Introduction to rstan: workshop

Ian Laga

What is Stan?

- Stan is, among other things, a modeling language:
    - Bayesian inference with MCMC samplings via NUTS, HMC (No-u-turns Hamiltonian Monte Carlo)
    - Approximate Bayesian inference with variational inference
    - Penalized maximum likelihood estimation with optimization

- Similar to BUGS

- Rstan is just a way to run Stan using R code
    - The same files can be run in python, shell, MATLAB, etc

What will we cover?

- Packages we need and initialization
- How to use rstan directly
- How to check diagnostics
- How to handle rstan objects
- How to avoid using rstan
    - brms (my favorite)
    - rstanarm (pretty good)

Downloading packages

- Need:
  - rstan
  - parallel
- Want:
  - rstanarm
  - brms
  - shinystan

Getting Started

- options(mc.cores = parallel::detectCores())
  - This tells rstan to use as many cores as your computer has available

- rstan_options(auto_write = TRUE)
  - Don’t have to recompile stan files

- Sys.setenv(LOCAL_CPPFLAGS = '-march=native')
  - Allows faster executive time, must may cause problems (I believe this is no longer recommended for R version 4.0)

rstan package

- “User-facing R functions are provided to parse, compile, test, estimate, and analyze Stan models by accessing the header-only Stan library provided by the 'StanHeaders' package.”

1. Write your model in a .stan file
2. Prepare the data
3. Call stan
4. Diagnose non-convergence
5. Conduct posterior inference

.stan file

- A .stan file entirely specifies your Bayesian model
- There are three required components:

1. “data” block
   - Specify all observed data in the model, including the dimensions of the data and parameters

2. “parameters” block
   - Specify all parameters you want to sample

3. “model” block
   - Specify the priors on your parameters and distributions on your data

Example Problem

Assume you observe $\boldsymbol{Y} = (y_1, \ldots, y_N),\ y_i \sim Exponential(\lambda)$

You place a half-normal prior on μ with standard deviation 10, i.e.

$$\pi(\lambda) = \frac{\sqrt{2}}{10\sqrt{\pi}} \exp\left(-\frac{\lambda^2}{200}\right), \lambda \geq 0$$

“data” block

```
data {
     int<lower=0> N; // Declare the dimension of your
                              //observations
     vector[N] y; // Declare your observations
}
```

“parameters” block

```
parameters {
    real<lower=0> lambda; // Declare your parameter you
                          //want to sample from
}
```

- Notice that we specified that lambda must be positive

“model” block

- This is where all the magic happens

```
model {
     lambda ~ normal(0, 10) // Declare your prior
     y ~ exponential(lambda) // Declare your likelihood
}
```

.stan file

```
data {
        int<lower=0> N; // Declare the dimension of your
                                                  //observations
        vector[N] y; // Declare your observations
}

parameters {
        real<lower=0> lambda; // Declare your parameter you
                                                          //want to sample from
}

model {
        lambda ~ normal(0, 10) // Declare your prior
        y ~ exponential(lambda) // Declare your likelihood
}
```

Rules/Common Problems

- Stan files must always end with a blank line
- Stan files need to have a “data”, “parameters”, and “model” chunk
    - Can also have “transformed parameters”, “functions”, and “generated quantities”, “transformed data”, and maybe a few others
- Always declare the dimensions and support for variables. This isn’t strictly required for the “data” block, but is good practice
    - Ex: real<lower=L, upper=U> y[N]

rstan package

- Rstan does a lot of things that we won’t cover
    - Solving Algebraic Equations
    - Ordinary Differential Equations
    - One Dimensional Integrals

brms package

- “Fit Bayesian generalized (non-)linear multivariate multilevel models using ‘Stan’ for full Bayesian inference.”
- Uses familiar R notation for models (lm, glm, lmer, etc)
- Nicer output formatting
- Note: data argument must **always** be supplied, unlike functions like lm

rstanarm package

- “Estimates previously compiled regression models using the 'rstan' package.”
- Uses familiar R notation for models, but not as familiar
- Plays a little nicer with rstan functions

Where to go for help

- Stan User's Guide: https://mc-stan.org/docs/2_19/stan-users-guide/index.html
- Initialization: https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started
- Reference Manual: https://mc-stan.org/docs/2_19/reference-manual/index.html#overview
- rstanarm vignettes: http://mc-stan.org/rstanarm/articles/index.html