)\} - Y_{\boldsymbol{\epsilon}0}(\mathbf{x}_i, \boldsymbol{\beta}_{\boldsymbol{\epsilon}0})}{Y_{\boldsymbol{\epsilon}0}(\mathbf{x}_i, \boldsymbol{\beta}_{\boldsymbol{\epsilon}0})[1 - Y_{\boldsymbol{\epsilon}0}(\mathbf{x}_i, \boldsymbol{\beta}_{\boldsymbol{\epsilon}0})]},$$

and weights

$$w_{ii}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}) = Y_{\boldsymbol{\epsilon}0}(\mathbf{x}_i, \boldsymbol{\beta}_{\boldsymbol{\epsilon}0})[1 - Y_{\boldsymbol{\epsilon}0}(\mathbf{x}_i, \boldsymbol{\beta}_{\boldsymbol{\epsilon}0})]I\{e_i \in B_\theta(\boldsymbol{\epsilon})\},$$

placed in the vector $\widetilde{\mathbf{e}}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}) = (\widetilde{e}_1(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}), \ldots, \widetilde{e}_n(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}))'$ and matrix

$$\mathbf{W}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}) = \mathrm{diag}(w_{11}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}), \ldots, w_{nn}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}))'.$$

Let $\mathbf{V}_{\boldsymbol{\epsilon}0} = \frac{2n}{cj^2}(\mathbf{X}'\mathbf{X})^{-1}$ for $\boldsymbol{\epsilon}0 \in E^j$, under the g-prior (7 in the paper) for $\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}$, the M-H proposal is $\boldsymbol{\beta}^*_{\boldsymbol{\epsilon}0} \sim N_{q+1}(\mathbf{m}(\boldsymbol{\beta}^{(s)}_{\boldsymbol{\epsilon}0}), \mathbf{C}(\boldsymbol{\beta}^{(s)}_{\boldsymbol{\epsilon}0}))$ where

$$\mathbf{m}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}) = \mathbf{C}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0})[\mathbf{X}'\mathbf{W}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0})\widetilde{\mathbf{e}}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0})] \text{ and } \mathbf{C}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}) = [\mathbf{V}^{-1}_{\boldsymbol{\epsilon}0} + \mathbf{X}'\mathbf{W}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0})\mathbf{X}]^{-1}. \tag{B.4}$$

This proposal is accepted with probability $\alpha(\boldsymbol{\beta}^*_{\boldsymbol{\epsilon}0}|\boldsymbol{\beta}^{(s)}_{\boldsymbol{\epsilon}0})$ defined by

$$\alpha(\boldsymbol{\beta}^*_{\boldsymbol{\epsilon}0}|\boldsymbol{\beta}^{(s)}_{\boldsymbol{\epsilon}0}) = 1 \wedge \left\{ \frac{\phi_{q+1}(\boldsymbol{\beta}^*_{\boldsymbol{\epsilon}0}|\mathbf{0}, \mathbf{V}_{\boldsymbol{\epsilon}0})}{\phi_{q+1}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}|\mathbf{0}, \mathbf{V}_{\boldsymbol{\epsilon}0})} \frac{\phi_{q+1}(\boldsymbol{\beta}^{(s)}_{\boldsymbol{\epsilon}0}|\mathbf{m}(\boldsymbol{\beta}^*_{\boldsymbol{\epsilon}0}), \mathbf{C}(\boldsymbol{\beta}^*_{\boldsymbol{\epsilon}0}))}{\phi_{q+1}(\boldsymbol{\beta}^*_{\boldsymbol{\epsilon}0}|\mathbf{m}(\boldsymbol{\beta}^{(s)}_{\boldsymbol{\epsilon}0}), \mathbf{C}(\boldsymbol{\beta}^{(s)}_{\boldsymbol{\epsilon}0}))} q(\boldsymbol{\beta}^*_{\boldsymbol{\epsilon}0}|\boldsymbol{\beta}^{(s)}_{\boldsymbol{\epsilon}0}) \right\} \tag{B.5}$$

where

$$q(\boldsymbol{\beta}^*_{\boldsymbol{\epsilon}0}|\boldsymbol{\beta}^{(s)}_{\boldsymbol{\epsilon}0}) = \prod_{i:e_i \in B_\theta(\boldsymbol{\epsilon})} \frac{\exp\left\{\tilde{\mathbf{x}}_i'(\boldsymbol{\beta}^*_{\boldsymbol{\epsilon}0} - \boldsymbol{\beta}^{(s)}_{\boldsymbol{\epsilon}0})\right\}^{I\{e_i \in B_\theta(\boldsymbol{\epsilon}0)\}}}{\left[1 + \exp(\tilde{\mathbf{x}}_i'\boldsymbol{\beta}^*_{\boldsymbol{\epsilon}0})\right] / \left[1 + \exp(\tilde{\mathbf{x}}_i'\boldsymbol{\beta}^{(s)}_{\boldsymbol{\epsilon}0})\right]},$$

and $\phi_q(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the density of a q-variate normal distribution with mean and covariance matrix $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively. Note that the M-H proposal and full conditional distribution for $\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}$ only depends on the observations $\{(\mathbf{x}_i, e_i) : e_i \in B_\theta(\boldsymbol{\epsilon})\}$. Computational speed can be greatly increased by making use of this fact, especially at higher tree levels.

Updating θ

The centering parameter θ can be updated via a random walk M–H step. The proposal is $\log(\theta^*) \sim N(\log(\theta), v_0)$ for some v_0. This candidate θ^* is accepted with probability

$$\alpha(\theta^*|\theta^{(s)}) = 1 \wedge \frac{p(e_1, \ldots, e_n|\theta^*, \boldsymbol{\beta})\pi(\theta^*)}{p(e_1, \ldots, e_n|\theta, \boldsymbol{\beta})\pi(\theta)}, \tag{B.6}$$

where $\pi(\cdot)$ is the prior density of θ.

Summary of the MCMC scheme

Putting all of these updating steps together yields an efficient sampling algorithm for approximating the joint posterior distribution $p(\mathbf{e}, \boldsymbol{\gamma}, \boldsymbol{\beta}, \theta|\mathcal{D})$, where $\mathcal{D} = \{\mathcal{D}_{ij} : i = 1, \ldots, n,\ j = 1, \ldots, n_i\}$ with $\mathcal{D}_{ij}$ being the ijth observed data point. Let $U(a, b)$ denote the uniform distribution with support $[a, b]$. Given the current values $(\mathbf{e}^{(s)}, \boldsymbol{\gamma}^{(s)}, \boldsymbol{\beta}^{(s)}, \theta^{(s)})$ from the sth scan of the Markov chain, we obtain new values as follows:

1. Update $\mathbf{e}$: for each $i \in \{i, \ldots, n\}$,
   a) Generate a candidate e_i^* from $N(e_i^{(s)}, V_i(\boldsymbol{\gamma}^{(s)}, e_i^{(s)}))$.
   b) Compute the acceptance probability $\alpha(e_i^*|e_i^{(s)})$ in (B.2) and sample $u \sim U(0, 1)$. If $u < \alpha(e_i^*|e_i^{(s)})$ assign $e_i^{(s+1)} = e_i^*$, otherwise $e_i^{(s+1)} = e_i^{(s)}$.
2. Update $\boldsymbol{\gamma}$:
   a) Generate a candidate $\boldsymbol{\gamma}^*$ from $N_{K+p}(\mathbf{m}(\boldsymbol{\gamma}^{(s)}, \mathbf{e}^{(s+1)}), \mathbf{V}(\boldsymbol{\gamma}^{(s)}, \mathbf{e}^{(s+1)}))$.
   b) Compute the acceptance probability $\alpha(\boldsymbol{\gamma}^*|\boldsymbol{\gamma}^{(s)})$ in (B.3) and sample $u \sim U(0, 1)$. If $u < \alpha(\boldsymbol{\gamma}^*|\boldsymbol{\gamma}^{(s)})$ assign $\boldsymbol{\gamma}^{(s+1)} = \boldsymbol{\gamma}^*$, otherwise $\boldsymbol{\gamma}^{(s+1)} = \boldsymbol{\gamma}^{(s)}$.
3. Update $\boldsymbol{\beta}$: for each $\boldsymbol{\epsilon} \in E_{J-1} \backslash \emptyset$,
   a) Generate a candidate $\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^*$ from $N_{q+1}(\mathbf{m}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^{(s)}), \mathbf{C}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^{(s)}))$.
   b) Compute the acceptance probability $\alpha(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^*|\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^{(s)})$ in (B.5) and sample $u \sim U(0, 1)$. If $u < \alpha(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^*|\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^{(s)})$ assign $\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^{(s+1)} = \boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^*$, otherwise $\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^{(s+1)} = \boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^{(s)}$. Note that $\mathbf{m}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^{(s)})$, $\mathbf{C}(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^{(s)})$ and $\alpha(\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^*|\boldsymbol{\beta}_{\boldsymbol{\epsilon}0}^{(s)})$ also depend on the updated $\mathbf{e}^{(s+1)}$ in step 1.

4. Update θ:

   a) Generate a candidate $\log(\theta^*)$ from $N(\log(\theta), v_0)$ for some v_0.

   b) Compute the acceptance probability $\alpha(\theta^*|\theta^{(s)})$ in (B.6) and sample $u \sim U(0,1)$. If $u < \alpha(\theta^*|\theta^{(s)})$ assign $\theta^{(s+1)} = \theta^*$, otherwise $\theta^{(s+1)} = \theta^{(s)}$.

The Markov chain achieves approximate stationarity after a large enough burn-in period of iterations; see [14] and [12] for some general convergence conditions. After the convergence is established, a random sample, say $\{(\mathbf{e}^{(s)}, \boldsymbol{\gamma}^{(s)}, \boldsymbol{\beta}^{(s)}, \theta^{(s)})\}_{s=1}^{\mathcal{S}}$, from the posterior distribution can be obtained by saving only every kth scan to reduce chain correlations.

Appendix C: Sample R code to analyze the Iowa SEER data

Subsets of the SEER database are obtained from
`https://seer.cancer.gov/seertrack/data/request/`.

```
#########################################################################
# Breast cancer patients;
# t      -- Follow-up time in Months
# delta  -- Status: 1 = death; 0 = alive(censored)
# ClusterID -- County: State County code (1-99)
# -------- Individual-level Covariates ---------------------------------
# w1     -- Age:    age of the patient at diagnosis in complete years
# w2     -- Stage:  1 = Regional; 0 = Distant or Local
# w3     -- Stage:  1 = Distant;  0 = Regional or Local
# -------- County-level Covariates -------------------------------------
# xtf.Income-- Median Household Income / 1000 in 1993
# xtf.Poverty-- Percentage of families in poverty in 1990
# xtf.Edu -- Percent of Bachelor's degree or higher in 1990
# xtf.RUCC -- Rural-Urban Continuum Codes in 1993
#########################################################################
library(survival)
library(DPpackage)
library(coxme)
library(MASS)

#########################################################
# read the data and create variables
#########################################################
# Individual Level:
data = read.table("SEER_BreastCancer_IA.txt", header = TRUE)
d = data[order(data$County),];
ni = as.vector(table(d$County))
n = length(ni)
N = sum(ni)
ClusterID = rep(1:n, ni)
t = d$t
delta = d$Status
w1 = d$Age;
w2 = d$Regional
w3 = d$Distance

# County Level:
d2 = read.table("CountyCovariates.txt", head=T)
xtf.Income = d2$Income_93/1000
xtf.Poverty = d2$Poverty_Family_90*100
xtf.Edu = d2$Education_90*100
xtf.RUCC = d2$RUCC_93

# Choose the county-level covariate that need to be included
Xindex = c(4) # Index for choosing RUCC only
Xtf = as.matrix( cbind(xtf.Income, xtf.Poverty, xtf.Edu, xtf.RUCC)[,Xindex])
xtf = cbind(rep(1,n), Xtf)
```

```
X = apply(Xtf, 2, function(x) rep(x, ni))
Xnames = c("Income", "Poverty", "Education", "RUCC")[Xindex]

Windex = c(1:3) # Index for choosing individual-level covariates
Ws = cbind(w1, w2, w3)[,Windex]
W = cbind(ClusterID, Ws, X)
Wnames = c("Age", "Regional", "Distance")[Windex]
pw  = length(Wnames); px=length(Xnames); p=pw+px+1; q=px+1;

WX = W[,-1]
colnames(W) = c("ClusterID", Wnames, Xnames)

##########################################################
# Fit coxme
##########################################################
fitcoxme = coxme(Surv(t,delta)~W[,-1] + (1|ClusterID))
fitcoxme

##########################################################
# Fit LDTFP
##########################################################
# Breslow estimate of the baseline hazard based coxme
lambdacoxme=function(time)
{
  pred.e=as.vector(fitcoxme$frail$ClusterID)
  n.pred.e=length(pred.e)
  dummy <- rep(0,n.pred.e)
  msurvival=rep(0,length(time))
  fitcoxme.coeff=as.vector(fitcoxme$coefficients)
  tf   = t[delta == 1];
  nf   = length(tf);
  rt   = matrix(t,N,nf);
  rtf  = matrix(tf,N,nf,byrow=TRUE);
  logic= (rt >= rtf);
  N.pred.e=rep(pred.e,ni)
  Sn0  = as.vector( exp(as.vector(WX%*%fitcoxme.coeff) + N.pred.e)%*%logic )+1e-10
  Lambda=rep(0,length(time))
  ntime = length(time)
  for (i in 1:length(time)){
Lambda[i]=sum( 1/Sn0*(tf<=time[i]) )
  }
  lambda = (Lambda[2:ntime]-Lambda[-ntime])/(time[2:ntime]-time[-ntime])
  lambda
}
# Plot the Breslow estimate for hazard values at each month
time0   = seq(0,47, 1)
lambda0 = lambdacoxme(time0)
sfun0   = stepfun(time0[-1], c(lambda0,0), right=T);
plot(sfun0)

# Determine the cut-points by examing the above plot
cutpoint = c(1, 11, 16, 17, 19, 20, 25, 28, 29, 36, 40, 44, 47); intervals=length(cutpoint)

# Estimated hazards based on the above cut-point,
# which will be used as the starting values of log(lambda) for LDTFP
```

```
hazards0 = rep(0, intervals)
hazards0[1] = mean( lambda0[1:(cutpoint[1])] )
# if intervals>=2
for (i in 2:intervals){
hazards0[i] = mean( lambda0[(cutpoint[i-1]+1): (cutpoint[i])] )
}
sfun1=stepfun(cutpoint, c(hazards0,0), right=T);
lines(sfun1, lwd=2, col=2)

# Function to make a row with '1' at ind----
onv = function(ind,len){onv=rep(0,len); onv[ind]=1; onv}

# Creat new data structure ---------
y={}; Zmat={}; tot=0; off={}; nW=ncol(W)
for(i in 1:N)
{
   tot=tot+1
   Zmat=matrix(append(Zmat,c(W[i,1:nW],onv(1,intervals))),c(nW+intervals,tot))
   off=append(off,min(cutpoint[1],t[i]))
   if(t[i]<=cutpoint[1] && delta[i]==1)
   {
      y=append(y,1)
   } else
   {
      y=append(y,0)
   }
   if (intervals>1)
   {
   for(j in 1:(intervals-1))
   {
       if(t[i]>cutpoint[j])
       {
  off=append(off,min(cutpoint[j+1],t[i])-cutpoint[j])
  tot=tot+1
  Zmat=matrix(append(Zmat,c(W[i,1:nW],onv(j+1,intervals))),c(nW+intervals,tot))
  if(t[i] <= cutpoint[j+1] && delta[i]==1)
  {
     y=append(y,1)
  } else
  {
     y=append(y,0)
  }
       }
   }
   }
}
Zmat = t(Zmat);
id = Zmat[,1];
loghazard = Zmat[,-(1:nW)];
Z = Zmat[,-1]    # design matrix for fixed effects
if ((p-1)==pw) {
colnames(Z)=c( Wnames, paste("loghazard",1:intervals, sep="") )
} else {
colnames(Z)=c( Wnames, Xnames, paste("loghazard",1:intervals, sep="") )
}
```

```
#########################################################
# Fit LDTFP: prior specifications and initial state
#########################################################
# Design matrix for prediction of frailties
xtfpred1=xtf[77,-1]; xtfpred2=xtf[80,-1]
wxpred1=c( c(68.8, 0, 1),xtfpred1); wxpred2=c(c(68.8, 0, 1),xtfpred2)
xpred=rbind( c(rep(0,pw), rep(0,p-1-pw), rep(0,(intervals))) ,
 c(rep(0,pw), rep(0,p-1-pw), rep(0,(intervals))))
xtfpred=rbind(c(1,xtfpred1),c(1,xtfpred2))
prediction=list(xpred=xpred,xtfpred=xtfpred,quans=c(0.025,0.50,0.975))

# Initial based on coxme
loghazards0= log(hazards0)
gammacox   = c(as.vector(fitcoxme$coefficients), loghazards0)
sigma2bcox = as.vector(fitcoxme$vcoef[[1]])
frailcox   = as.vector(fitcoxme$frail$ClusterID)

# Prior information:
maxJ = 4 #
prior=list(maxm=maxJ,alpha=1,mub=rep(0,(p-1+intervals)),a0=1,b0=1,
Sb=diag(rep(10000,(p-1+intervals))),taub1=4,taub2=2*sigma2bcox)

# Initial state
betatf   = matrix(0,nrow=(2**maxJ-1),ncol=q)
gamma0   = gammacox
sigma2b0 = sigma2bcox
frail    = frailcox
state = list(alpha=1,beta=gamma0,b=frail,sigma2b=sigma2b0,betatf=betatf)

# MCMC parameters
nskip=30
mcmc=list(nburn=1000,nsave=500,nskip=nskip,ndisplay=500)

# Fitting the model
mingrid=-1.5; maxgrid=1.5; ngrid=200; xgrid = seq(mingrid, maxgrid, length.out=ngrid);
fitLDTF=LDTFPglmm(y=y,x=Z,roffset=log(off),g=id,family=poisson(log),
                xtf=xtf,prior=prior,prediction=prediction,grid=seq(mingrid,maxgrid,len=ngrid),
  mcmc=mcmc,state=state,status=FALSE)
# Results
sLDTF=summary(fitLDTF)
sLDTF

#-----------functions to plot the densities for frailties -------------
## convert a binary vector to decimal
bintodeci = function (x) {
nx=length(x); index=nx:1
deci=sum(x*2^(index-1))
deci
}
## generate one frailty from G_xtf
gentf = function (xtf, beta, theta2) {
nxtf  = length(xtf)
nbeta = length(beta)/nxtf
J     = log(nbeta+2)/log(2)
```

```
e   = rep(0,J)
e[1]  = rbinom(1,1,0.5)
for (j in 2:J){
pos  = 2^(j-1)-2 + bintodeci(e[1:j])/2 +1
betaj= beta[(nxtf*pos-nxtf+1):(nxtf*pos)]
xbeta= (xtf%*%betaj)[1,1]
probj= exp(xbeta) / (1+exp(xbeta))
e[j] = rbinom(1,1,1-probj)
}
m=bintodeci(e)
ulow= m/2^J; uup=(m+1)/2^J
u=runif(1,ulow, uup)
qnorm(u,0, sqrt(theta2))
}
## frailty density evaluated at e
denfrail = function (e, xtf, beta, theta2) {
nxtf = length(xtf)
nbeta = length(beta)/nxtf
J = log(nbeta+2)/log(2)
tmp = e/sqrt(theta2)
loglik= dnorm(e, 0, sqrt(theta2), log=T)
if (tmp > 4.0) tmp2 = 0.9999683
else if (tmp < -4.0) tmp2 = 3.167124e-05
else tmp2 = pnorm(e, 0, sqrt(theta2))
for (j in 2:J) {
indx = floor( 2^j*tmp2+1 )
for (k in 1:2^(j-1)){
if(indx==(2*k-1) | indx==2*k) {
pos = 2^(j-1)-2 + k
betaj = beta[(nxtf*pos-nxtf+1):(nxtf*pos)]
xbeta = (xtf%*%betaj)[1,1]
Y0 = exp(xbeta)/( 1+exp(xbeta) )
if (indx==(2*k-1)) loglik = loglik + log(Y0)
else loglik = loglik - log( 1+exp(xbeta) )
}
}
}
loglik = loglik + (J-1)*log(2)
exp(loglik)
}
## posterior mean of frailty density
densm = function(xgrid, xtf) {
nwx    = pw + px
ngrid  = length(xgrid)
betatf = fitLDTF$save.state$tfpsave
theta2 = fitLDTF$save.state$thetasave[,(nwx+intervals+1)]
nsave  = length(theta2)
dummy  = rep(0, nsave)
denm  = rep(0, ngrid)
denu  = denm; denl = denm;
for (i in 1:ngrid){
for (j in 1:nsave){
dummy[j] = denfrail(xgrid[i], c(1,xtf), betatf[j,], theta2[j])
}
denm[i] = mean (dummy)
```

```
}
denm
}

## posterior mean of frailty density shifted by the main effect
densmshift = function(xgrid, xtf) {
nwx    = pw + px
ngrid  = length(xgrid)
betatf = fitLDTF$save.state$tfpsave
theta2 = fitLDTF$save.state$thetasave[,(nwx+intervals+1)]
gammax = as.matrix( fitLDTF$save.state$thetasave[,(pw+1):(nwx)] )
nsave  = length(theta2)
dummy  = rep(0, nsave)
denm  = rep(0, ngrid)
denu  = denm; denl = denm;
for (i in 1:ngrid){
for (j in 1:nsave){
evalue   = xgrid[i] - (gammax[j,]%*%xtf)[1,1]
dummy[j] = denfrail(evalue, c(1,xtf), betatf[j,], theta2[j])
}
denm[i] = mean (dummy)
}
denm
}

#----------------- functinos to plot survival curves for LDTFP ---------
ScLDTF = function(t,wx,gamma.h,frailty,intervals)
{
p = length(wx)
k = 1
temp = exp(gamma.h[p+1])*min(cutpoint[1],t)
while (t>cutpoint[k] && k<intervals)
{
   temp = temp+exp(gamma.h[p+k+1])*(min(cutpoint[k+1],t)-cutpoint[k])
   k=k+1
}
exp(-temp*exp((wx%*%gamma.h[1:p])[1,1] + frailty))
}
## Function to plot survival curve
surLDTF = function (t, wx)
{
  nwx    = length(wx)
  xtf    = c( 1, wx[-(1:pw)] )
  betatf = fitLDTF$save.state$tfpsave
  theta2 = fitLDTF$save.state$thetasave[,(nwx+intervals+1)]
  gamma.h= fitLDTF$save.state$thetasave[,1:(nwx+intervals)]
  nsave  = nrow(gamma.h)
  nt     = length(t)
  dummy  = matrix(0, nsave, nt)
  n.pred.e=500;
  for (j in 1:nsave)
  {
   tmp = matrix(0, n.pred.e, nt);
   for (i in 1:n.pred.e) {
   pred.e=gentf (xtf, betatf[j,], theta2[j]);
```

```
   tmp[i,] = as.vector( sapply(t, ScLDTF, wx, gamma.h[j,], pred.e, intervals))
   }
   remove(tmp);
   dummy[j,] = colMeans(tmp);
  }
  colMeans(dummy)
}

#-------------------------- LPML and DIC ----------------------------
# density function evaluated at each data point
densfun = function(t,delta,wx,gamma.h,frailty,intervals){
p = length(wx)
k = 1
temp = exp(gamma.h[p+1])*min(cutpoint[1],t)
while (t>cutpoint[k] && k<intervals)
{
   temp = temp+exp(gamma.h[p+k+1])*(min(cutpoint[k+1],t)-cutpoint[k])
   k=k+1
}
lambda.K= gamma.h[p+k]
temp2 = (wx%*%gamma.h[1:p])[1,1] + frailty
exp((lambda.K+temp2)*delta-temp*exp(temp2))
}
## calculate LPML and DIC
nwx = pw + px
frails= fitLDTF$save.state$randsave
betatf= fitLDTF$save.state$tfpsave
theta2 = fitLDTF$save.state$thetasave[,(nwx+intervals+1)]
gamma.h = fitLDTF$save.state$thetasave[,1:(nwx+intervals)]
nsave = nrow(gamma.h)
cpo1 = rep(0, N)
cpo2 = rep(0, N)
dummy = rep(0, nsave)
for (i in 1:N){
Ni = rep(1:n, ni)
for (j in 1:nsave){
pred.e   = frails[j,Ni[i]]
dummy[j] = densfun(t[i],delta[i],W[i,-1],gamma.h[j,],pred.e,intervals)
}
cpo1[i] = 1/mean(1/dummy)
cpo2[i] = mean(log(dummy))
}
LPML.LDTF = sum(log(cpo1))
LPML.LDTF

Dbar = -2*sum(cpo2)
tmp = rep(0, N)
mean.pred.e = colMeans(frails)
mean.gamma.h = as.vector(colMeans(gamma.h))
for (i in 1:N){
Ni = rep(1:n, ni)
pred.e = mean.pred.e[Ni[i]]
tmp[i] = densfun(t[i],delta[i],W[i,-1],mean.gamma.h,pred.e,intervals)
}
pD = Dbar + 2*sum(log(tmp))
```

```
DIC.LDTF = Dbar+pD
DIC.LDTF

#########################################################
# Plots: frailty densities at xtfpred1, xtfpred0 and xtfpred2
#     survival curves at wxpred1, wxpred0, wxpred2
#########################################################
# Take xtfpred0 as the mean of county-level covariate values
xtf.quan = apply(Xtf, 2, function(x) quantile(x, c(0.05, 0.95)))
xtfpred0 = apply(Xtf, 2, mean);
if (Xtf[1,px]==8) {xtfpred0[px]=5; xtf.quan[,px]=c(2,9)}

# plot frailty densities and shifted version
xgrid2 = seq(-2.5, 2.5, length=300) + ( xtfpred0%*%fitLDTF$coeff[(pw+1):(pw+px)] )[1,1]
ngrid2 = length(xgrid2)
densm0 = densm(xgrid, xtfpred0)
densmshift0 = densmshift(xgrid2, xtfpred0)
densm1 = matrix(0,px,ngrid)
densm2 = densm1
densmshift1 = matrix(0,px,ngrid2)
densmshift2 = densmshift1
for (i in 1:px){
pindx = i
xtfpred1 = xtfpred0; xtfpred1[pindx] = xtf.quan[1,pindx]
xtfpred2 = xtfpred0; xtfpred2[pindx] = xtf.quan[2,pindx]
densm1[i,] = densm(xgrid, xtfpred1)
densm2[i,] = densm(xgrid, xtfpred2)
densmshift1[i,] = densmshift(xgrid2, xtfpred1)
densmshift2[i,] = densmshift(xgrid2, xtfpred2)
}


time = seq(0, max(t), length.out=200); ntime=length(time)
wpred0 = c(mean(d$Age),0,1)
wxpred0 = c(wpred0,xtfpred0)
mLDTF0 =surLDTF(time, wxpred0)
survm1 = matrix(0,px,ntime)
survm2 = survm1
for (i in 1:px){
pindx = i
xtfpred1 = xtfpred0; xtfpred1[pindx] = xtf.quan[1,pindx]
xtfpred2 = xtfpred0; xtfpred2[pindx] = xtf.quan[2,pindx]
wxpred1 = c(wpred0,xtfpred1)
wxpred2 = c(wpred0,xtfpred2)
survm1[i,] = surLDTF(time, wxpred1)
survm2[i,] = surLDTF(time, wxpred2)
}

#save.image("RUCC.RData")

par(mfrow=c(3,px))
par(mar = c(3, 3, 2, 1)+0.2)
par(mgp = c(2.2, 1, 0))
# frailty density
for (i in 1:px) {
```

```
plot(xgrid, densm0, "l", lty="solid", xlab="values", ylab="density", ylim=c(0,1.9), lwd=2,
main=paste("Model 1:", Xnames[i], sep=" " ))
lines(xgrid, densm1[i,], "l", lty="dashed", lwd=2, col=2)
lines(xgrid, densm2[i,], "l", lty="dotted", lwd=2, col=4)
legend(0.52, 1.9, c(paste(Xnames[i], "2", sep="="), paste(Xnames[i], "5", sep="="),
        paste(Xnames[i], "9", sep="=")), col = c(2,1,4),
       lty = c("dashed", "solid", "dotted"), cex=1)
}
# shifted frailty density
for (i in 1:px) {
plot(xgrid2, densmshift0, "l", lty="solid", xlab="values", ylab="density", ylim=c(0,1.9), lwd=2,
main=paste("Model 1:", Xnames[i], sep=" " ))
lines(xgrid2, densmshift1[i,], "l", lty="dashed", lwd=2, col=2)
lines(xgrid2, densmshift2[i,], "l", lty="dotted", lwd=2, col=4)
legend(0.4, 1.9, c(paste(Xnames[i], "2", sep="="), paste(Xnames[i], "5", sep="="),
        paste(Xnames[i], "9", sep="=")), col = c(2,1,4),
       lty = c("dashed", "solid", "dotted"), cex=1)
}
# survival curve
for (i in 1:px) {
plot(time, mLDTF0, "l", lty="solid", xlab="values", ylab="survival", ylim=c(0,1), lwd=2,
main=paste("Model 1:", Xnames[i], sep=" " ))
lines(time, survm1[i,], "l", lty="dashed", lwd=2, col=2)
lines(time, survm2[i,], "l", lty="dotted", lwd=2, col=4)
legend(31.5, 0.99, c(paste(Xnames[i], "2", sep="="), paste(Xnames[i], "5", sep="="),
        paste(Xnames[i], "9", sep="=")), col = c(2,1,4),
       lty = c("dashed", "solid", "dotted"), cex=1)
}

# Add the fitted baseline hazards plot based on LDTFP
# to the previous plot based on coxme
plot(sfun0, xlim=c(0,50))
lambda2 = as.vector(exp( fitLDTF$coefficients[p:(p+intervals-1)] ));
sfun2=stepfun(cutpoint, c(lambda2,0), right=T); lines(sfun2, lwd=2, col=4)
```

Appendix D: Additional simulation results

Effect of level J for larger sample size under Scenario I

To assess the effect of the level of the LDTFP on the posterior inferences, simulated 15 dat sets from the model with $n = 1,000$, under **Scenario I**. For each data set, we fitted different versions of the proposed model by considering $J = 4, 5, 6, 7$. Figures 1-4 present the fitted frailty densities and survival curves averaged over replicates. For $J = 4$ (Figure 1), we do see increased accuracy of estimated frailty and survival curves from increasing the number of clusters to $n = 1000$. Increasing from $J = 4$ to $J = 5$ does lead to a slight improvement on the estimated shape of the frailty density. Further increasing J to 6 or 7 does not help much, but roughly doubles the computing time for each additional level.

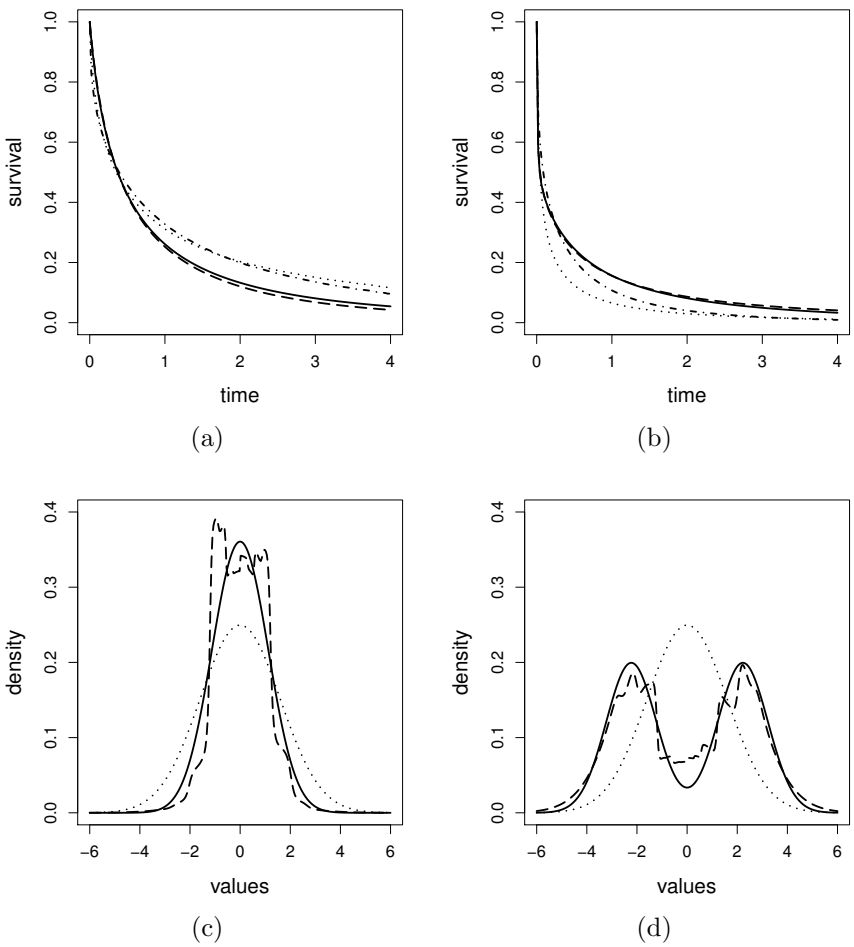


Figure 1: Simulated data – Scenario III: Mean over simulations of the estimated curves under the proposed model with $J = 4$ and sample size $n = 1000$. Panels (a) and (b) show the results for the survival functions. Panels (c) and (d) show the results for the frailty densities. Panels (a) and (c) show the results for covariate values $(2, 1, -2)$. Panels (b) and (d) show the results for covariate values $(0, 1,\ 2)$. The true curves are represented by continuous lines. The results under the proposed model are represented by dashed lines. The results under the exchangeable Gaussian frailty model are represented by dotted lines. In Panels (a) and (b) the results obtained under of PSF approach are represented by a dot-dashed line.

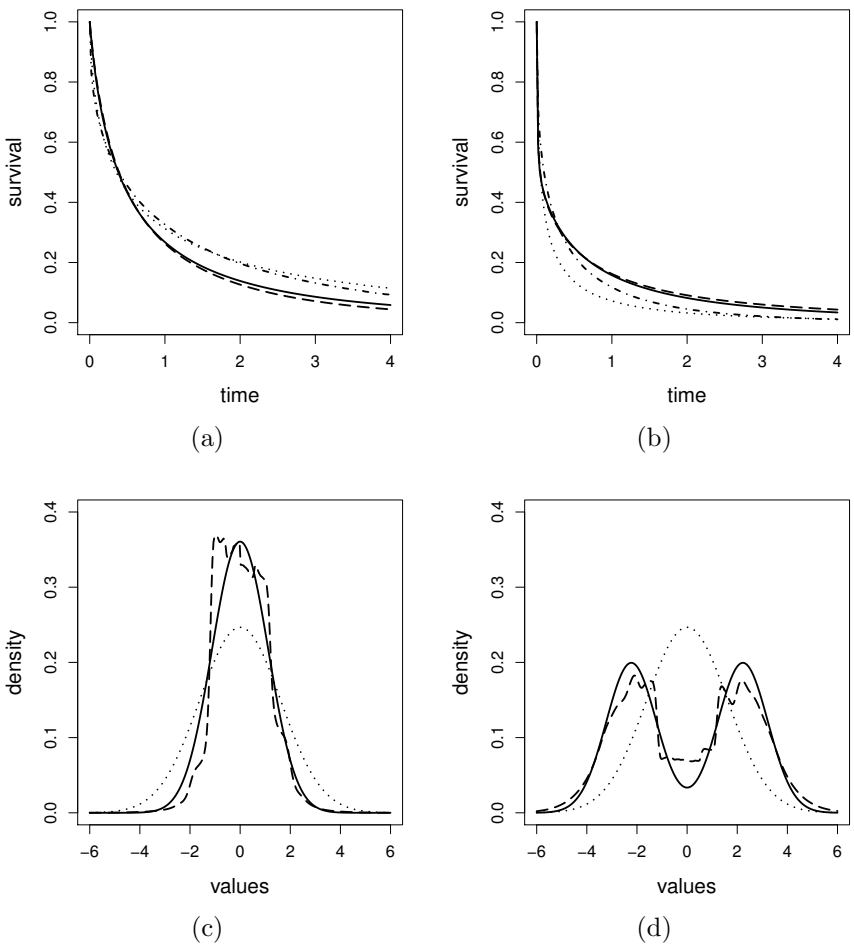


Figure 2: Simulated data – Scenario III: Mean over simulations of the estimated curves under the proposed model with $J = 5$ and sample size $n = 1000$. Panels (a) and (b) show the results for the survival functions. Panels (c) and (d) show the results for the frailty densities. Panels (a) and (c) show the results for covariate values $(2, 1, -2)$. Panels (b) and (d) show the results for covariate values $(0, 1,\ 2)$. The true curves are represented by continuous lines. The results under the proposed model are represented by dashed lines. The results under the exchangeable Gaussian frailty model are represented by dotted lines. In Panels (a) and (b) the results obtained under of PSF approach are represented by a dot-dashed line.

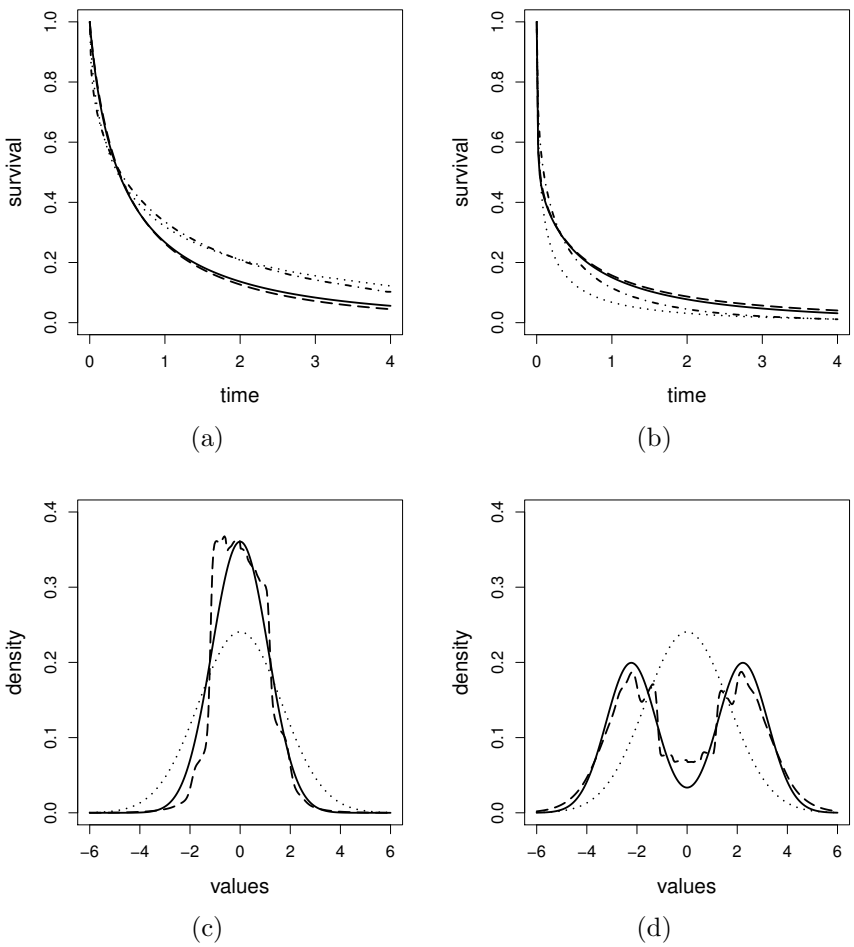


Figure 3: Simulated data – Scenario III: Mean over simulations of the estimated curves under the proposed model with $J = 6$ and sample size $n = 1000$. Panels (a) and (b) show the results for the survival functions. Panels (c) and (d) show the results for the frailty densities. Panels (a) and (c) show the results for covariate values $(2, 1, -2)$. Panels (b) and (d) show the results for covariate values $(0, 1,\ 2)$. The true curves are represented by continuous lines. The results under the proposed model are represented by dashed lines. The results under the exchangeable Gaussian frailty model are represented by dotted lines. In Panels (a) and (b) the results obtained under of PSF approach are represented by a dot-dashed line.

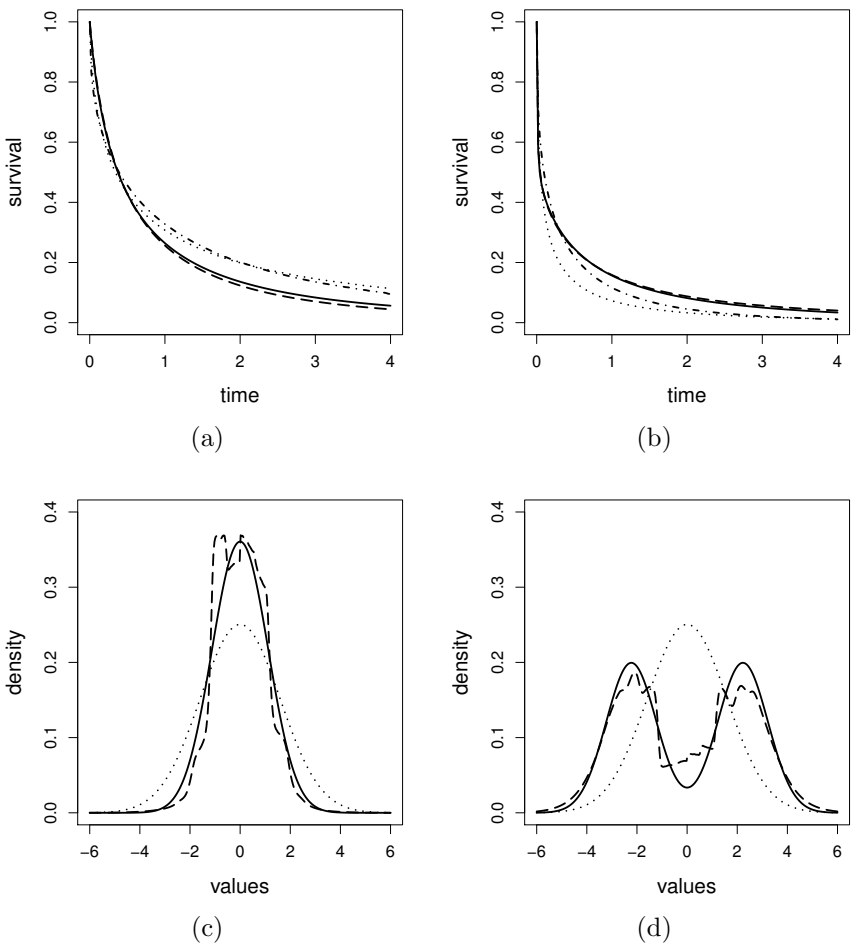


Figure 4: Simulated data – Scenario III: Mean over simulations of the estimated curves under the proposed model with $J = 7$ and sample size $n = 1000$. Panels (a) and (b) show the results for the survival functions. Panels (c) and (d) show the results for the frailty densities. Panels (a) and (c) show the results for covariate values $(2, 1, -2)$. Panels (b) and (d) show the results for covariate values $(0, 1,\ 2)$. The true curves are represented by continuous lines. The results under the proposed model are represented by dashed lines. The results under the exchangeable Gaussian frailty model are represented by dotted lines. In Panels (a) and (b) the results obtained under of PSF approach are represented by a dot-dashed line.

Comparison with the MPT frailty Cox model under Scenario I

Under **Scenario I**, we also fitted the exchangeable mixture of Polya trees (MPT) [7] frailty Cox's model using the function `PTglmm` available in `DPpackage` [10]. The results for regression coefficients under the proposed model and the MPT approach are given in Table 2. The average of the estimated frailty distributions and survival functions across simulation data sets for some specific covariate values are presented in Figure 5, and the corresponding Monte Carlo mean and standard deviations for the ISE are given in Table 3. The MPT approach outperforms the GF and PSF methods when the cluster-level covariate $x = 2$, but performs worse when $x = -2$. This is not surprising, since the MPT gives us a nonparametric frailty distribution estimate which is a balance between one-mode density and two-mode density. Thus the performance of MPT estimates will depends on the value of cluster-level covariates.

Table 2: Simulation data – Scenario I: True value, bias of the point estimator, mean (across Monte Carlo simulations) of the posterior standard deviations/standard errors (MEAN-SD), standard deviation (across Monte Carlo simulations) of the point estimator (SD-MEAN) and Monte Carlo coverage probability for the 95% credible interval/confidence interval (CP) for the regression parameters. The results are presented under the proposed model and under the MPT approach.

			Proposed Model			MPT Model			
Parameters	True	BIAS	MEAN-SD	SD-MEAN	CP	BIAS	MEAN-SD	SD-MEAN	CP
ξ_1	1.0	0.011	0.052	0.054	0.930	0.009	0.046	0.041	0.985
ξ_2	0.5	0.008	0.088	0.090	0.945	0.005	0.087	0.087	0.945
ξ_x	1.0	-0.009	0.141	0.126	0.965	0.081	0.133	0.176	0.770

Table 3: Simulated data – Scenario I: Monte Carlo mean (Monte Carlo standard deviation) for the ISE of the survival function for two different predictor values. The results for the different approaches under both simulation scenarios are presented. The numbers correspond to 10^3 times the original values.

(w_1, w_2, x)	Proposal	GF approach	PSF approach	MPT approach
$(2, 1, -2)$	2.02 (2.48)	4.37 (3.46)	6.28 (3.49)	7.71 (6.03)
$(0, 1, 2)$	1.94 (2.53)	10.5 (6.86)	14.3 (10.9)	8.19 (5.55)

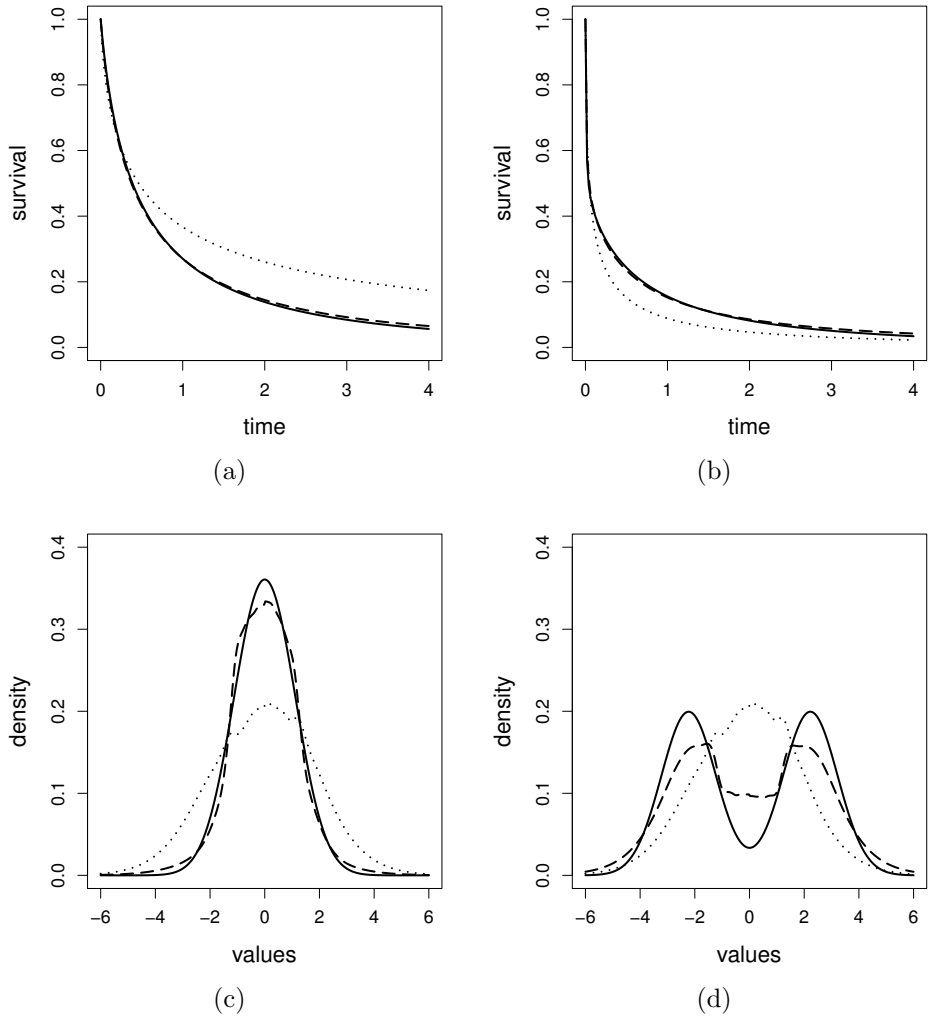


Figure 5: Simulated data – Scenario I: Mean, across simulations, of the posterior mean of the survival and frailty density functions under the proposed model. Panels (a) and (b) show the results for the survival functions. Panels (c) and (d) show the results for the frailty densities. Panels (a) and (c) show the results for covariate values $(2, 1, -2)$. Panels (b) and (d) show the results for covariate values $(0, 1,\ 2)$. The true curves are represented by continuous lines. The results under the proposed model are represented by dashed lines. The results under the MPT frailty model are represented by dotted lines.

Comparison with the GF approach under additional Scenario III

We also considered a third scenario favorable to the GF model to evaluate the behaviour of the proposed approach when a standard parametric exchangeable (covariate-free) frailty model is correct. Clustered failure time data were simulated in the same way as Scenario I, but with frailties generated from standard normal distribution. The results for the regression coefficients under the proposed model and the GF approach are given in Table 4. We can see that both approaches yielded unbiased estimates of $\boldsymbol{\xi}$ and θ^2, and almost the same values for MEAN-SD. However, the estimated standard error for ξ_x is severely underestimated by the GF approach, with CP of 86%. The average of the estimated frailty distributions and survival functions across simulated data sets for some specific covariate values are presented Figure 4. The results suggest that essentially no differences among the three methods are observed; all estimated functions are close to the truth, indicating that there is little price to be paid for the extra generality when using the proposed model when normality and exchangeability are valid assumptions.

Table 4: Simulation data – Scenario III: True value, bias of the point estimator, mean (across Monte Carlo simulations) of the posterior standard deviations/standard errors (MEAN-SD), standard deviation (across Monte Carlo simulations) of the point estimator (SD-MEAN) and Monte Carlo coverage probability for the 95% credible interval/confidence interval (CP) for the regression parameters. The results are presented under the proposed model and under the GF approach.

		Proposed Model				GF Model			
Parameters	True	BIAS	MEAN-SD	SD-MEAN	CP	BIAS	MEAN-SD	SD-MEAN	CP
ξ_1	1.0	-0.002	0.044	0.045	0.940	-0.005	0.050	0.054	0.920
ξ_2	0.5	0.002	0.084	0.088	0.940	0.001	0.085	0.091	0.935
ξ_x	1.0	0.002	0.087	0.068	0.975	-0.018	0.055	0.070	0.860
θ^2	1.0	0.042		0.209		-0.036		0.185	

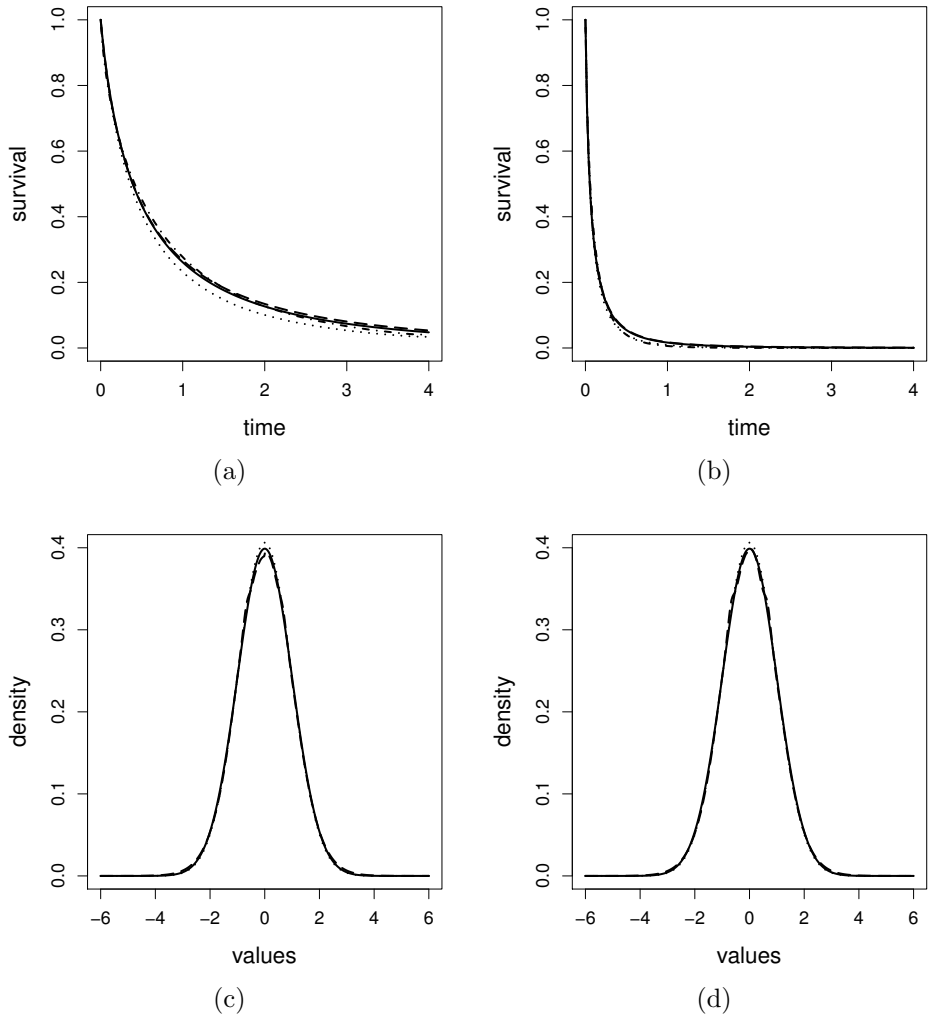


Figure 6: Simulated data – Scenario III: Mean, across simulations, of the posterior mean of the survival and frailty density functions under the proposed model. Panels (a) and (b) show the results for the survival functions. Panels (c) and (d) show the results for the frailty densities. Panels (a) and (c) show the results for covariate values $(2, 1, -2)$. Panels (b) and (d) show the results for covariate values $(0, 1,\ 2)$. The true curves are represented by continuous lines. The results under the proposed model are represented by dashed lines. The results under the exchangeable Gaussian frailty model are represented by dotted lines. In Panels (a) and (b) the results obtained under of PSF approach are represented by a dot-dashed line.

In addition, the results of the comparison of the estimated survival curves are presented in Table 5, where the Monte Carlo mean and standard deviations for the ISE for two different predictor values are given. Even when an exchangeable frailty model with normal distribution applies, the proposed model is slightly more beneficial in estimating the survival functions. This may partly come from the difference between the Bayesian and frequentist methods for computation. The Bayesian approach averages over the posterior whereas the frequentist approach uses "plug-in" estimates which can underestimate variability.

Table 5: Simulated data – Scenario III: Monte Carlo mean (Monte Carlo standard deviation) for the ISE of the survival function for two different predictor values. The results for the different approaches are presented. The numbers correspond to 10^3 times the original values.

(w_1, w_2)	Proposal	GF approach	PSF approach
$(2, 1)$	1.66 (2.66)	2.02 (2.89)	2.38 (3.46)
$(0, 1)$	1.53 (1.88)	1.70 (1.96)	2.16 (2.46)

Appendix E: Additional analysis of SEER data

Additional cut-point vector specifications

We consider additional cut-point vector specifications as follows:

Case IV: $\mathbf{a} = (2, 4, 7, 11, 15, 18, 22, 26, 32, 47)$, which is determined by the quantiles of the distribution of event times based on Kaplan-Meier curve so that each interval contains almost equal number of events.

Case IVb: $\mathbf{a} = (1, 3, 5, 8, 11, 13, 16, 18, 22, 25, 29, 34, 47)$, which is determined in the same way as **Case IV** but with size 13.

Case IIb: $\mathbf{a} = (2, 5, 9, 12, 15, 17, 21, 25, 28.2, 32.6, 37, 42.5, 47)$, where a_k is the $\frac{k}{13}$th quantile of the empirical distribution of observed survival times.

Table 6 shows the DIC and LPML for the three models considered in the main article. We can see that **Case IV** gives a little better fit than the **Case II**, but still much worse than the **Case I**. It suggests that the Kaplan-Meier based specification of the cut-points provides slightly better fit than the empirical distribution based specification. When we increase the size of cut-point vector from 10 to 13 for **Case II** and **Case IV**, the model fit only improves 1 unit for LPML and 4 units for DIC across the three models, compared with the Case II with 10 cut-points. It indicates that carefully choosing the cut-points is more important than simply increasing the number of cut-points.

Table 6: Iowa SEER data: Deviance information criteria (DIC) and log of the pseudo marginal likelihood (LPML) for models under consideration.

	Case IV		**Case IVb**		**Case IIb**	
Model	DIC	LPML	DIC	LPML	DIC	LPML
1	4457	-2232	4456	-2232	4459	-2233
2	4459	-2233	4460	-2234	4461	-2234
3	4458	-2232	4459	-2234	4459	-2234

Figure 7 presents the fitted predictive frailty densities for both $\mathbf{e}_i$ (median-zero) and $\mathbf{e}_i + \mathbf{x}_i'\boldsymbol{\xi}_x$ and survival curves for women with mean entry age 68.8 years and distant stage of disease who live in the counties with different levels of median household income or RUCC, under **Model 1** and **Model 2** for **Case IV**. We can see that all the estimated curves are very similar to those obtained under **Case I** considered in the main article.

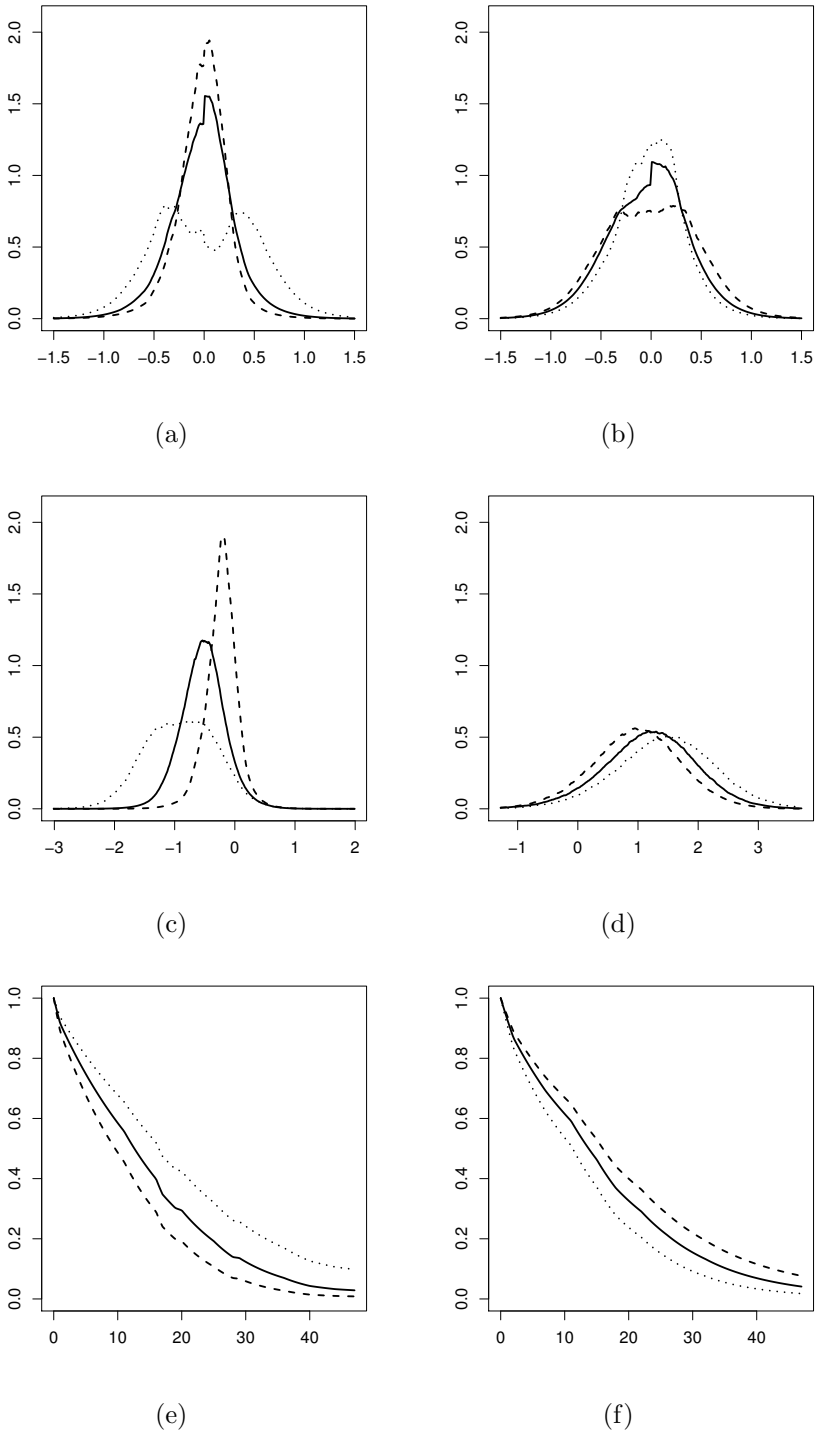


Figure 7: Iowa SEER data: Fitted predictive frailty densities (Panels (a) and (b)), frailty densities with location shifts (Panels (c) and (d)) and survival curves (Panels (e) and (f)) for women with mean entry age 68.8 years and distant stage of disease who live in the counties with different county covariate levels under Model 1 (Panels (a), (c) and (e)) and Model 2 (Panels (b), (d) and (f)) under **Case IV**. In Panels (a), (c) and (e), the results for RUCC=2, 5 and 9 are displayed as dashed, continuous and dotted lines, respectively. In Panels (b), (d) and (f), the results for Income=23.354, 29.176 and 35.301 are displayed as dashed, continuous and dotted lines, respectively.

Additional comparison between ours and marginal PH model

We additionally presented the fixed effects under the marginal non-frailty PH model (i.e. using the R function `coxph` with the option `cluster`) across **Model 1**–**Model 3** in Table 7. Note that the marginal PH model is equivalent to the PSF model from the marginal model perspective, so these fixed effects are exactly the same as those obtained under the PSF model. Note that the coefficient estimates under the marginal PH model have population-averaged interpretations, and cannot be directly compared with those fitted from the proposed frailty PH model due to different model structures.

Table 7: Iowa SEER data: Point estimates (95% confidence intervals) of fixed effects $\boldsymbol{\xi}$ from various models under the marginal PH model.

Predictor	**Model 1**	**Model 2**	**Model 3**	**Model 0**
ξ_1 (Age)	0.02	0.021	0.021	0.019
	(0.014, 0.026)	(0.015, 0.027)	(0.015, 0.027)	(0.013, 0.025)
ξ_2 (Regional)	0.30	0.30	0.30	0.30
	(0.08, 0.51)	(0.09, 0.52)	(0.08, 0.52)	(0.09, 0.50)
ξ_3 (Distant)	1.63	1.68	1.64	1.64
	(1.35, 1.92)	(1.39, 1.97)	(1.36, 1.93)	(1.34, 1.95)
ξ_{x_1} (RUCC)	-0.106		-0.087	
	(-0.154, -0.058)		(-0.152, -0.022)	
ξ_{x_2} (Income)		0.049	0.014	
		(0.019, 0.079)	(-0.025, 0.054)	

Although the fixed effects are not appropriate to be compared between the proposed model and the marginal PH model, we carefully compared the fitted survival curves across **Model 1**–**Model 3** for different covariates levels in Figure 8 and 9. Overall, the marginal PH model under-predicts survival time up to about 1 month compared with our proposed model for patients with mean entry age 68.8 years and distant stage of disease who live in the same county. It may be partly due to the fact that the marginal PH model and the PSF model fail to detect the bi-modal behavior of the frailty distribution for rural counties.

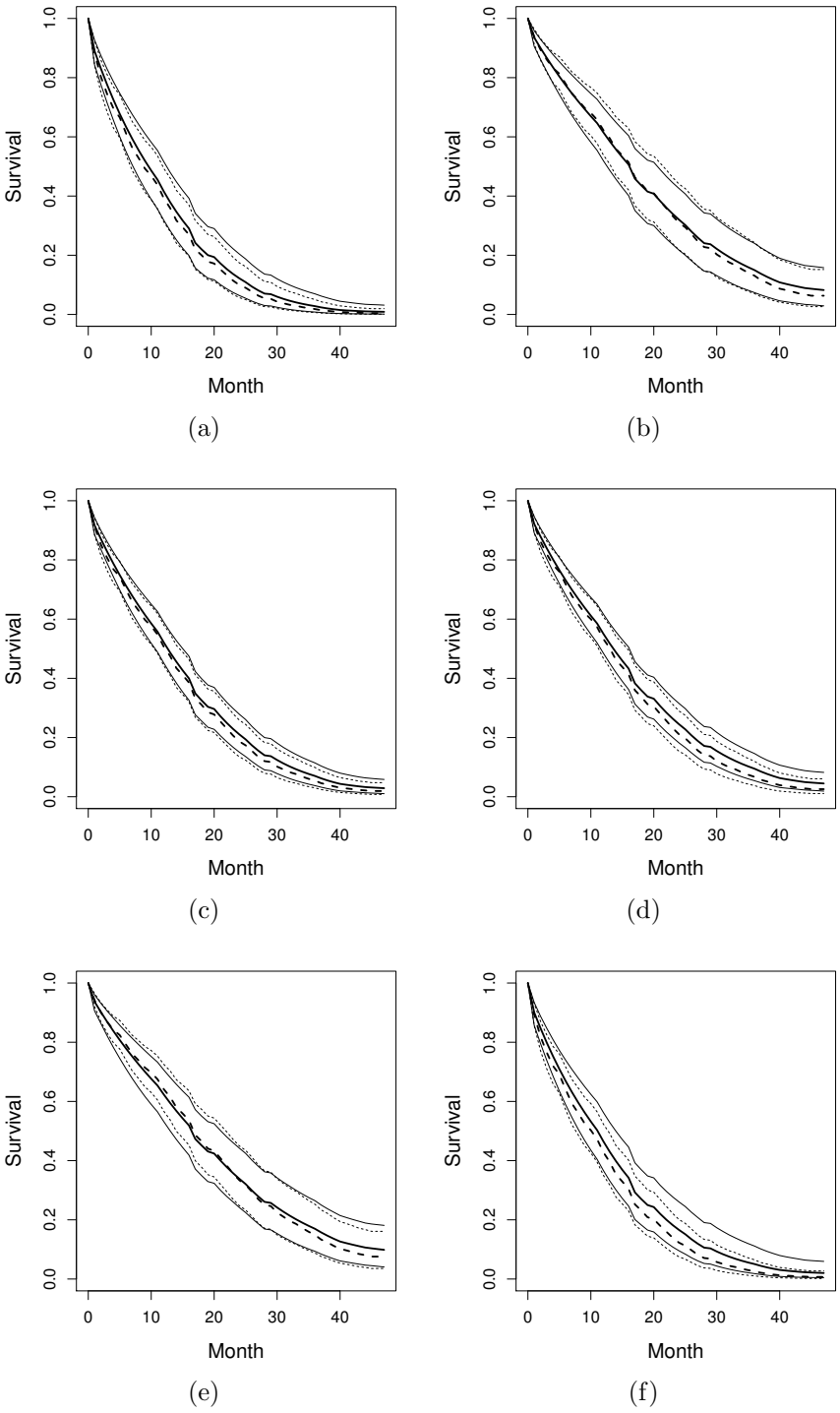
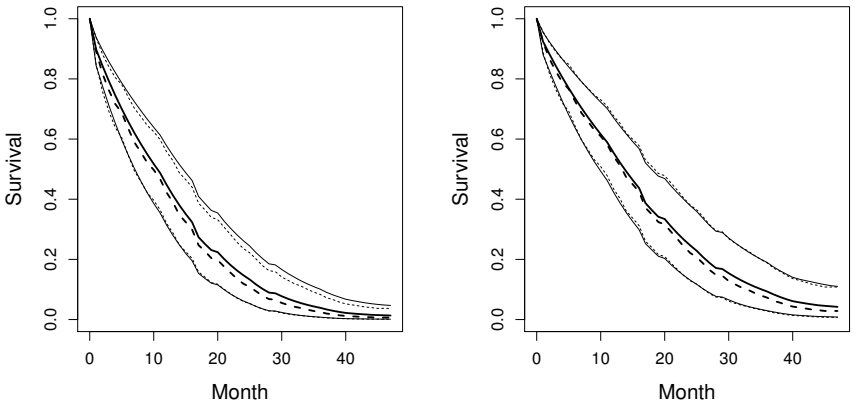
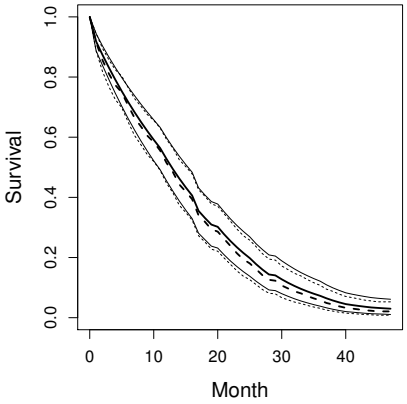


Figure 8: Iowa SEER data: Fitted predictive survival curves (thicker lines) with 95% confidence/credible intervals (thinner lines) for women with mean entry age 68.8 years and distant stage of disease who live in the counties with different county covariate levels under the proposed model (solid lines) and under the marginal PH model (dashed lines). Panels (a), (c) and (e) are for RUCC=2, 5 and 9, respectively under Model 1. Panels (b), (d) and (f) are for Income=23.354, 29.176 and 35.301, respectively under Model 12.

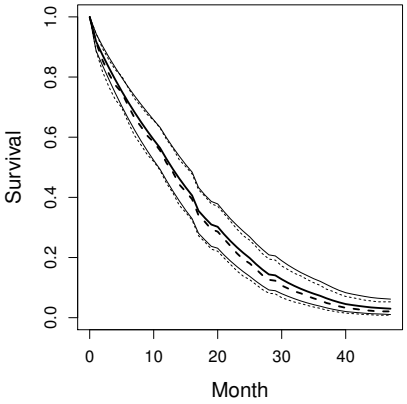


(a)

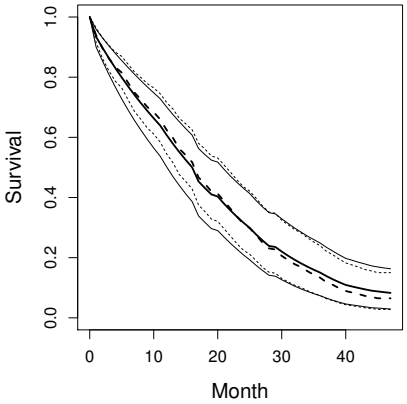
(b)



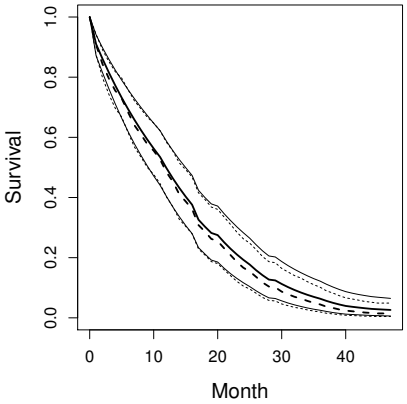
(c)



(d)



(e)



(f)

Figure 9: Iowa SEER data: Fitted predictive survival curves (thicker lines) with 95% confidence/credible intervals (thinner lines) for women with mean entry age 68.8 years and distant stage of disease who live in the counties with different county covariate levels under the proposed model (solid lines) and under the marginal PH model (dashed lines). Panels (a), (c) and (e) are for RUCC=2, 5 and 9. Panels (b), (d) and (f) are for Income=23.354, 29.176 and 35.301.

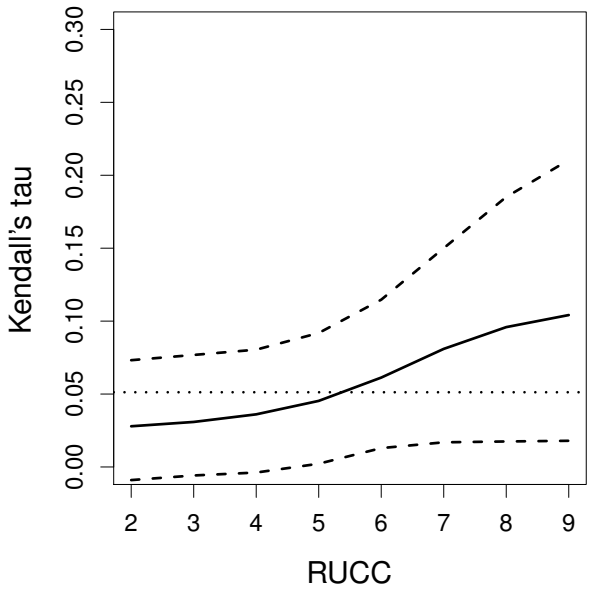


Figure 10: Iowa SEER data: Estimated Kendall's tau as a function of RUCC (solid line) with point-wise 95% credible interval (dashed line) for individuals with mean entry age 68.8 years and distant stage of disease under Model 1. The Kendall's tau under the gamma frailty PH model is also presented in dotted line.

Measures of dependence within cluster

Kendall's tau is widely used to measure the overall dependence of a pair of subjects over the entire lifespan by integrating over time, which is based on a form of dependence known as *c*oncordance [11]. As suggested by one of the referees, we calculated the Kendall's tau for different level of county-specific covariates under the proposed model. Suppose a posterior sample $\{(\boldsymbol{\gamma}^{(s)}, \mathbf{e}^{(s)}, \theta^{(s)}, \boldsymbol{\beta}^{(s)})\}_{s=1}^{S}$ has been obtained. For the sth iteration, we first draw a random sample, say $\{e_k^{(s)}\}_{k=1}^{\mathcal{K}}$, from the posterior frailty density $g_{\mathbf{x}}(\cdot|\theta^{(s)}, \boldsymbol{\beta}^{(s)})$, then draw a pair of survival times $(t_{1,k}^{(s)}, t_{2,k}^{(s)})$ independently from the predictive survival function $S_{\mathbf{w}}(t|\boldsymbol{\gamma}^{(s)}, e_k^{(s)})$, where $\mathbf{w} = (\tilde{\mathbf{w}}', \mathbf{x}')'$ with $\tilde{\mathbf{w}}$ being any fixed vector of individual level covariates. Now there are a total of $\mathcal{K}(\mathcal{K}-1)/2$ pairs of survival times $(t_{1,k}^{(s)}, t_{2,k}^{(s)})$ and $(t_{1,\ell}^{(s)}, t_{2,\ell}^{(s)})$ among the random sample $\{(t_{1,k}^{(s)}, t_{2,k}^{(s)})\}_{k=1}^{\mathcal{K}}$. We then denote by $C_{\mathbf{x}}^{(s)}$ the total number of concordant pairs, that is, if $(t_{1,k}^{(s)} - t_{1,\ell}^{(s)})(t_{2,k}^{(s)} - t_{2,\ell}^{(s)})$ is strictly positive. It follows that the Kendall's tau for the sth iteration can be estimated by

$$\tau_s(\mathbf{x}) = \frac{2C_{\mathbf{x}}^{(s)}}{\mathcal{K}(\mathcal{K}-1)/2} - 1.$$

Thus we could estimate $\tau(\mathbf{x})$ and its credible interval based on the posterior sample $\{\tau_s(\mathbf{x})\}_{s=1}^{\mathcal{S}}$. For example, Figure 10 shows the estimated Kendall's tau for each possible RUCC level (from 2 to 9) with point-wise 95% credible interval under the **Model I** considered in the main body of paper. We can see that the within cluster correlation increases with RUCC, which agrees perfectly with the finding that frailty variance increases with RUCC. For comparison, we also calculated the Kendall's tau for the gamma frailty PH model, which is static at a constant value, as shown in Figure 10 with dotted line.

References

[1] Ronald Christensen. *Plane answers to complex questions: the theory of linear models*. Springer Science+ Business Media, 2011.

[2] J. Fabius. Asymptotic behavior of Bayes' estimates. *The Annals of Mathematical Statistics*, 35(2):846–856, 1964.

[3] T S Ferguson. Prior distributions on spaces of probability measures. *The Annals of Statistics*, 2:615–629, 1974.

[4] D.A. Freedman. On the asymptotic behavior of Bayes' estimates in the discrete case. *The Annals of Mathematical Statistics*, 34(4):1386–1403, 1963.

[5] D. Gamerman. Sampling from the posterior distribution in generalized linear mixed models. *Statistics and Computing*, 7(1):57–68, 1997.

[6] T Hanson, A Kottas, and A Branscum. Modelling stochastic order in the analysis of receiver operating characteristic data: Bayesian nonparametric approaches. *Journal of the Royal Statistical Society: Series C*, 57:207–225, 2008.

[7] T E Hanson. Inference for mixtures of finite Polya tree models. *Journal of the American Statistical Association*, 101(476):1548–1565, 2006.

[8] A Jara and T E Hanson. A class of mixtures of dependent tailfree processes. *Biometrika*, 98:553–566, 2011.

[9] A Jara, T E Hanson, and E Lesaffre. Robustifying generalized linear mixed models using a new class of mixtures of multivariate Polya trees. *Journal of Computational and Graphical Statistics*, 18(4):838–860, 2009.

[10] A. Jara, T.E. Hanson, F.A. Quintana, P. Müller, and G.L. Rosner. DPpackage: Bayesian semi- and nonparametric modeling in R. *Journal of Statistical Software*, 40(5):1, 2011.

[11] Roger B Nelsen. *An Introduction to Copulas*. Springer, 2nd edition, 2006.

[12] Gareth O Roberts and Adrian FM Smith. Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Processes and Their Applications*, 49(2):207–216, 1994.

[13] Angela Schörgendorfer, Adam J Branscum, and Timothy E Hanson. A Bayesian goodness of fit test and semiparametric generalization of logistic regression with measurement data. *Biometrics*, 69(2):508–519, 2013.

[14] L. Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1701–1728, 1994.

[15] Mike West. Outlier models and prior distributions in Bayesian linear regression. *Journal of the Royal Statistical Society, Series B*, 46(3):431–439, 1984.

[16] A Zellner. Applications of Bayesian analysis in econometrics. *The Statistician*, 32:23–34, 1983.