Hierarchical Jump-point PLP (JPLP) Estimation

Miao Cai miao.cai@slu.edu

2020-05-20

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

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

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)
source("functions/JPLP_functions.R")

dt = sim_mul_jplp()
str(dt)
```

```
## List of 3
##  $ event_time:'data.frame':  112 obs. of  2 variables:
##   ..$ shift_id  : int [1:112] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ event_time: num [1:112] 0.627 0.677 1.436 2.521 4.572 ...
##  $ trip_time :'data.frame':  27 obs. of  2 variables:
```

```
##   ..$ shift_id : int [1:27] 1 1 1 2 2 2 2 3 3 3 ...
##   ..$ trip_time: num [1:27] 1.98 4.85 7.52 2.73 5.27 ...
##  $ shift_time:'data.frame':  10 obs. of  3 variables:
##   ..$ shift_id  : int [1:10] 1 2 3 4 5 6 7 8 9 10
##   ..$ start_time: num [1:10] 0 0 0 0 0 0 0 0 0 0
##   ..$ end_time  : num [1:10] 9.53 12.47 9.59 10.62 9.99 ...
```

2.2 Estimating JPLP using Stan

```
fit = stan("stan/jplp_simple.stan",
           chains = 1, iter = 1000, refresh = 0,
           data = z$hier_dat, seed = 123)
```

3 Multiple drivers scenario

3.1 Simulating data for multiple shifts from multiple drivers

3.2 Estimating JPLP using Stan

```
fit = stan("stan/jplp_hierarchical.stan",
           chains = 1, iter = 1000, refresh = 0,
           data = z$hier_dat, seed = 123)
```