Title

Bayesian modeling using Stan

Abstract

The goal of this course is to assist researchers in learning to use Stan to perform Bayesian inference with applications to multilevel modeling and item response theory. The course will cover (1) an overview of Bayesian inference, (2) fitting multilevel models using the **rstanarm** package for R, (3) fitting models using the Stan language via the **rstan** package, and (4) applications to item response theory.

Prerequisites

Though multiple interfaces for Stan exist, the course will focus on using Stan through R. As such, some familiarity with R is required, such as how to estimate generalized linear models with the `glm` function and ideally multilevel models with the **lme4** package. Knowledge of Bayesian inference is helpful but not necessary.

A day before the workshop, please install the **rstan** package by following these instructions . Then, you can install the **rstanarm** package in the conventional R fashion via

```r
install.packages("rstanarm")
```

Questions about the installation process can be directed to the Stan-users Google group .

Presenters

Ben Goodrich (benjamin.goodrich@columbia.edu)
Columbia University

Daniel Furr (danielcfurr@berekley.edu)
University of California at Berkeley

Length

1 day

Summary

In the past 20 or 30 years, Bayesian inference has become increasingly prevalent. However, frequentist methods are still much more common in applied research since Bayesian methods are perceived as requiring a strong computational background.

The Stan project implements a probabilistic programming language, a library of mathematical and statistical functions, and a variety of algorithms to estimate statistical models in order to make Bayesian inferences from data. The main sections of this course will

1. Provide an introduction to modern Bayesian inference using Hamiltonian Markov Chain Monte Carlo (MCMC) as implemented in Stan.
2. Teach the process of Bayesian inference using the **rstanarm** R package, which comes with all the necessary functions to support a variety of applied regression models, including multilevel models.
3. Demonstrate the power of the Stan language, which allows users to define their own models.
4. Provide syntax for standard item response theory models and illustrate how these common models can easily be extended in the Stan language.

Stan uses Hamiltonian MCMC, which tends to be more efficient than the Gibbs sampler implemented in WinBUGS, OpenBUGS, and JAGS. Hamiltonian MCMC also offers additional diagnostics to verify whether the sampling has gone well, and there are more opportunities for reparameterizations that affect the efficiency of the sampling. Importantly, MCMC via Stan can be extremely efficient and can be successful for many models that are infeasible with BUGS. At the same time, the Stan language is at least as general as the BUGS language, so Stan can be used to estimate both familiar and cutting-edge models.

The **rstan** package has quickly become the most advanced and popular interface to Stan. Nevertheless, for many users, writing a Bayesian model in any language may be too difficult or too time-consuming. The **rstanarm** package goes a step further toward democratizing Bayesian inference by obviating the need for the user to personally write a text file in the Stan language. Instead, the **rstanarm** package comes with pre-compiled executables that implement the same likelihoods as many common R functions, such as `lm()`, `glm()`, and `glmer()`. Moreover, they can be called by R functions (which are prefixed by `stan_`) that have the same syntax as these popular model-fitting functions, and they provide additional optional arguments to specify the prior distributions over the unknown parameters.

Course outline

Modern Bayesian inference with Stan's implementation of Markov chain Monte Carlo

- Brief introduction to Bayesian inference
- Hamiltonian Markov Chain Monte Carlo as the engine for Bayesian inference
  - Troubleshooting common sampling problems
  - Effective Sample Size
- Bayesian methods of model comparison

Using the rstanarm package for Bayesian inference

- Stan-based counterparts to core model-fitting functions in R
  - `stan_lm()`
  - `stan_glm()`
  - `stan_polr()`
- Stan-based counterparts to **lme4**-style fitting functions in R to estimate multilevel models
  - `stan_glmer()`

Defining your own model with the Stan language using the rstan package

- Elements of the Stan language
- Blocks of a Stan program

  - functions
  - data
  - transformed data
  - parameters
  - transformed parameters
  - model
  - generated quantities

- Examples of defining simple and complicated models in the Stan language
- Drawing from an arbitrary posterior distribution using the `stan()` function in the **rstan** package

Applications to item response theory

- Standard models, such as the two-parameter IRT model
- Extending standard models, such as by including hierarchical priors

Presenter bios

Ben Goodrich is a core developer of the Stan project and a frequent contributor to both the Stan Users Google Group (660+ threads) and the Stan Developer Google Group (705+ threads). He is the maintainer of two Stan-related R packages that will be heavily used in this tutorial, a coauthor of a forthcoming article on Stan in the *Journal of Statistical Software*, and a Lecturer at Columbia University where he teaches graduate classes in quantitative methodology, including a masters-level course based on Stan entitled "Bayesian Statistics for the Social Sciences" (3 times). He is supported in part by a grant from the Sloan Foundation to build the Stan community and to ensure that the Stan project thrives over the long-term.

Daniel Furr is a graduate student in the Quantitative Methods and Evaluation program at the University of California at Berkeley. He is writing tutorials and documentation regarding Stan for models used in education research, particularly item response models, as a part an IES grant: "Solving difficult Bayesian computation problems in education research using Stan" by Gelman, et al. He is a coauthor of "Fitting Bayesian item response models in Stata and Stan," available on arXiv.