Supplementary material: Bayesian inference implementation details and `R` code for "Incorporating unobserved heterogeneity in Weibull survival models: A Bayesian approach" by C.A. Vallejos and M.F.J. Steel

Bayesian inference for the AFT-RMW model under the weakly informative priors presented in Section 3.1 is implemented via an Adaptive Metropolis-within-Gibbs algorithm with Gaussian Random Walk proposals (see Section 3 in Roberts and Rosenthal, 2009). We assume right-censoring, which is most common for survival data. Mixing parameters are handled through data augmentation (Tanner and Wong, 1987). Although the usual approach for dealing for censored (and set) observations is also by data augmentation, we do not use it for that because the Weibull survival function has a known simple form (Ibrahim et al., 2001; Kottas, 2006). Inference was implemented in `R` and code is available in the 'RMWcode.zip' file. This includes the MCMC algorithm, criteria for Bayesian model comparison and a procedure for outlier detection. To save space, some details of the implementation were omitted in the article. These are explained below. Throughout, equation numbers refer to the main article.

Update of γ

In principle, γ can take any value in $(0, \infty)$. However, numerical problems are observed when very small values of γ are proposed, in which case $cv^W(\gamma)$ becomes very large (in fact, $cv^W(\gamma) \to \infty$ when $\gamma \to 0$). As a solution, regardless of the mixing distribution, we truncate the range of γ to $(0.06, \infty)$. This has no practical consequences as such small values of γ are very rarely required for real datasets.

Update of the mixing parameters

The sampler involves the update of $\Lambda_1, \ldots, \Lambda_n$ 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 mitigate this problem, the Λ_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. Even though longer chains are required for $Q > 1$ (the mixing of the chains is affected), the reduction in term of CPU time can be substantial.

For the inverse gamma and inverse Gaussian mixing distributions, the full conditionals of the mixing parameters are given by Generalized Inverse Gaussian distributions. The algorithm proposed in Leydold and Hörmann (2011) is used when sampling from these full conditionals, via the function `rgig` which is contained in the `R` library `ghyp`.

Detection of outliers and influential observations

For each observation, the models $M_0 : \Lambda_i = \lambda_{ref}$ and $M_1 : \Lambda_i \neq \lambda_{ref}$ are contrasted. Bayes factors between them can be computed as the generalized Savage-Dickey density ratio proposed by Verdinelli and Wasserman (1995) and stated in (10). When the parameter θ does not appear in the model, this simplifies to the original version of the Savage-Dickey ratio in (11). However, if θ is unknown, the procedure is computationally intensive, since a reduced run of the MCMC algorithm (in which Λ_i is fixed at λ_{ref}) is required for each observation i. 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 environment, each run can be sent to a different node.

Evidence of influential observations is evaluated through the effect on the posterior distribution when deleting one observation at the time. This evidence is quantified by means of the Kullback-Leiber divergence function $K_i = KL(\pi(\beta, \gamma, \theta|t), \pi(\beta, \gamma, \theta|t_{-i}))$. As explained in Cho et al. (2009) it can easily be computed from MCMC output. Occasionally, numerical issues can lead to a negative estimate of K_i. In such a situation, a warning is printed.

R code

The 'RMWcode.zip' file contains code to implement the algorithms and the Bayesian model comparison and outlier detection methods discussed in the article. The code was developed in `R`, version 3.0.1. First, the libraries `ghyp` and `compiler` must be installed in `R`. The last library speeds up the "for" loops. These libraries are freely available from standard `R` repositories and are loaded 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×k`
`N`	Total number of iterations (MCMC algorithms)
`thin`	Thinning period (MCMC algorithms)
`burn`	Burn-in period (MCMC algorithms)
`Q`	Update period for the λ_i's (MCMC algorithms, except unmixed exponential/Weibull model)
`beta0`	Starting value for β (MCMC algorithms)
`gam0`	Starting value for γ (MCMC algorithms)
`theta0`	Starting value for θ (MCMC algorithms, if required)
`typ.theta`	Type of prior assigned to θ. Options: `TruncExp`, `Pareto`.
`hyp.theta`	Hyper-parameter value for the prior assigned to θ.
`hyp.gam1`	Shape hyper-parameter value for the Gamma prior assigned to γ.
`hyp.gam2`	Rate hyper-parameter value for the Gamma prior assigned to γ.
`ar`	Optimal acceptance rate for the adaptive Metropolis-Hastings updates (default value: 0.44)
`EXP`	Logical indicator. If `TRUE`, the value of γ is fixed and equal to `gam0`
`obs`	Indicates the number of the observation under analysis (outlier detection only)
`ref`	Reference value λ_{ref} (outlier detection only)

The code is separated in 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: Weibull (`WEI`), RMW with exponential(1) mixing (`RMWEXP`), RMW with Gamma(θ, θ) mixing (`RMWGAM`), RMW with inv-gamma($\theta, 1$) mixing (`RMWIGAM`), RMW with inv-Gauss($\theta, 1$) mixing (`RMWIGAUSS`) and RMW with log-normal($0, \theta$) mixing (`RMWLN`). In the following, a short description of these functions is provided. The use of these function is also illustrated in the file "Example.R".

- `MCMC`. Adaptive Metropolis-within-Gibbs sampler with univariate Gaussian random walk proposals (see Section 3 in Roberts and Rosenthal, 2009). The output is a matrix with `N/thin+1` rows and the columns are the MCMC chains for β (k columns), γ (1 column), θ (1 column, if appropriate), λ (n columns, not provided for Weibull model) and the logarithm of the adaptive variances (the number varies among models). The latter allows the user to evaluate whether the adaptive variances have stabilized. Overall acceptance rates are printed in the `R` console (if appropriate). This value should be close to the optimal acceptance rate `ar`.

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

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

$$DIC \equiv E(D(\beta, \gamma, \theta, t)|t) + p_D = E(D(\beta, \gamma, \theta, t)|t) + \{E(D(\beta, \gamma, \theta, t)|t) - D(E(\beta, \gamma, \theta|t), t)\},$$

  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 likelihood marginalized w.r.t. the mixing parameters (numerical integration via the `R` function `integrate` is used for `DIC.RMWLN`). 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

$$m(t) = \int_{-\infty}^{\infty} \int_{0}^{\infty} \int_{\Theta} f(t|\beta, \gamma, \theta) \pi(\beta, \gamma, \theta)\, d\beta\, d\gamma\, d\theta,$$

  and $\log(m(t))$ is computed using the algorithm in Chib (1995) and Chib and Jeliazkov (2001). The output is a list containing the value of the log-likelihood ordinate, log-prior ordinate, log-posterior ordinate and log-marginal likelihood estimate. Several messages are printed in order to indicate the progress of the algorithm. The latter was developed for a non-adaptive scheme so we compute the marginal likelihoods from non-adaptive chains, using the stabilized proposal variances and started at the posterior median values of the adaptive chains.

- `CaseDeletion`. Leave-one-out cross-validation. The function returns a matrix with `n` rows. Its first column contains the logarithm of the CPO criterion in (9) (Geisser and Eddy, 1979; Geisser, 1993). Larger values of CPO indicate better predictive accuracy of the model. The second and third columns contain the KL divergence between $\pi(\beta, \gamma, \theta|t_{-i})$ and $\pi(\beta, \gamma, \theta|t)$ and its calibration index (Cho et al., 2009), respectively. This is used in order to evaluate the existence of influential observations. If the calibration index for observation i is much larger than 0.5, it is declared an influential observation. The logarithm of PsML can be computed as the sum of log CPO across the `n` observations.

- `BF.lambda.obs`. Outlier detection for observation `obs`. This returns a unique number corresponding to the Bayes Factor in favour of $M_0 : \Lambda_i = \lambda_{ref}$ versus $M_1 : \Lambda_i \neq \lambda_{ref}$ (with all other $\Lambda_j, j \neq i$ free). The value of λ_{ref} is required as input (references values are provided in the paper). The user should expect longer running times for RMW models containing an unknown parameter θ (*e.g.* RMW model with gamma(θ, θ) mixing), in which case n reduced chains (given fixed $\Lambda_i = \lambda_{ref}$) need to be generated (for which `N`, `thin`, `burn` and `Q` are required).

- `OD.Correction`. Correction factor applied to the reference value for right-censored observations in the outlier detection procedure implemented in `BF.lambda.obs` .

Convergence and mixing of the sampler

Convergence of the MCMC chains was never a problem with the burn-in used for the examples displayed in Section 4. Mixing is very good for models without an extra parameter θ in the mixing distribution (e.g. Weibull and RMW model with exponential(1) mixing). In the presence of an unknown θ, reliable inference is obtained through the described algorithms, but the chains are mixing a bit less well for some of the parameters, requiring MCMC run lengths of the order used here.

References

S. Chib. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, 90: 1313–1321, 1995.

S. Chib and I. Jeliazkov. Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association*, 96:270–281, 2001.

H. Cho, J. G. Ibrahim, D. Sinha, and H. Zhu. Bayesian case influence diagnostics for survival models. *Biometrics*, 65:116–124, 2009.

S. Geisser. *Predictive interference: an introduction.* Monographs on Statistics and Applied Probability Series. Chapman and Hall, 1993.

S. Geisser and W.F. Eddy. A predictive approach to model selection. *Journal of the American Statistical Association*, 74:153–160, 1979.

J.G. Ibrahim, M-H. Chen, and D. Singha. *Bayesian Survival Analysis*. Springer, 2001.

A. Kottas. Nonparametric Bayesian survival analysis using mixtures of Weibull distributions. *Journal of Statistical Planning and Inference*, 136:578–596, 2006.

J. Leydold and W. Hörmann. Generating generalized inverse Gaussian random variates by fast inversion. *Computational Statistics & Data Analysis*, 55:213–217, 2011.

G.O. Roberts and J.S. Rosenthal. Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18:349–367, 2009.

D.J. Spiegelhalter, N.G. Best, B.P. Carlin, and A. Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B*, 64:583–639, 2002.

M.A. Tanner and W.H. Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82:528–540, 1987.

I. Verdinelli and L. Wasserman. Computing Bayes factors by using a generalization of the Savage-Dickey density ratio. *Journal of the American Statistical Association*, 90:614–618, 1995.