PLP inference for data generated from a JPLP process

Miao Cai

2020-05-25

Contents

This vignette considers the scenario where the data is generated from a JPLP process, but statistical inference is made assuming a PLP.

1 Generate data

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

df = sim_hier_JPLP(D = 10, beta = 1.2)
str(df$stan_jplp_dt_for_plp)
```

```
## List of 9
##  $ N           : int 517
##  $ K           : num 3
##  $ S           : int 106
##  $ D           : num 10
##  $ id          : int [1:106] 1 1 1 1 1 1 1 1 1 1 ...
##  $ tau         : num [1:106] 11.08 10.28 9.78 8.61 9.39 ...
##  $ event_time  : num [1:517] 5.96 7.63 6.89 2.44 3.57 ...
##  $ group_size  : int [1:106] 0 2 1 1 2 0 0 0 5 0 ...
##  $ X_predictors:'data.frame':    106 obs. of  3 variables:
##   ..$ x1: num [1:106] -0.269 1.084 0.916 1.395 2.197 ...
##   ..$ x2: num [1:106] 0.351 1.983 1.173 0.651 0.435 ...
##   ..$ x3: int [1:106] 7 0 6 2 2 3 1 1 3 3 ...
```

2 Estimating using Stan

```
fit0 = stan("stan/nhppnoevent_lp.stan", chains = 1, iter = 1000,
         data = df$stan_jplp_dt_for_plp, refresh = 0)
```

```r
pull_use(var = "beta|kappa|mu0_true|sigma0|R1_K", fit0)
```

```
## # A tibble: 6 x 3
##   term     estimate std.error
##   <chr>       <dbl>     <dbl>
## 1 sigma0     0.820     0.215
## 2 beta       1.12      0.0478
## 3 R1_K[1]    1.14      0.0784
## 4 R1_K[2]    0.270     0.0804
## 5 R1_K[3]    0.223     0.0328
## 6 mu0_true  -0.0481    0.305
```