元器件生存时间模拟分析-代码

Miao Cai

2019-04-23

目录

附录: R及Stan代码

定义函数

```r
set.seed(123)

alpha = matrix(c(0.001, 0.05, 0.0001, 0.08), ncol = 2, byrow = TRUE)
w = c(45, 55)
t = matrix(seq(10, 40, 10), ncol = 2, byrow = TRUE)
K = c(10, 50, 100)


lambda = function(r = 1:2, j = 1:2) {
  alpha_r0 = alpha[r, 1]
  alpha_r1 = alpha[r, 2]
  return(alpha_r0*exp(alpha_r1*w[j]))
}

p = function(i = 1:2, j = 1:2){
  lam1j = lambda(1, j) # failure due to factor 1
  lam2j = lambda(2, j) # failure due to factor 2

  p0 = exp(-(lam1j + lam2j)*t[i, j])
  p1 = (lam1j/(lam1j + lam2j))*(1 - p0)
  p2 = (lam2j/(lam1j + lam2j))*(1 - p0)

  return(c(p0, p1, p2))
}

SimMatrix = function(N = 10, N_device = 10){
  T1 = matrix(rep(-1, 3*N), ncol = 3, byrow = TRUE)
  T2 = T1; T3 = T1; T4 = T1;
  T1 = t(rmultinom(N, N_device, prob = p(1, 1)))
  T2 = t(rmultinom(N, N_device, prob = p(1, 2)))
  T3 = t(rmultinom(N, N_device, prob = p(2, 1)))
  T4 = t(rmultinom(N, N_device, prob = p(2, 2)))

  return(list(T1, T2, T3, T4))
}
```

```r
DAT0 = SimMatrix(100, 100)


create_standat = function(N = 10, N_device = 10){
  dat_list = SimMatrix(N, N_device)
  t[2,1] = t[2,1] - t[1,2] + t[1,2]*lambda(1, 1)/lambda(1, 2)
  t[2,2] = t[2,2] - t[1,2] + t[1,2]*lambda(2, 1)/lambda(1, 2)
  stan_dat = list(
    n = nrow(dat_list[[1]]),
    DAT1 = dat_list[[1]],
    DAT2 = dat_list[[2]],
    DAT3 = dat_list[[3]],
    DAT4 = dat_list[[4]],
    W = w,
    Tim = t
  )
  return(stan_dat)
}

post_result = function(N_sim = 10, N = 10, N_device = 10){
  tab_result = matrix(rep(NA, 12*N_sim), byrow = TRUE, ncol = 12)

  for (i in 1:N_sim) {
    stan_dat = create_standat(N, N_device)
    fit00 <- stan(
      model_code = reliabMAC, data = stan_dat,
      warmup = 500, iter = 1000, chains = 1, cores = 1, seed = i)
    tab_result[i,1:4] = summary(fit00)$summary[1:4,"mean"]
    tab_result[i,5:8] = summary(fit00)$summary[1:4,"n_eff"]
    tab_result[i,9:12] = summary(fit00)$summary[1:4,"Rhat"]
  }

  return(tab_result)
}
```

Stan代码

```
library(rstan)
rstan_options(auto_write = TRUE)
options(scipen = 99)

reliabMAC = "
data {
  int n;
  int DAT1[n, 3];
  int DAT2[n, 3];
  int DAT3[n, 3];
  int DAT4[n, 3];
  vector[2] W;
  matrix[2, 2] Tim;
}
parameters{
  real<lower=0, upper=0.1> a10;
  real<lower=0, upper=0.1> a11;
  real<lower=0, upper=0.1> a20;
  real<lower=0, upper=0.1> a21;
}
transformed parameters{
  simplex[3] p11;
  simplex[3] p12;
  simplex[3] p21;
  simplex[3] p22;

  p11[1] = exp(-(a10*exp(a11*W[1]) + a20*exp(a21*W[1]))*Tim[1, 1]);
  p11[2]  = (a10*exp(a11*W[1]))/(a10*exp(a11*W[1]) + a20*exp(a21*W[1]))*(1 - p11[1]);
  p11[3]  = 1-p11[1]-p11[2];

  p12[1] = exp(-(a10*exp(a11*W[2]) + a20*exp(a21*W[2]))*Tim[1, 2]);
  p12[2]  = (a10*exp(a11*W[2]))/(a10*exp(a11*W[2]) + a20*exp(a21*W[2]))*(1 - p12[1]);
  p12[3]  = 1-p12[1]-p12[2];

  p21[1] = exp(-(a10*exp(a11*W[1]) + a20*exp(a21*W[1]))*Tim[2, 1]);
  p21[2]  = (a10*exp(a11*W[1]))/(a10*exp(a11*W[1]) + a20*exp(a21*W[1]))*(1 - p21[1]);
  p21[3]  = 1-p21[1]-p21[2];
  p22[1] = exp(-(a10*exp(a11*W[2]) + a20*exp(a21*W[2]))*Tim[2, 2]);
  p22[2]  = (a10*exp(a11*W[2]))/(a10*exp(a11*W[2]) + a20*exp(a21*W[2]))*(1 - p22[1]);
  p22[3]  = 1-p22[1]-p22[2];
  //p12 = [p0, p1, p2]';
  //p21 = [p0, p1, p2]';
  //p22 = [p0, p1, p2]';
}
model{
  for (i in 1:n){
    //target += multinomial_lpmf(DAT[i,] | p);
    DAT1[i,] ~ multinomial(p11);
    //DAT[i,] ~ multi_log(p0, p1, p2);
  }
  for (i in 1:n){
    DAT2[i,] ~ multinomial(p12);
  }
  for (i in 1:n){
    DAT3[i,] ~ multinomial(p21);
  }
  for (i in 1:n){
    DAT4[i,] ~ multinomial(p22);
  }
  a10 ~ uniform(0, 0.1);
  a11 ~ uniform(0, 0.1);
  a20 ~ uniform(0, 0.1);
  a21 ~ uniform(0, 0.1);
}
"
stan_dat = create_standat(1, 10)
fit_example <- stan(
  model_code = reliabMAC, data = stan_dat,
  warmup = 500, iter = 1000, chains = 1, cores = 1, seed = 123)
```

正式模拟

模拟 A1

```
set.seed(123)
library(data.table)
alpha = matrix(c(0.001, 0.05, 0.0001, 0.08), ncol = 2, byrow = TRUE)

A1_10 = post_result(100, 1, 10)
A1_50 = post_result(100, 1, 50)
A1_100 = post_result(100, 1, 100)

data.table::fwrite(as.data.frame(A1_10), 'data/20190409serve200/A1_10.csv')
data.table::fwrite(as.data.frame(A1_50), 'data/20190409serve200/A1_50.csv')
data.table::fwrite(as.data.frame(A1_100), 'data/20190409serve200/A1_100.csv')


## 模拟 A2
set.seed(123)
alpha = matrix(c(0.005, 0.05, 0.0005, 0.08), ncol = 2, byrow = TRUE)

A2_10 = post_result(100, 1, 10)
A2_50 = post_result(100, 1, 50)
A2_100 = post_result(100, 1, 100)

data.table::fwrite(as.data.frame(A2_10), 'data/20190409serve200/A2_10.csv')
data.table::fwrite(as.data.frame(A2_50), 'data/20190409serve200/A2_50.csv')
data.table::fwrite(as.data.frame(A2_100), 'data/20190409serve200/A2_100.csv')


## 模拟 A3
set.seed(123)
alpha = matrix(c(0.008, 0.05, 0.0008, 0.08), ncol = 2, byrow = TRUE)

A3_10 = post_result(100, 1, 10)
A3_50 = post_result(100, 1, 50)
A3_100 = post_result(100, 1, 100)

data.table::fwrite(as.data.frame(A3_10), 'data/20190409serve200/A3_10.csv')
data.table::fwrite(as.data.frame(A3_50), 'data/20190409serve200/A3_50.csv')
```

```r
data.table::fwrite(as.data.frame(A3_100), 'data/20190409serve200/A3_100.csv')
```

生成结果报告

```
pacman::p_load(data.table, tidyverse)



report_tab = function(dat, alpha){
  dat_result = rbindlist(list(dat[,c(1, 5, 9)],
                 dat[,c(2, 6, 10)],
                 dat[,c(3, 7, 11)],
                 dat[,c(4, 8, 12)]))
  dat_result[, alpha_id := c(replicate(100, 'a01'),
                      replicate(100, 'a11'),
                      replicate(100, 'a20'),
                      replicate(100, 'a21'))]
  dat_result[, true_est := rep(alpha, each = 100)]
  names(dat_result) = c('estimate', 'n_eff', 'Rhat', 'alpha_id', 'true_est')

  final_dat = dat_result[,.(Bias = 1/100*sum(abs(estimate - true_est)),
            MSE = 1/100*sum((estimate - true_est)^2),
            n_eff = mean(n_eff),
            Rhat = mean(Rhat)), by = alpha_id]

  return(final_dat)
}

# A1
A1_10 = fread("data/20190409serve200/A1_10.csv")
A1_50 = fread("data/20190409serve200/A1_50.csv")
A1_100 = fread("data/20190409serve200/A1_100.csv")

A1_10 %>% as.data.frame() %>%
  report_tab(c(0.001, 0.05, 0.0001, 0.08)) %>%
  fwrite('data/20190409serve200/reportA1_10.csv')
A1_50 %>% as.data.frame() %>%
  report_tab(c(0.001, 0.05, 0.0001, 0.08)) %>%
  fwrite('data/20190409serve200/reportA1_50.csv')
A1_100 %>% as.data.frame() %>%
  report_tab(c(0.001, 0.05, 0.0001, 0.08)) %>%
  fwrite('data/20190409serve200/reportA1_100.csv')
```

```r
# A2
A2_10 = fread("data/20190409serve200/A2_10.csv")
A2_50 = fread("data/20190409serve200/A2_50.csv")
A2_100 = fread("data/20190409serve200/A2_100.csv")

A2_10 %>% as.data.frame() %>%
  report_tab(c(0.005, 0.05, 0.0005, 0.08)) %>%
  fwrite('data/20190409serve200/reportA2_10.csv')
A2_50 %>% as.data.frame() %>%
  report_tab(c(0.005, 0.05, 0.0005, 0.08)) %>%
  fwrite('data/20190409serve200/reportA2_50.csv')
A2_100 %>% as.data.frame() %>%
  report_tab(c(0.005, 0.05, 0.0005, 0.08)) %>%
  fwrite('data/20190409serve200/reportA2_100.csv')

# A3
A3_10 = fread("data/20190409serve200/A3_10.csv")
A3_50 = fread("data/20190409serve200/A3_50.csv")
A3_100 = fread("data/20190409serve200/A3_100.csv")

A3_10 %>% as.data.frame() %>%
  report_tab(c(0.008, 0.05, 0.0008, 0.08)) %>%
  fwrite('data/20190409serve200/reportA3_10.csv')
A3_50%>% as.data.frame() %>%
  report_tab(c(0.008, 0.05, 0.0008, 0.08)) %>%
  fwrite('data/20190409serve200/reportA3_50.csv')
A3_100 %>% as.data.frame() %>%
  report_tab(c(0.008, 0.05, 0.0008, 0.08)) %>%
  fwrite('data/20190409serve200/reportA3_100.csv')
```

轨迹图示例

```r
require(rstan)
rstan_options(auto_write = TRUE)

# A1
set.seed(123)
alpha = matrix(c(0.001, 0.05, 0.0001, 0.08), ncol = 2, byrow = TRUE)
stan_dat = create_standat(10, 100)

fit01 <- stan(
  model_code = reliabMAC, data = stan_dat,
  warmup = 500, iter = 4000, chains = 3, cores = 3, seed = 1)
saveRDS(fit01, 'data/20190409serve200/fit01.Rds')

fit01 = readRDS('data/20190409serve200/fit01.Rds')
plot(fit01, plotfun = 'trace', pars = c('a10', 'a11', 'a20', 'a21'))
ggplot2::ggsave('data/20190409serve200/sample trace01.png',
                dpi = 300, width = 10, height = 6.18)


# A2
set.seed(123)
alpha = matrix(c(0.005, 0.05, 0.0005, 0.08), ncol = 2, byrow = TRUE)
stan_dat = create_standat(10, 100)

fit02 <- stan(
  model_code = reliabMAC, data = stan_dat,
  warmup = 500, iter = 4000, chains = 3, cores = 3, seed = 1)
saveRDS(fit02, 'data/20190409serve200/fit02.Rds')

fit02 = readRDS('data/20190409serve200/fit02.Rds')
plot(fit02, plotfun = 'trace', pars = c('a10', 'a11', 'a20', 'a21'))
ggplot2::ggsave('data/20190409serve200/sample trace02.png',
                dpi = 300, width = 10, height = 6.18)



# A3
set.seed(123)
```

```
alpha = matrix(c(0.008, 0.05, 0.0008, 0.08), ncol = 2, byrow = TRUE)
stan_dat = create_standat(10, 100)

fit03 <- stan(
  model_code = reliabMAC, data = stan_dat,
  warmup = 500, iter = 4000, chains = 3, cores = 3, seed = 1)
saveRDS(fit03, 'data/20190409serve200/fit03.Rds')

fit03 = readRDS('data/20190409serve200/fit03.Rds')
plot(fit03, plotfun = 'trace', pars = c('a10', 'a11', 'a20', 'a21'))
ggplot2::ggsave('data/20190409serve200/sample trace03.png',
                dpi = 300, width = 10, height = 6.18)
```