

**Supplementary material A: Implementation details and R code**  
**“Objective Bayesian survival analysis using shape mixtures of log-normal distributions”**

Bayesian inference is implemented through Markov chain Monte Carlo (MCMC) using the hierarchical representation of the SMLN family and the data augmentation idea of Tanner and Wong (1987). Throughout, we use the priors presented in Section 3.1 of the paper. Inference was implemented in R<sup>1</sup> and code is freely available in the “SMLN codes.zip” file at **\*\*add link\*\***. This includes the MCMC algorithm, criteria for Bayesian model comparison and a procedure for outlier detection. Because of space reasons, some details of the implementation were omitted in the article. These are explained below.

**Full conditionals for the Gibbs Sampler**

Define  $y = (\log(t_1), \dots, \log(t_n))'$ ,  $X = (x_1, \dots, x_n)'$  and  $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ . Regardless of the mixing distribution and provided that  $n > 2 - 2p$ , the full conditionals for  $\beta$  and  $\sigma^2$  have a known form given by  $N_k((X'X)^{-1}X'Dy, \sigma^2(X'X)^{-1})$  and inverse-Gamma( $\frac{n+2p-2}{2}, \frac{1}{2}(y - X\beta)'D(y - X\beta)$ ) distributions, respectively.

**Full conditionals for the Gibbs Sampler (log-exponential power model only)**

$$\pi(\beta|\sigma^2, \alpha, t) \propto \exp \left\{ -\frac{1}{\sigma^\alpha} \sum_{i=1}^n |\log(t_i) - x'_i \beta|^\alpha \right\}, \quad (1)$$

$$\pi(\sigma^2|\beta, \alpha, t) \propto (\sigma^2)^{-(n/2+p)} \exp \left\{ -\frac{1}{\sigma^\alpha} \sum_{i=1}^n |\log(t_i) - x'_i \beta|^\alpha \right\}, \quad (2)$$

$$\pi(\alpha|\beta, \sigma^2, t) \propto \frac{\alpha^n}{\Gamma^n(1/\alpha)} \exp \left\{ -\frac{1}{\sigma^\alpha} \sum_{i=1}^n |\log(t_i) - x'_i \beta|^\alpha \right\} \pi(\alpha). \quad (3)$$

None of these has a known form and Adaptive Metropolis-Hastings steps are implemented (Roberts and Rosenthal, 2009). Using Gaussian random walk proposals, we implement univariate samplers for each component of  $\beta$ ,  $\sigma^2$  and  $\alpha$ .

**Update of censored and set observations**

We deal with censored and set observations survival times using the data augmentation idea in Tanner and Wong (1987). Denote  $t_i^*$  the simulated survival times. Based on the mixing representation,

$$\log(T_i)|\beta, \sigma^2, \lambda_i \sim \text{Normal} \left( x'_i \beta, \frac{\sigma^2}{\lambda_i} \right). \quad (4)$$

Hence, for right-censored observations,  $\log(t_i^*)$  is simulated by truncating (4) to  $(\log(t_i), \infty)$ , where  $t_i$  is the recorded censored time. Analogously, for set observations, the range of (4) is truncated to  $(\log(t_i - \epsilon_l), \log(t_i + \epsilon_r))$ . Obviously, if  $t_i < \epsilon_l$  we take  $-\infty$  as the lower bound.

The latter is modified in case of the log-exponential power model, for which a representation based on a mixture of uniform distributions is used for the implementation. In this case,

$$\log(T_i)|\beta, \sigma^2, \alpha, u_i \sim \text{Unif} \left( x'_i \beta - \sigma u_i^{1/\alpha}, x'_i \beta + \sigma u_i^{1/\alpha} \right). \quad (5)$$

---

<sup>1</sup>Copyright (C) The R Foundation for Statistical Computing.

As in the previous case, censored observations and set observations are simulated from a truncated version of (5) to  $(\log(t_i), \infty)$  and  $(\log(t_i - \epsilon_l), \log(t_i + \epsilon_r))$ , respectively. Occasionally,  $(x'_i\beta - \sigma u_i^{1/\alpha}, x'_i\beta + \sigma u_i^{1/\alpha})$  and the truncated range for censored or set observations did not overlap for particular draws of  $\beta, \sigma^2, \alpha$  and  $u_i$ . The survival time was not updated in those iterations.

### **Update period for the mixing parameters**

The samplers presented previously update  $n$  mixing parameters at each step of the chain. This may be computationally inefficient (especially in situations in which sampling from the mixing variables is cumbersome). In order to avoid this problem, the  $\Lambda_i$ 's will be sampled only every  $Q$  iterations of the chain. The value of  $Q$  is chosen by considering the Effective Sample Size (ESS) of the chain and the CPU time required. This simplification is not applicable to the log-exponential power model because the truncation of the full conditionals of the  $U_i$ 's requires the update of these mixing parameters at each iteration of the algorithm. For the simulation study described in Section 4 and the application displayed in Section 5 of the paper, we choose  $Q = 10$  for the log-logistic model and  $Q = 1$  otherwise.

### **Detection of influential observations and outliers**

Evidence of influential observations is analysed in terms of the effect that the removal of one particular observation has on the posterior distribution of the model parameters. It is measured as the Kullback-Leiber divergence function  $K_i = KL(\pi(\beta, \sigma^2, \theta|t), \pi(\beta, \sigma^2, \theta|t_{-i}))$  (Peng and Dey, 1995; Cho et al. 2009). This is closely related to the conditional predictive ordinate (CPO) introduced in Section 3.5. In fact,

$$K_i = E_{(\beta, \sigma^2, \theta)} (\log(f(t_i|\beta, \sigma^2, \theta))) - \log(\text{CPO}_i), \quad (6)$$

and it can be easily estimated using the draws of an MCMC algorithm.

For each observation, the models  $M_0 : \lambda_i = \lambda_{ref}(u_i = u_{ref})$  and  $M_1 : \lambda_i \neq \lambda_{ref}(u_i \neq u_{ref})$  are compared. Bayes factors between these models can be computed as the generalized Savage-Dickey density ratio proposed by Verdinelli and Wasserman (1995). The latter is computationally intensive. When the parameter  $\theta$  does not appear in the model, the process is fast as it simplifies to the original version of the Savage-Dickey ratio. However, if  $\theta$  is unknown, a reduced run of the MCMC algorithm (in which  $\lambda_i$  is fixed equal to  $\lambda_{ref}$ ) is required for each observation  $i$ . Long running times are required for this. Nevertheless, as the  $n$  runs are independent, the process can be easily parallelized. Our R functions for outlier detection receive  $i$  as input in order to facilitate this. In a multi-core processor, each run can be sent to a different node.

## R code

The “SMLN codes.zip” file contains R code to implement the algorithms and Bayesian model comparison methods discussed in the paper. The code was developed in R version 2.14.1 and has been also tested in the version 3.0.1 of the software. Before using this code, the following libraries must be installed in R: `MASS`, `truncnorm`, `VGAM`, `MCMCpack`, `mvtnorm` and `compiler`. All of these are freely available from standard R repositories. The last library speeds up the “for” loops. These libraries are loaded in R when “Internal.Codes.R” is executed. Table 1 indicates the notation used throughout the code.

Table 1: Notation used throughout the R code

Variable name	Description
Time	Vector containing the survival times
Cens	Censoring indication (1: observed, 0: right-censored)
n	Total number of observations
k	Number of covariates (including intercept)
X	Design matrix with dimensions $n \times k$
N	Total number of iterations (MCMC algorithms)
thin	Thinning period (MCMC algorithms)
burn	Burn-in period (MCMC algorithms)
Q	Update period for the $\lambda_i$ 's (MCMC algorithms, except log-normal and log-exp. power models)
beta0	Starting value for $\beta$ (MCMC algorithms)
sigma20	Starting value for $\sigma^2$ (MCMC algorithms)
nu0	Starting value for $\nu$ (MCMC algorithms, log-Student's $t$ model only)
alpha0	Starting value for $\alpha$ (MCMC algorithms, log-exponential power model only)
prior	Indicator of prior (1: Jeffreys, 2: Type I Ind. Jeffreys, 3: Ind. Jeffreys)
set	Indicator for the use of set observations (1: set observations, 0: point observations)
eps_l	Lower imprecision ( $\epsilon_l$ ) for set observations (default value: 0.5)
eps_r	Upper imprecision ( $\epsilon_r$ ) for set observations (default value: 0.5)
ar	Optimal acceptance rate for the adaptive Metropolis-Hastings updates (default value: 0.44)
N_AKS	Log-logistic model only. Maximum number of terms of the Kolmogorov-Smirnov density used the rejection sampling when updating mixing parameters (default value: 3)
obs	Indicates the number of the observation under analysis (outlier detection only)
ref	Reference value $\lambda_{ref}$ or $u_{ref}$ (outlier detection only)

The code is separated into two files. The file “Internal.Codes.R” contains functions that are required for the implementation but the user is not expected to directly interact with these. These functions must be loaded in R before doing any calculations. The remaining functions are contained in the file “User.Codes.R”. The names of these functions have two components which are separated by a dot. While the first component indicates the algorithm or method implemented in the function, the second component indicates the model for which the function should be used: log-normal (LN), log-Student's  $t$  (LST), log-Laplace (LLAP), log-exponential power (LEP) and log-logistic (LLOG). In the following, a short description of these functions is provided. The use of these function is illustrated in the file “Example.R”.

- MCMC. Adaptive Metropolis-within-Gibbs algorithm with univariate Gaussian random walk proposals (Roberts and Rosenthal, 2009). The output is a matrix with  $N/\text{thin}+1$  rows and the columns are the MCMC chains for  $\beta$  ( $k$  columns),  $\sigma^2$  (1 column),  $\theta$  (1 column, if appropriate),  $\lambda$  ( $n$  columns, not provided for log-normal model),  $\log(t)$  ( $n$  columns, simulated via data augmentation) and the logarithm of the adaptive variances (the number varies among models). The latter allows the user to evaluate

if the adaptive variances have been stabilized. Overall acceptance rates are printed in the R console (if appropriate). This value should be close to the optimal acceptance rate `a.r.` For `MCMC.LEP`,  $\lambda$  is replaced by  $u$ .

For the following functions, an MCMC chain generated using the function `MCMC` (after removing the burn-in) is required as an argument.

- `DIC`. Deviance Information Criteria (Spiegelhalter et al., 2002). It is based on the deviance function  $D(\theta, y) = -2\log(f(y|\theta))$  but also incorporates a penalization factor of the complexity of the model. DIC is defined as

$$DIC \equiv E(D(\theta, y)|y) + p_D = E(D(\theta, y)|y) + (E(D(\theta, y)|y) - D(E(\theta|y), y)), \quad (7)$$

where  $p_D$  can be interpreted as the effective number of parameters of the model. This function returns a single number which is a Monte Carlo estimate of the DIC. This is computed using the marginal likelihood (after integrating out the mixing parameters) of the model. The effective and actual number of model parameters are printed in the R console.

- `LML`. Log-Marginal likelihood estimator. The marginal likelihood is defined as

$$m(t) = \int_{-\infty}^{\infty} \int_0^{\infty} \int_{\Theta} f(t|\beta, \sigma^2, \theta) \pi(\beta, \sigma^2, \theta) d\beta d\sigma^2 d\theta, \quad (8)$$

and  $\log(m(t))$  is computed using the algorithm in Chib (1995) and Chib and Jeliaskov (2001). The output is a list containing the value of the log-likelihood ordinate, log-prior ordinate, log-posterior ordinates and log-marginal likelihood estimation. Several messages are printed in order to indicate the user the status of progress of the algorithm. The user should expect particularly long running times for the log-exponential power model (all the parameters are sampled using Metropolis-Hastings updates in this case).

- `CaseDeletion`. Leave-one-out cross validation analysis. The function returns a matrix with  $n$  rows. Its first column contains the logarithm of the CPO (Geisser and Eddy, 1979). Larger values of the CPO indicate better predictive accuracy of the model. The second and third columns contain the KL divergence between  $\pi(\beta, \sigma^2, \theta|t_{-i})$  and  $\pi(\beta, \sigma^2, \theta|t)$  and its calibration index  $p_i$  (Cho et al., 2009), respectively. The later is used in order to evaluate the existence of influential observations. If  $p_i$  is substantially larger than 0.5, observation  $i$  is declared influential. Suggested cut-off values are 0.8 and 0.9. The logarithm of PsML can be computed as the sum of `logCPO` across the  $n$  observations.
- `BF.lambda.obs`. Outlier detection for observation `obs`. This returns a unique number corresponding to the Bayes Factor associated to the test  $M_0 : \Lambda_{obs} = \lambda_{ref}$  versus  $M_1 : \Lambda_{obs} \neq \lambda_{ref}$  (with all other  $\Lambda_j, j \neq obs$  free). The value of  $\lambda_{ref}$  is required as input (reference values are provided in the paper). The user should expect long running times for the log-Student's  $t$  model, in which case a reduced chain given  $\Lambda_{obs} = \lambda_{ref}$  needs to be generated (`N`, `thin`, `burn` and `Q` are required for `BF.lambda.obs.LST`).
- `BF.u.obs`. Outlier detection for observation `obs` (log-exponential power model only). This returns a unique number corresponding to the Bayes Factor associated to the test  $M_0 : U_{obs} = u_{ref}$  versus  $M_1 : U_{obs} \neq u_{ref}$  (with all other  $U_j, j \neq obs$  free). The value of  $u_{ref}$  is required as input (reference value is provided in the paper). The user should expect long running times because a reduced chain given the  $U_{obs} = u_{ref}$  needs to be generated (`N`, `thin` and `burn` are required).