Hierarchical Jump-point PLP (JPLP) Estimation

Miao Cai miao.cai@slu.edu

2020-05-25

Contents

1 Theory

1.1 Intensity function of JPLP

We proposes a Bayesian hierarchical JPLP, with the following piecewise intensity function:

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

where the introduced parameter κ is the percent of intensity function recovery once the driver takes a break, and $a_{d,s,r}$ is the end time of trip r within shift s for driver d. By definition, the end time of the 0-th trip $a_{d,s,0} = 0$, and the end time of the last trip for the d-driver within s-shift $a_{d,s,R}$ equals the shift end time $\tau_{d,s}$. We assume that this κ is constant across drivers and shifts.

1.2 Parameterization of JPLP

The Bayesian hierarchical JPLP model is parameterized as

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

1.3 Likelihood function of JPLP

The likelihood function of event times generated from a JPLP for driver d on shift s is

$$
\begin{aligned}
&L_{s,d}^*(\kappa, \beta, \gamma_{0d}, \gamma | \mathbf{X}_d, \mathbf{W}_s) = \\
&\qquad \begin{cases}
\exp\Big(-\int_0^{a_{d,s,R}} \lambda_{\text{JPLP}}\big(u|d, s, r, k, \beta, \gamma_{0d}, \gamma, \mathbf{X}_d, \mathbf{W}_s\big)du\Big), & \text{if } n_{d,s} = 0, \\
\Big(\prod_{i=1}^{n_{d,s}} \lambda_{\text{JPLP}}\big(t_{i,d,s}|d, s, r, k, \beta, \gamma_{0d}, \gamma, \mathbf{X}_d, \mathbf{W}_s\big)\Big) & \\
\times \exp\Big(-\int_0^{a_{d,s,R}} \lambda_{\text{JPLP}}\big(u|d, s, r, k, \beta, \gamma_{0d}, \gamma, \mathbf{X}_d, \mathbf{W}_s\big)du\Big), & \text{if } n_{d,s} > 0,
\end{cases}
\end{aligned}
\tag{3}
$$

where $t_{i,d,s}$ is the time to the i-th SCE for driver d on shift s, $n_{d,s}$ is the number of SCEs for driver d on shift s. Therefore, the overall likelihood function for drivers $d \in 1, 2, \ldots, D$ and their corresponding shifts $s \in d$ is:

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

Since λ_{JPLP} is a piecewise likelihood function that depends on event time and trip time, we will not spell out the details of the full likelihood or log likelihood.

2 One driver scenario

2.1 Simulating data for multiple shifts from one driver

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

set.seed(123)
dt = sim_mul_jplp(kappa = 0.8, beta = 1.5, theta = 2, n_shift = 500)
str(dt)
```

```
## List of 4
##  $ event_dt:'data.frame':    4061 obs. of  3 variables:
##   ..$ shift_id  : int [1:4061] 1 1 1 1 1 1 1 1 2 2 ...
##   ..$ trip_id   : int [1:4061] 1 1 1 2 3 3 3 4 1 1 ...
```

```
##   ..$ event_time: num [1:4061] 1.7 1.76 2.76 3.26 3.98 ...
##  $ trip_dt :'data.frame':    1758 obs. of  5 variables:
##   ..$ shift_id    : int [1:1758] 1 1 1 1 2 2 3 3 3 3 ...
##   ..$ trip_id     : int [1:1758] 1 2 3 4 1 2 1 2 3 4 ...
##   ..$ t_trip_start: num [1:1758] 0 2.87 3.85 7.53 0 5.42 0 2.25 4.43 7.34 ...
##   ..$ t_trip_end  : num [1:1758] 2.87 3.85 7.53 9.27 5.42 ...
##   ..$ N_events    : int [1:1758] 3 1 3 1 6 5 0 1 1 2 ...
##  $ shift_dt:'data.frame':    500 obs. of  3 variables:
##   ..$ shift_id  : int [1:500] 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ start_time: num [1:500] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ end_time  : num [1:500] 9.27 9.42 9.93 11.39 9.8 ...
##  $ stan_dt :List of 7
##   ..$ N           : int 4061
##   ..$ S           : int 1758
##   ..$ r_trip      : int [1:1758] 1 2 3 4 1 2 1 2 3 4 ...
##   ..$ t_trip_start: num [1:1758] 0 2.87 3.85 7.53 0 5.42 0 2.25 4.43 7.34 ...
##   ..$ t_trip_end  : num [1:1758] 2.87 3.85 7.53 9.27 5.42 ...
##   ..$ event_time  : num [1:4061] 1.7 1.76 2.76 3.26 3.98 ...
##   ..$ group_size  : int [1:1758] 3 1 3 1 6 5 0 1 1 2 ...
```

2.2 Estimating JPLP using Stan

```r
fit = stan("stan/jplp_simple.stan",
           chains = 1, iter = 1000, refresh = 0,
           data = dt$stan_dt, seed = 123)
broom::tidy(fit)
```

```
## # A tibble: 3 x 3
##   term  estimate std.error
##   <chr>    <dbl>     <dbl>
## 1 beta     1.49     0.0302
## 2 theta    2.01     0.0540
## 3 kappa    0.807    0.0154
```

3 Multiple drivers scenario

3.1 Simulating data for multiple shifts from multiple drivers

```r
set.seed(123)
dt1 = sim_hier_JPLP(beta = 1.2,
                    kappa = 0.8,
                    mu0 = 0.2,
                    sigma0 = 0.5,
                    R_K = c(1, 0.3, 0.2),
                    group_size_lambda = 50,
                    D = 10)
str(dt1)
```

```
## List of 4
##  $ event_time:'data.frame':  1763 obs. of  3 variables:
##   ..$ driver_id : int [1:1763] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ shift_id  : int [1:1763] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ event_time: num [1:1763] 0.955 4.091 5.954 6.774 7.281 ...
##  $ trip_time :'data.frame':  1790 obs. of  6 variables:
##   ..$ driver_id   : int [1:1790] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ shift_id    : int [1:1790] 1 1 1 1 2 2 2 2 2 3 ...
##   ..$ trip_id     : int [1:1790] 1 2 3 4 1 2 3 4 5 1 ...
##   ..$ t_trip_start: num [1:1790] 0 3.68 6.8 8.97 0 1.91 4.28 5.87 7.83 0 ...
##   ..$ t_trip_end  : num [1:1790] 3.68 6.8 8.97 12.3 1.91 ...
##   ..$ N_events    : int [1:1790] 1 3 2 5 0 2 1 0 4 0 ...
##  $ shift_time:'data.frame':  507 obs. of  6 variables:
##   ..$ driver_id : int [1:507] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ shift_id  : int [1:507] 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ start_time: num [1:507] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ end_time  : num [1:507] 12.3 10.1 12.2 10.6 10.1 ...
##   ..$ n_trip    : num [1:507] 4 5 2 2 2 5 5 5 2 5 ...
##   ..$ n_event   : int [1:507] 11 7 5 0 10 7 2 0 1 7 ...
##  $ stan_dt   :List of 11
##   ..$ N           : int 1763
##   ..$ K           : num 3
##   ..$ S           : int 1790
##   ..$ D           : num 10
##   ..$ id          : int [1:1790] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ r_trip      : int [1:1790] 1 2 3 4 1 2 3 4 5 1 ...
##   ..$ t_trip_start: num [1:1790] 0 3.68 6.8 8.97 0 1.91 4.28 5.87 7.83 0 ...
##   ..$ t_trip_end  : num [1:1790] 3.68 6.8 8.97 12.3 1.91 ...
##   ..$ event_time  : num [1:1763] 0.955 4.091 5.954 6.774 7.281 ...
##   ..$ group_size  : int [1:1790] 1 3 2 5 0 2 1 0 4 0 ...
##   ..$ X_predictors: num [1:1790, 1:3] 0.6651 0.6651 0.6651 0.6651 -0.0857 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:1790] "1" "1.1" "1.2" "1.3" ...
##   .. .. ..$ : chr [1:3] "x1" "x2" "x3"
```

3.2 Estimating JPLP using Stan

```
fit1 = stan("stan/jplp_hierarchical.stan",
            chains = 1, iter = 1000, refresh = 0,
            data = dt1$stan_dt, seed = 123)

pull_use(var = "beta|kappa|mu0_true|sigma0|R1_K", fit1)
```

```
## # A tibble: 7 x 3
##   term     estimate std.error
##   <chr>       <dbl>     <dbl>
## 1 sigma0      0.485    0.141
## 2 beta        1.24     0.0329
```

```
## 3 kappa       0.792    0.0237
## 4 R1_K[1]     0.990    0.0325
## 5 R1_K[2]     0.311    0.0333
## 6 R1_K[3]     0.199    0.0161
## 7 mu0_true    0.263    0.182
```