Exercise 4: Mixed Effects Model

Jason Doll

June 8, 2018

Objective

This exercise will demonstrate divergent transitions in mixed effects models and how to fix them. You will apply skills learned in the previous example to address divergent transitions in a mxied effects model. The skills learned in this exercise can be applied to any model where you are incorporating a random effect and expereince divergent transitions.

Background

Previously, you learned about divergent transitions, how to detect them, and how to fix them. This exercise will demonstrate a common model used in most fields where divergent transtions are likely arise. In this exercise, you will learn how to program a mixed effects model in Stan using the centered parameterization and how to reparamiterize the mixed effect model to a non-centered parameterizaion to fix the divergent transtions. The R code to initialize workspace, generate data, and package and send the data with initial values to Stan are provided in the "Ex4_RE.R" file located in the "Ex4_RE_Divergent" folder. The instructor will review this R file with you. Your task will be to create two Stan files using the code provided below. You are encouraged to type rather than copy/paste the code into a new .stan document. The first file describes a model that produces the divergent transitions, parametrization described as "centered", and the second file uses reparametrizes the model to be describe the same model but does not produce divergent transitions, this parametrization is described as "non-centered". The model you will fit to individual observations i and group j is:

$$
\begin{aligned}
\text{Model:} \qquad & \mathrm{y}_i = \alpha_{j[i]} + \beta \mathrm{X}_i + \epsilon_i \\
& \epsilon_i \sim \text{Normal}(0, \sigma) \\
\text{Priors:} \qquad & \alpha_j \sim \text{Normal}(\mu, \tau) \\
& \mu \sim \text{Normal}(0, 100) \\
& \tau \sim \text{half-cauchy}(0, 5) \\
& \beta \sim \text{Normal}(0, 100) \\
& \sigma \sim \text{half-cauchy}(0, 5)
\end{aligned}
$$

R packages required for this exercise

1. rstan
2. ggplot2
3. shinystan

Directions

Create a new text file and save it as "Ex4_Cent_RE.stan" in the "Ex4_RE_Divergent" folder. Enter the Stan code below in the "Ex4_Cent_RE.stan" file.

Stan code for centered parameterization

```
data{
  int<lower=0> n; //number of observations
  vector[n] y;    //observations as a vector
  vector[n] x;     //observed x values, predictors
  int gp[n];      //subject indicator
  int ngps;       //number of groups
}

parameters{
  real<lower=0> sigma_y;   //individual standard deviation to be estimated
  real<lower=0> sigma_mu;  //group standard deviation to be estimated
  real gpmu[ngps];         //Group mean
  real mu;                 //Global mean across all groups
  real beta;               //slope
}

model {
  //reference priors
  sigma_y ~ cauchy(0,5);
  sigma_mu ~ cauchy(0,5);
  beta ~ normal(0,100);
  mu ~ normal(0,100);

  for(k in 1:ngps){
    gpmu[k] ~ normal(mu,sigma_mu); //individual group means
  }

  //likelihood, loop through number of observations
  for(i in 1:n){
    y[i] ~ normal(gpmu[gp[i]] + beta * x[i], sigma_y);
  }
}
```

R Code

Open "Ex4_RE.R" and run lines 1 through lines:

```r
#Centered Example with Divergent Transitions
Center_RE <- stan(file = 'Ex4_Cent_RE.stan',
             data = dataList ,
             init = initslst,
             chains = nchains,
             iter = 1000 ,
             warmup = 500 ,
             thin = 2,
             control = list(adapt_delta=0.99))
```

Traceplots and parameter estimates can be viewed using the code below (note this code is provided). The offending parameter is the SD from the random effects, sigma_mu. Note the divergent transitions that are flagged with a red color in the graphs creted in the shinystan object.

```r
traceplot(Center_RE,pars=c("sigma_mu"))
#Shinystan
launch_shinystan(Center_RE)
```

Stan code for non-centered parameterization

Create a second new text file and save it as "Ex4_Non_Cent_RE.stan" in the "Ex4_RE_Divergent" folder. Enter the Stan code below in the "Ex4_Non_Cent_RE.stan" file. The instructor will review the code modifications in the workshop.

```
data{
  int<lower=0> n; //number of obserations
  vector[n] y;    //observations as a vector
  vector[n] x;     //observed x values, predictors
  int gp[n];      //subject indicator
  int ngps;       //number of groups
}

parameters{
  real<lower=0> sigma_y;   //individual standard deviation to be estimated
  real<lower=0> sigma_mu;  //group standard deviation to be estimated
  real gpmu_raw[ngps];     //Group mean_raw
  real mu;                 //Global mean across all groups
  real beta;               //slope
}

transformed parameters{
  real gpmu[ngps];              //Group mean

  for(k in 1:ngps){
    gpmu[k] = mu + gpmu_raw[k] * sigma_mu;
  }
}

model {
  //reference priors
  sigma_y ~ cauchy(0,5);
  sigma_mu ~ cauchy(0,5);
  beta ~ normal(0,100);
  mu ~ normal(0,100);

  for(k in 1:ngps){
    gpmu_raw[k] ~ normal(0,1);//implies gpmu[k] ~ normal(mu,sigma_mu); //individual group mean
  }

  //likelihood, loop through number of observations
  for(i in 1:n){
    y[i] ~ normal(gpmu[gp[i]] + beta * x[i], sigma_y);
  }
}
```

R Code

Open "Ex4_RE.R" and run the lines below:

```r
nchains = 3
dataList = list(
  n= (gp*n),
  y = data1$y,
  x=data1$x,
  gp = data1$subject,
  ngps = gp
)

#Initialize values
#convergence can be improved by setting reasonable starting values
#i.e, range of observations from 1-20, don't intialize mean at 100000
#Use different starting values for each chain
initslst <- lapply(1:nchains,function(i) {
  list(
    sigma_y=runif(1,1,10),
    sigma_mu=runif(1,1,10),
    mu=runif(1,min(y),max(y)),
    beta=runif(1,-10,10),
    gpmu_raw = rnorm(gp,0,1)
  )
})

#Noncentered Example without Divergent transitions
non_Center_RE <- stan(file = 'Ex4_Non_Cent_RE.stan',
                  data = dataList ,
                  init = initslst,
                  chains = nchains,
                  iter = 1000 ,
                  warmup = 500 ,
                  thin = 2,
                  control = list(adapt_delta=0.99))
```

Traceplots and parameter estimates can be viewed using the code below (note this code is provided). The offending parameter is the SD from the random effects, sigma_mu. Note the divergent transitions are no longer appearing. However, there remains some pattern in the sigma_mu vs _lp parameter, suggesting more work should be done to improve convergence. Some steps to be taken could be an informative prior or a different prior probability distribution.

```r
traceplot(non_Center_RE,pars=c("sigma_mu"))
#Shinystan
launch_shinystan(non_Center_RE)
```