Exercise 3: Divergent Transitions

Jason Doll

June 8, 2018

Objective

This execise will demonstrate divergent transitions, a common error that is returned when using Stan, and how to address this error. This exercise will use concepts describe in the workshop to detect and fix divergent transtions. The skills learned here will be applied to a mixed effect model later, to demonstrate a practical application. This exercise will also introduce the shinystan R package.

Background

In this example, we will use Stan to generate data based on fixed parameters of a normal distribution which return divergent transtions. Code and concepts are pulled from the Stan users manual. The R code to initialize workspace, generate data, and package and send the data with initial values to Stan are provided in the "Ex3_Div.R" file located in the "Ex3_DivergentTrans" 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 is 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".

R packages required for this exercise

1. rstan
2. shinystan

Directions

Create a new text file and save it as "Ex3_Cent.stan" in the "Ex3_DivergentTrans" folder. Enter the Stan code below in the "Ex3_Cent.stan" file.

Stan code for centered parameterization

```
parameters{
  real y;
  vector[9] x;
}
model {
  y~normal(0,3);
  x~normal(0,exp(y/2));
}
```

R Code

Open “Ex3_Div.R” and run lines 1 through lines:

```r
#Centered Example with Divergent Transitions
Centered <- stan(file = 'Ex3_Cent.stan',
                 init = initslst,
                 chains = nchains,
                 iter = 4000 ,
                 warmup = 2000 ,
                 thin = 1 )
```

Traceplots and parameter estimates can be viewed using the code below (note this code is provided). We will compare the results here with the results from the non-centered parameterization at the end of this exercise:

```r
#View traceplots
traceplot(Centered,pars=c("y","x[1]"))
#view results
Centered
```

Stan code for non-centered parameterization

Create a second new text file and save it as “Ex3_Non_Cent.stan” in the “Ex3_DivergentTrans” folder. Enter the Stan code below in the “Ex3_Non_Cent.stan” file. The instructor will review the code modifications during the workshop.

```stan
parameters {
  real y_raw;
  vector[9] x_raw;
}
transformed parameters {
  real y;
  vector[9] x;
  y = 3.0 * y_raw;
  x = exp(y/2) * x_raw;
}
model {
  y_raw ~ normal(0, 1); // implies y ~ normal(0, 3)
  x_raw ~ normal(0, 1); // implies x ~ normal(0, exp(y/2))
}
```

R Code

Open “Ex3_Div.R” and run the lines below:

```r
#specify number of chains, used to initialize values and specify chains
nchains = 3

#Initialize values
initslst <- lapply(1:3,function(i) {
  list(
    y=runif(1,0,1),
    x=runif(9,0,1)
  )
})
```

```
#NOncentered Example without Divergent transitions
non_centered <- stan(file = 'Ex3_Non_Cent.stan',
                       init = initslst,
                       chains = nchains,
                       iter = 4000 ,
                       warmup = 2000 ,
                       thin = 1 )
```

Traceplots and parameter estimates can be viewed using the code below (note this code is provided).

```
#View traceplots
traceplot(non_centered,pars=c("y","x[1]"))
#view results
non_centered
```

Graph the estimated y values to observe the “funnel” that is produced by the model.

```
#Compare y values
par(mfrow=c(2,1))
 hist(cent_df$y,breaks=25,xlim=c(-10,10), main = c("Centered"))
 hist(noncent_df$y,breaks=25,xlim=c(-10,10), main = c("Non-Centered"))
par(mfrow=c(1,1))
```

Use the shinystan pacakge to view results of each model to diagnose error. Run each line below separately. Your browser will be opened to display the output of the shinystan object. Your instructor will guide you through the output.

```
#Diagnose with Shinystan
launch_shinystan(Centered)

launch_shinystan(non_centered)
```