Bayesian Methods - Ex3c - Joint Model (LME + PH)

Jan Vávra

Full assignment in PDF

This assignment is supposed to be solved via JAGS and `library(runjags)`. List through the manual to find what you need.

Data and model description

We will work with the `aids` data from `library(JM)`, see `help(aids)` for more details. It is of both longitudinal and survival data nature. This time we will try to model them **simultaneously**.

```
set.seed(123456789)
library(JM)
head(aids[,c("patient", "Time", "death", "CD4", "obstime", "drug", "start", "stop", "event")], 10)
```

```
##    patient  Time death       CD4 obstime drug start  stop event
## 1        1 16.97     0 10.677078       0  ddC     0  6.00     0
## 2        1 16.97     0  8.426150       6  ddC     6 12.00     0
## 3        1 16.97     0  9.433981      12  ddC    12 16.97     0
## 4        2 19.00     0  6.324555       0  ddI     0  6.00     0
## 5        2 19.00     0  8.124038       6  ddI     6 12.00     0
## 6        2 19.00     0  4.582576      12  ddI    12 18.00     0
## 7        2 19.00     0  5.000000      18  ddI    18 19.00     0
## 8        3 18.53     1  3.464102       0  ddI     0  2.00     0
## 9        3 18.53     1  3.605551       2  ddI     2  6.00     0
## 10       3 18.53     1  6.164414       6  ddI     6 18.53     1
```

It consists of 467 HIV infected patients who were treated with two antiretroviral drugs (`drug`).

We will model the evolution of `CD4` cell count for each individual `patient`, as in Exercise 3a. We assume LME model with random intercept and slope:

- $Y_{ij} = m_i(t_{ij}) + \varepsilon_{ij} = B_{1i} + B_{2i}t_{ij} + \varepsilon_{ij}$, is the `CD4` cell count of patient i at visit j,
- $\varepsilon_{ij} \sim \mathsf{N}(0, \tau^{-1})$ is the iid model error,
- t_{ij} is the time of the observation of patient i at visit j,
- D_i is the indicator of `drug` level `ddI`,
- B_{1i} and B_{2i} are the random intercept and slope for patient i,
- $\mathbf{B}_i = (B_{1i}, B_{2i})^\top \sim \mathsf{N}_2\left(\begin{pmatrix}\beta_1 \\ \beta_2 + \beta_3 D_i\end{pmatrix}, \Omega^{-1}\right)$, where Ω is general positive-definite *precision* matrix.

According to this model, the expected `CD4` cell count $m_i(t)$ evolves linearly with time for each patient differently $m_i(t) = B_{1i} + B_{2i}t$.

Cox proportional hazards model can be estimated only under the assumption of **piece-wise constant** `CD4` cell count:

```
library(survival)
fitcoxph <- coxph(Surv(start, stop, event) ~ drug + CD4, data = aids)
summary(fitcoxph)
```

This assumption is unrealistic. We can expect some gradual change in time between visits. This motivates us to use $m_i(t)$ as a covariate to be used within (Cox) proportional hazards model. Then, the hazard function for patient i becomes complicated by t appearing within the exponential factor:

$$\begin{aligned} h_i(t) &= h_0(t)\, \exp\{\gamma_1 D_i + \gamma_2 m_i(t)\} \\ &= h_0(t)\, \exp\{\gamma_1 D_i + \gamma_2 B_{1i} + t \cdot \gamma_2 B_{2i}\} \end{aligned}$$

When we choose $h_0(t) = \alpha t^{\alpha-1}$, we **no longer obtain Weibull distribution**! The distributional family is given by viewing $h_i(t)$ as a function of t, which would conceptually yield

$$h_i(t) \propto t^{\psi_i} \exp\{\zeta_i\, t\}, \qquad H_i(t) = \text{const.} \int_0^t s^{\psi_i} \exp\{\zeta_i\, s\}\, \mathrm{d}s,$$

which could be viewed as a generalization of Weibull. Unfortunately, the implementation of JAGS does not cover this distribution.

For simplification, let us assume $\alpha = 1$, which yields constant $h_0(t)$. This option usually results in exponential distribution. However, the exponential term with t changes the distribution to something different. Conceptually, we have the (cumulative) hazard function of the form

$$h_i(t) = \xi\, \exp\{a_i + t\, b_i\}, \qquad H_i(t) = \xi\, \frac{\exp\{a_i\}}{b_i}\, (\exp\{b_i t\} - 1)\,,$$

which is how Gompertz distribution is defined. Sadly, this distribution is not implemented in JAGS as well. However, since $\log h_i(t)$ and $H_i(t)$ can be expressed in closed formula, we can fit the model using **zero-Poisson trick**. We only need to supply our own implementation of the corresponding log-likelihood.

From NMST511 Course Notes by Theorem 2.2 under the assumption of censoring time C_i independent of time to death T_i the log-likelihood under presence of right-censoring indicated by $\delta_i = \mathbf{1}(T_i \le C_i)$ takes the following form:

$$\ell(\boldsymbol{\theta}) = \text{const.} + \sum_{i=1}^{n} \left[\delta_i \log h_i(X_i, \boldsymbol{\theta}) - H_i(X_i, \boldsymbol{\theta})\right],$$

where $X_i = \min\{T_i, C_i\}$ is the event time (`Time`) and $\boldsymbol{\theta}$ is the vector of unknown parameters.

How to tell JAGS to work with a custom likelihood? Using zero-Poisson trick Consider a random variable $O \sim \mathsf{Pois}(\phi)$ with $\phi > 0$. Then, $\mathsf{P}(O = 0) = \exp\{-\phi\}$ and the contribution to log-likelihood when $O = 0$ is $-\phi$. We just need to set $-\phi$ to be the contribution to log-likelihood we desire. There is minor issue with the requirement that ϕ has to be positive. If we set up

$$\phi = -\text{loglik} + C,$$

where C is sufficiently large constant to make (all) ϕ positive. Shifting each contribution to log-likelihood by the same constant does not have any effect to it.

Alltogether, the pseudocode to be used in JAGS with right-censoring is below:

```
"model{
  C <- 10^5

  ...

  for(i in 1:n){
    ...
    logh[i] <- ...
    H[i] <- ...
    loglik[i] <- delta[i] * logh[i] - H[i]
```

```
    phi[i] <- C - loglik[i]
    zeros[i] ~ dpois(phi[i])
  }

  ...

}
"
```

Variable `zeros` is a vector of n zeros to be given as data. Constant C can be given as data in advance as well.

Task 1 - JAGS implementation of Joint model

Extend the JAGS code from Exercise 3a by the survival model with Gompertz hazard function $h_i(t) = \xi \exp\{a_i + t\, b_i\}$ outlined above using the zero-Poisson trick.

Assume the independent block structure of the prior for model parameters:

$$p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \xi, \tau, \Omega) = \prod_{j=1}^{3} p(\beta_j) \prod_{j=1}^{2} p(\gamma_j)\, p(\xi)\, p(\tau)\, p(\Omega)$$

Choose weakly informative *normal* prior for β_j and γ_j, *gamma* prior for τ and ξ, *Wishart* distribution (`dwish`) for Ω.

Write down (and print) the model implementation within JAGS.

Task 2 - Running JAGS

Sample (at least) two Markov chains using JAGS to approximate the posterior distribution $p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \xi, \tau, \Omega \,|\, \texttt{data})$. Be very careful about initial values, some may lead to unstable chains. Choose appropriate `burnin` and `thin` by monitoring the trajectories and autocorrelation.

Task 3 - Monte Carlo estimates

Provide summaries including ET and HPD intervals for primary model parameters. Monitor also standard deviations of random effects and their correlation.

Task 4 - Prediction for two patients

Explore and plot characteristics of the posterior distribution (posterior mean or median and credible intervals) of the following parametric functions: $m_{\text{new}}(t)$, $h_{\text{new}}(t)$, $H_{\text{new}}(t)$, $S_{\text{new}}(t)$ for two newly observed patients with *average* evolution of `CD4` each treated with different drug.

BONUS Task - piecewise constant baseline hazard function

Instead of assuming constant baseline hazard function $h_0(t) = \xi$ use

$$h_0(t) = \prod_{k=1}^{K} \xi_k^{\mathbf{1}(t_k \le t < t_{k+1})},$$

where $0 = t_1 < t_2 < \cdots < t_{K+1} = \infty$ form K predefined intervals, on which we have different baseline hazards ξ_k. Use intervals defined by empirical quantiles:

```
(breaks <- c(0, quantile(data$Time, probs = seq(0,1,length.out = 8))[2:7], Inf))
```

```
##           14.28571% 28.57143% 42.85714% 57.14286% 71.42857% 85.71429%
##  0.000000  6.227143 11.078571 12.530000 13.930000 15.970000 17.800000       Inf
```

where `data` is a `data.frame` containing only one row per patient.