Documentation for the R code
"Bayesian Survival Modelling of University Outcomes"
C.A. Vallejos and M.F.J. Steel

Bayesian inference is implemented through the Markov chain Monte Carlo (MCMC) sampler and priors described in Section 4. Inference was implemented in R[1] version 3.0.1. The code is freely available at `https://github.com/catavallejos/UniversitySurvival`

This includes the MCMC algorithm and the Bayesian variable selection methods described in the paper. Before using this code, the following libraries must be installed in R: `BayesLogit`, `MASS`, `mvtnorm`, `Matrix` and `compiler`. All of these are freely available from standard R repositories and are loaded in R when "Internal_Codes.R" is executed. The last two libraries speed up matrix calculations and the "for" loops, respectively. Table S1 explains the notation used throughout the code. The implementation was based on three-dimensional arrays, with the third dimension representing the event type.

Table S1: Notation used throughout the R code

Variable name	Description
`CATEGORIES`	Number of possible outcomes, excluding censoring (equal to 3 for the PUC dataset)
`n`	Total number of students
`nt`	Total number of multinomial outcomes (i.e. $\sum t_i$ across all students)
`t0`	Number of period-specific baseline log-odds coefficients δ_{rt} (for each cause)
`k`	Number of effects (`t0` + number of covariate effects)
`Y`	Vector of outcomes. Dimension: `n` $\times 1$
`X`	Design matrix, including the binary indicators (denoted by Z in the paper). Dimension: `n` $\times$ `k`
`X.Period`	Design matrix related to period-specific baseline log-odds coefficients δ_{rt}'s only. Dimension `nt` $\times$ `t0`
`inc`	Vector containing covariate indicators $\gamma_1, \ldots, \gamma_{k*}$
`beta`	β^* (period-specific baseline log-odds and covariates effects for all event types)
`mean.beta`	Prior mean for $\{\beta_1^*, \ldots, \beta_{\mathcal{R}}^*\}$. Dimension: $1\times$ `k` $\times$ `CATEGORIES`
`prec.delta`	Precision matrix for $(\delta_{r1}, \ldots, \delta_{rt_0})'$. Dimension: `t0` $\times$ `t0`
`df.delta`	Degrees of freedom for prior of $(\delta_{r1}, \ldots, \delta_{rt_0})'$. Default value: 1
`fix.g`	If `TRUE`, $g_1, \ldots, g_{\mathcal{R}}$ are fixed. Default value: `FALSE`
`prior`	Choice of hyper prior for g_r: *(i)* Benchmark-Beta or *(ii)* Hyper-g/n (see **?**)
`N`	Total number of MCMC iterations
`thin`	Thinning period for MCMC algorithm
`burn`	Burn-in period for MCMC algorithm
`beta0`	Starting value for $\{\beta_1^*, \ldots, \beta_{\mathcal{R}}^*\}$. Dimension: $1\times$ `k` $\times$ `CATEGORIES`
`logg0`	Starting value (log-scale) of $\{g_1, \ldots, g_{\mathcal{R}}\}$. Dimension: $1\times$ `CATEGORIES`
`ls.g0`	Starting value (log-scale) of the adaptive proposal variance used in Metropolis-Hastings updates of $\log(g_1), \ldots, \log(g_{\mathcal{R}})$. Dimension: $1\times$ `CATEGORIES`
`ar`	Optimal acceptance rate for the adaptive Metropolis-Hastings updates. Default value: 0.44
`ncov`	Indicates how many potential covariates are included in the design matrix (might not match the number of columns due to categorical covariates with more than two levels) Default value: 8 (as in the PUC dataset)
`include`	Vector indicating which covariates from the design matrix are to be included in the model If missing a sampler over the model space will be run. Default: `NULL`
`gamma`	γ (covariate inclusion indicators)

The code is separated into two files. The file "Internal_Codes.R" contains functions that are required for

[1]Copyright (C) The R Foundation for Statistical Computing.

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 main function — used to run the MCMC algorithm — is contained in the file "User_Codes.R". In the following, a short description of this function is provided. Its use is illustrated in the file "Example.R" using a simulated dataset.

- `MCMC.MLOG`. Adaptive Metropolis-within-Gibbs algorithm [2] for the competing risks Proportional Odds model used throughout the paper. If not fixed, univariate Gaussian random walk proposals are implemented for $\log(g_1), \ldots, \log(g_{\mathcal{R}})$. Arguments: `N`, `thin`, `Y`, `X`, `t0`, `beta0`, `mean.beta`, `prec.delta`, `df.delta`, `logg0`, `ls.g0`, `prior`, `ar`, `fix.g`, `ncov` and `include`. The output is a list containing the following elements: `beta` MCMC sample of β^* (array of dimension (`N/thin+1`) $\times$ `k` $\times$ `CATEGORIES`), `gamma` MCMC sample of γ (matrix of dimension (`N/thin+1`) $\times$ `k`, `logg` MCMC sample of $\log(g_1), \ldots, \log(g_{\mathcal{R}})$ (dimension (`N/thin+1`) $\times$ `CATEGORIES`), `ls.g` stored values for the logarithm of the proposal variances for $\log(g_1), \ldots, \log(g_{\mathcal{R}})$ (dimension (`N/thin+1`) $\times$ `CATEGORIES`) and `lambda` MCMC sample of $\lambda_1, \ldots, \lambda_{\mathcal{R}}$), which are defined in equation (9) in the paper (dimension (`N/thin+1`) $\times$ `CATEGORIES`). Recording `ls.g` 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 `ar`.

[2] Roberts and Rosenthal, 2009, *Journal of Computational and Graphical Statistics*