

# Bayesian hierarchical models for NHPP using **rstan**

*Miao Cai miao.cai@slu.edu*

*2019-07-09*

## Contents

<b>1</b>	<b>Model setting</b>	<b>1</b>
<b>2</b>	<b>Simulating data</b>	<b>2</b>
<b>3</b>	<b>Estimation</b>	<b>4</b>

## 1 Model setting

Let  $T_{d,s,i}$  denote the time to the  $d$ -th driver's  $s$ -th shift's  $i$ -th critical event. The total number critical events of  $d$ -th driver's  $s$ -th shift is  $n_{d,s}$ . The ranges of these notations are:

- $i = 1, 2, \dots, n_{d,S_d}$ ,
- $s = 1, 2, \dots, S_d$ ,
- $d = 1, 2, \dots, D$ .

We assume the times of critical events within the  $d$ -th driver's  $s$ -th shift were generated from a non-homogeneous Poisson process (NHPP) with a power law process (PLP), with a fix rate parameter  $\beta$  and varying scale parameters  $\theta_{d,s}$  across drivers. The data generating process is then:

$$T_{d,s,1}, T_{d,s,2}, \dots, T_{d,s,n_{d,s}} \sim \text{PLP}(\beta, \theta_{d,s})$$

$$\beta \sim \text{Gamma}(1, 1)$$

$$\log \theta_{d,s} = \gamma_{0d} + \gamma_1 x_{d,s,1} + \gamma_2 x_{d,s,2} + \dots + \gamma_k x_{d,s,k}$$

$$\gamma_{01}, \gamma_{02}, \dots, \gamma_{0D} \sim \text{i.i.d. } N(\mu_0, \sigma_0^2)$$

$$\gamma_1, \gamma_2, \dots, \gamma_k \sim \text{i.i.d. } N(0, 10^2)$$

$$\mu_0 \sim N(0, 10^2)$$

$$\sigma_0 \sim \text{Gamma}(1, 1)$$

## 2 Simulating data

1. Random intercepts  $\gamma_{01}, \gamma_{02}, \dots, \gamma_{0D}$ . The standard deviation of  $\mu_0$  was intentionally set to small number 2 to make  $\theta_{d,s}$  fall into a reasonably small range. If I otherwise set it as 10,  $\theta_{d,s}$  may be more than  $10^5$  due to the exponentiation, which may not be realistic in real-life data.

$$\mu_0 = 1, \quad \sigma_0 = 1$$

$$\sigma_0 \sim \text{Gamma}(1, 1)$$

$$\gamma_{01}, \gamma_{02}, \dots, \gamma_{0D} \sim \text{i.i.d. } N(\mu_0, \sigma_0^2)$$

2. fixed parameters: 3 fixed parameters  $\gamma_1, \gamma_2, \gamma_3$ .

$$\gamma_1, \gamma_2, \gamma_3 \sim \text{i.i.d. } N(0, 10^2)$$

3. The number of observations in the  $d$ -th driver:  $N_d$ .

$$N_d \sim \text{Poisson}(100)$$

4. Data: 3 predictor variables  $x_{d,s,1}, x_{d,s,2}, x_{d,s,3}$ .

$$x_{d,s,1} \sim N(0, 10)$$

$$x_{d,s,2} \sim \text{Gamma}(10, 2)$$

$$x_{d,s,3} \sim \text{Poisson}(3.5)$$

5. Scale parameters of a NHPP (random effects):  $\theta_{d,s}$ .

$$\theta_{d,s} = \text{EXP}(\gamma_{0d} + \gamma_1 x_{d,s,1} + \gamma_2 x_{d,s,2} + \gamma_3 x_{d,s,3})$$

6. Shape parameter of a NHPP (fixed effect):  $\beta \sim \text{Gamma}(1, 1)$ . Set

$$\beta = 1.5$$

7. Simulate a NHPP based on  $\beta$  and  $\theta_{d,s}$ .

$$T_{d,s,1}, T_{d,s,2}, \dots, T_{d,s,n_{d,s}} \sim \text{PLP}(\beta, \theta_{d,s})$$

```

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

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

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

# 2. Fixed-effects parameters
R_K = rnorm(K, mean = 0, sd = 1)

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

# 4. Generate data:  $x_1, x_2, \dots, x_K$ 
sim1 = function(n = 10){
  x1 = rnorm(n, 0, 5)
  x2 = rgamma(n, 5, 1)
  x3 = rpois(n, 3.5)
  return(data.frame(x1, x2, x3))
}
simXD = function(ndrivers = D){
  XD = rep(list(data.frame()), ndrivers)
  for (i in 1:D) {
    XD[[i]] = sim1(N_K[i])
  }
  return(data.table::rbindlist(XD, idcol = "driver"))
}
X = simXD()

# 5. Scale parameters of a NHPP
# 5a. parameter matrix:  $P$ 
N_D = X[,.N,driver][["N"]]# N by driver
N_all = sum(N_D) # total N
P = cbind(r0 = rep(r_OD, N_D),
          t(replicate(N_all, R_K)))
M_logtheta = P*X

theta = exp(rowSums(M_logtheta))
beta = 1.5

```

### 3 Estimation

```
plptauML = '  
functions{  
  real nhpp_log(vector t, real beta, real theta, real tau){  
    vector[num_elements(t)] loglik_part;  
    real loglikelihood;  
    for (i in 1:num_elements(t)){  
      loglik_part[i] = log(beta) - beta*log(theta) + (beta - 1)*log(t[i]);  
    }  
    loglikelihood = sum(loglik_part) - (tau/theta)^beta;  
    return loglikelihood;  
  }  
}  
data {  
  int<lower=0> D; //driver index  
  int<lower=0> N; //total # of obs  
  int<lower=0> K; //total # of shifts  
  vector<lower=0>[K] tau;//truncated time  
  vector<lower=0>[N] event_time; //failure time  
  int s[K]; //group sizes  
}  
parameters{  
  real<lower=0> beta;  
  vector[K] r0; // random intercept  
  vector[3] r; // fixed parameters  
  real mu0; // hyperparameter  
  real<lower=0> sigma0;// hyperparameter  
}  
transformed parameters{  
  vector<lower=0>[K] theta;  
  
  for (k0 in 1:K)  
    theta[k0] = r0[D[k0]] + x1[i]*r[1] + x2[i]*r[2] + x3[i]*r[3];  
}  
model{  
  int position;  
  position = 1;  
  for (k in 1:K){  
    segment(event_time, position, s[k]) ~ nhpp(beta, theta[k], tau[k]);  
    position = position + s[k];  
  }  
//PRIORS  
  beta ~ gamma(1, 1);  
  r0 ~ normal(mu0, sigma0);  
  r ~ normal(0, 10);  
  mu0 ~ normal(0, 10);  
  sigma0 ~ gamma(1, 1);  
  theta ~ gamma(1, 0.01);  
}  
,
```

In Stan code:

- Need a data matrix  $X$ ,
- Need matrix multiplication,

In data:

- Need a driver index  $d = 1, 2, \dots, K$  for each shift  $k$
- Need a data matrix  $X$