Hierarchical Jump-point PLP (JPLP) simulation

Miao Cai miao.cai@slu.edu

2020-04-22

Contents

1 Bayesian Hierarchical Jump Power Law Process (JPLP)

1.1 Model setting

The Bayesian hierarchical JPLP model is parameterized as

$$\begin{aligned}
t_{d,s,1}, t_{d,s,2}, \cdots, t_{d,s,n_{d,s}} &\sim \text{JPLP}(\beta, \theta_{d,s}, \kappa) \\
\beta &\sim \text{Gamma}(1, 1) \\
\log \theta_{d,s} &= \gamma_{0d} + \gamma_1 x_{d,s,1} + \gamma_2 x_{d,s,2} + \cdots + \gamma_k x_{d,s,k} \\
\kappa &\sim \text{Uniform}(0, 1) \\
\gamma_{01}, \gamma_{02}, \cdots, \gamma_{0D} &\sim \text{i.i.d. } N(\mu_0, \sigma_0^2) \\
\gamma_1, \gamma_2, \cdots, \gamma_k &\sim \text{i.i.d. } N(0, 10^2) \\
\mu_0 &\sim N(0, 5^2) \\
\sigma_0 &\sim \text{Gamma}(1, 1),
\end{aligned} \tag{1}$$

where the introduced parameter κ is the percent of intensity function recovery once the driver takes a break. By definition, $a_{d,s,0} = 0$. We assume that this κ is constant across drivers and shifts.

1.2 Intensity function of JPLP

Since the Bayesian hierarchical PLP in Subsection **??** does not account for the rests within a shift and associated potential reliability repairment. In this subsection, we proposes a Bayesian hierarchical JPLP, with the following intensity function:

$$\lambda_{\text{JPLP}}(t|d,s,r,\beta,\gamma_{0,d},\gamma,\mathbf{X}_d,\mathbf{W}) = \begin{cases} \kappa^0\lambda(t|\beta,\gamma_{0,d},\gamma,\mathbf{X}_d,\mathbf{W}) & 0 < t \leq a_{d,s,1} \\ \kappa^1\lambda(t|\beta,\gamma_{0,d},\gamma,\mathbf{X}_d,\mathbf{W}) & a_{d,s,1} < t \leq a_{d,s,2} \\ \cdots & \cdots \\ \kappa^{R-1}\lambda(t|\beta,\gamma_{0,d},\gamma,\mathbf{X}_d,\mathbf{W}) & a_{d,s,R-1} < t \leq a_{d,s,R} \end{cases}$$
$$= \kappa^{r-1}\lambda(t|d,s,r,\kappa,\beta,\gamma_{0,d},\gamma,\mathbf{X}_d,\mathbf{W}) \quad a_{d,s,r-1} < t \leq a_{d,s,r}, \tag{2}$$

The notations are identical with those in Equation **??** except for the extra κ parameter.

1.3 The likelihood function of JPLP

The likelihood function for driver d on shift s is

$$\begin{aligned} L_{s,d}(\kappa,\beta,\gamma_{0,d},\gamma|\text{Data}_{d,s}) = & \Big(\prod_{i=1}^{c_{d,s}}\lambda\big(t_{i,d,s}|d,s,r,k,\beta,\gamma_{0,d},\gamma,\mathbf{X}_d,\mathbf{W}\big)\Big) \\ & \times \exp\Big(-\int_0^{a_{d,s,r}\lambda\big(u|d,s,r,k,\beta,\gamma_{0,d},\gamma,\mathbf{X}_d,\mathbf{W}\big)du}\Big) \end{aligned} \tag{3}$$

The overall likelihood function is

$$L = \prod_d\prod_{s\in d}L_{s,d} \tag{4}$$

2 Simulating data

- Parameters needed: $\kappa,\beta,\theta,\gamma_{0,d},\gamma$
  - $\theta \leftarrow \gamma_{0,d},\gamma,\mathbf{X}$
- Data needed: $\mathbf{X}$
  - x_1,x_2,x_3

```
pacman::p_load(rstan, tidyverse, data.table)
#source("functions/JPLP_functions.R")

set.seed(123)
D = 10 # the number of drivers
K = 3 # the number of predictor variables

# 1. Random-effect intercepts
# hyperparameters
mu0 = 0
sigma0 = 0.5
r_0D = rnorm(D, mean = mu0, sd = sigma0)

# 2. Fixed-effects parameters
R_K = rnorm(K, mean = 0, sd = 0.5)

# 3. The number of observations (shifts) in the $d$-th driver: $N_{d}$
N_K = rpois(D, 10)
```

```r
N = sum(N_K) # the total number of obs
id = rep(1:D, N_K)

# 4. Generate data: x_1, x_2, .. x_K
simX = function(group_sizes = N_K){
  ntot = sum(group_sizes)

  int1 = rep(1, ntot)
  x1 =  rnorm(ntot, 0, 1)
  x2 = rgamma(ntot, 1, 1)
  x3 =  rpois(ntot, 0.2)

  return(data.frame(int1, x1, x2, x3))
}
X = simX(N_K)

# 5. Scale parameters of a JPLP
# 5a. parameter matrix: P
P = cbind(r0 = rep(r_0D, N_K), t(replicate(N, R_K)))
M_logtheta = P*X

# returned parameter for each observed shift
beta = 1.5
kappa = 0.8
theta = exp(rowSums(M_logtheta))
```

Simulated parameters:

- κ: 0.8
- β: 1.5
- θ:
- $\gamma_{0,d}$:
- γ: 0.6120409, 0.1799069, 0.2003857

Simulated data:

```r
str(X)
```

```
## 'data.frame':    82 obs. of  4 variables:
##  $ int1: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ x1  : num  -1.687 0.838 0.153 -1.138 1.254 ...
##  $ x2  : num  0.302 0.613 1.947 0.381 0.15 ...
##  $ x3  : int  0 1 1 0 0 0 0 2 0 0 ...
```