

06Conquer_hierarchical_NHPP_no_events

Miao Cai miao.cai@slu.edu

2019-07-16

Contents

1	Revised code	2
2	Estimation results - 500 simulations for each sample size	3
2.1	beta	3
2.2	mu0	4
2.3	sigma0	5
2.4	Fixed parameters	6

1 Revised code

```
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> N; //total # of obs
  int<lower=0> K; //total # of shifts
  int<lower=0> D[K]; //driver index, this must be an array
  vector<lower=0>[K] tau; //truncated time
  vector<lower=0>[N] event_time; //failure time
  int s[K]; //group sizes
  vector[K] x1;
  vector[K] x2;
  vector[K] x3;
}

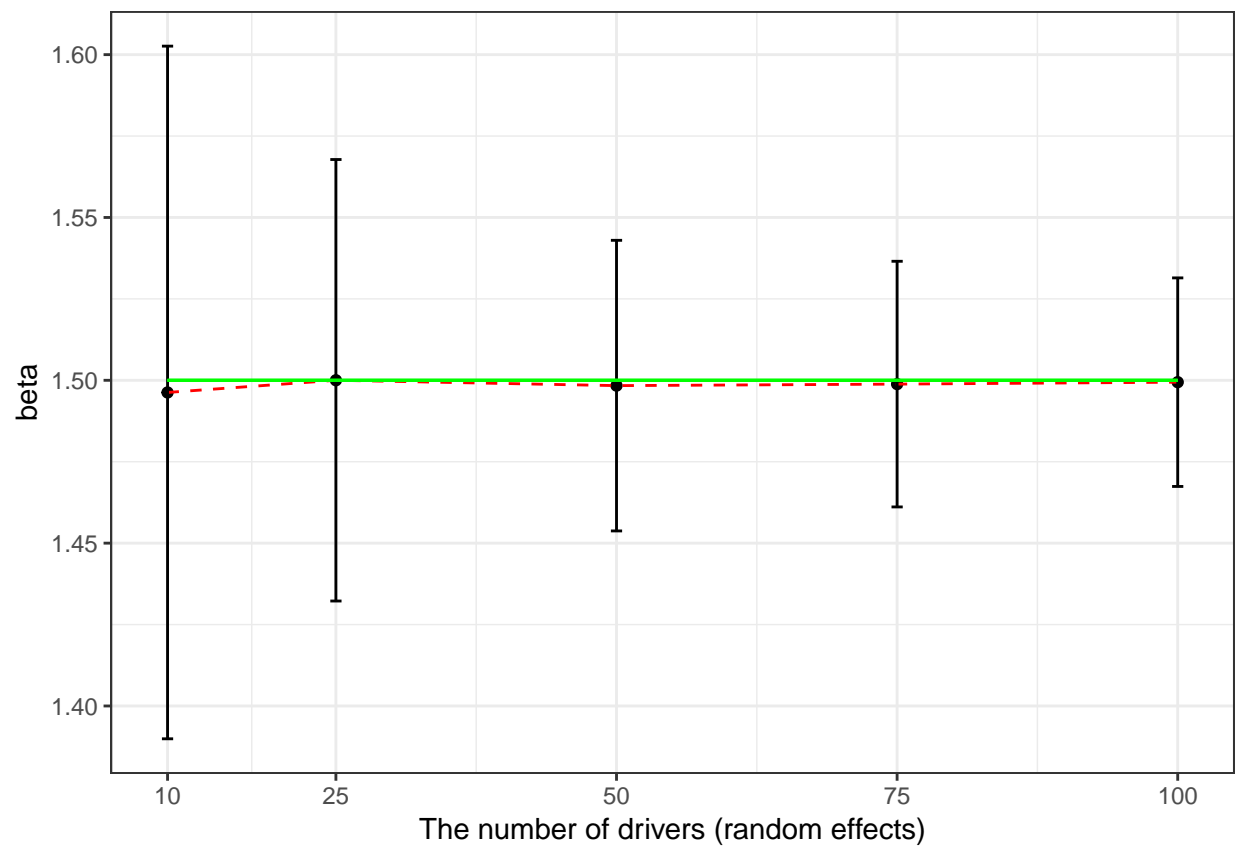
parameters{
  real<lower=0> beta;
  vector[K] r0; // random intercept
  vector[3] r; // fixed parameters
  real mu0; // hyperparameter
  real<lower=0> sigma0; // hyperparameter
}

transformed parameters{
  vector<lower=0>[K] theta;
  for (k0 in 1:K){
    theta[k0] = exp(r0[ D[k0] ] + x1[k0]*r[1] + x2[k0]*r[2] + x3[k0]*r[3]);
  }
}

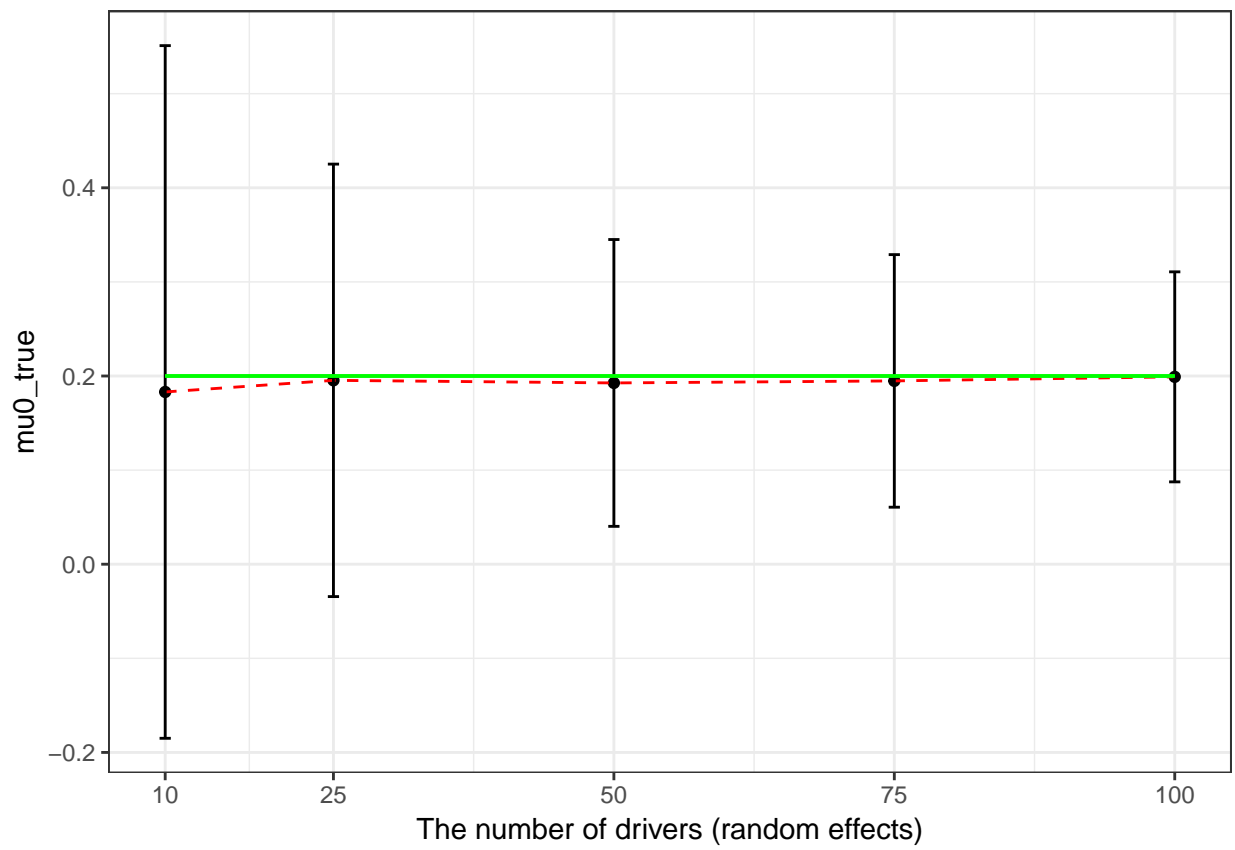
model{
  int position;
  position = 1;
  for (k in 1:K){
    if(s[k] == 0) continue;
    segment(event_time, position, s[k]) ~ nhpp(beta, theta[k], tau[k]);
    position = position + s[k];
  }
  beta ~ gamma(1, 1);
  r0 ~ normal(mu0, sigma0);
  r ~ normal(0, 10);
  mu0 ~ normal(0, 10);
  sigma0 ~ gamma(1, 1);
  theta ~ gamma(1, 0.01);
}
```

2 Estimation results - 500 simulations for each sample size

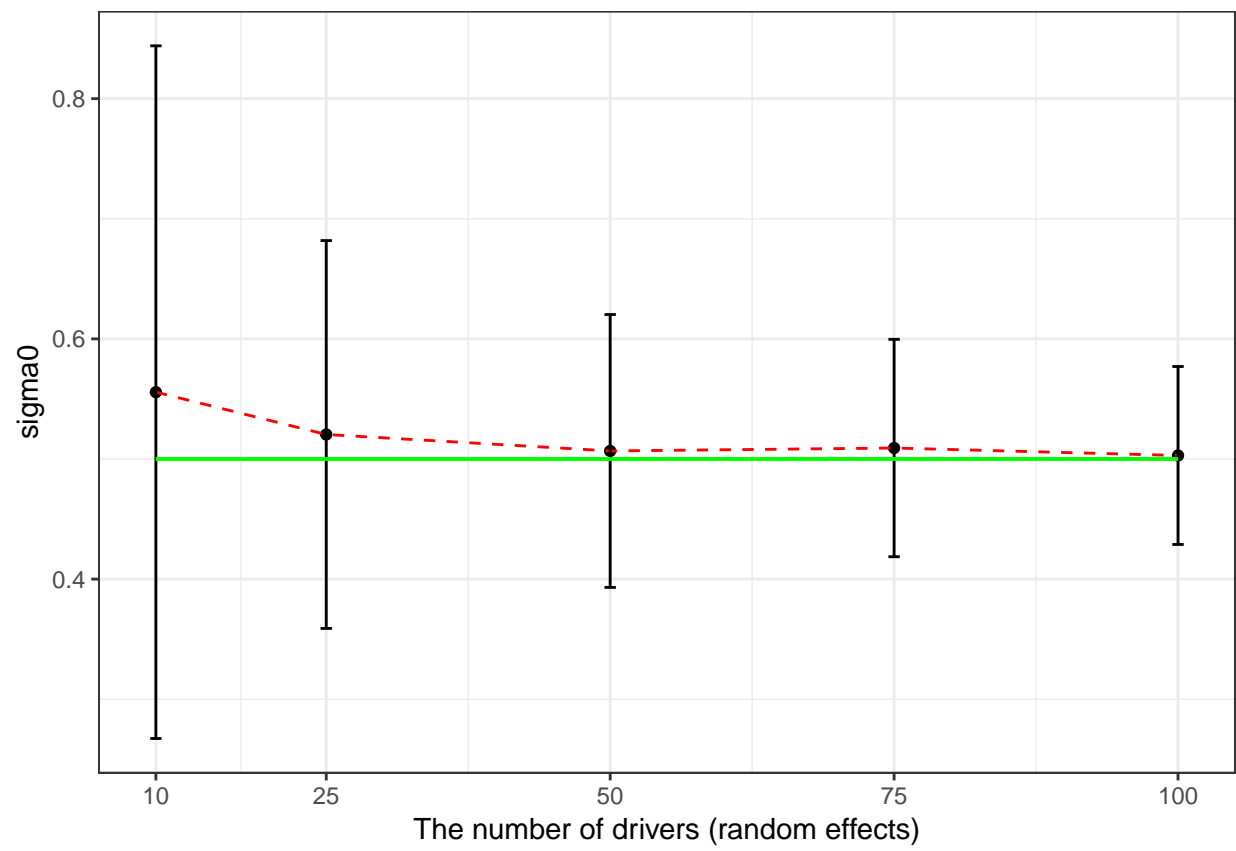
2.1 beta



2.2 μ_0



2.3 sigma0



2.4 Fixed parameters

