Bayesian Data Analysis with BRMS

Mitzi Morris
Stan Development Team
Columbia University, New York NY

2/28/23

BRMS: Bayesian Regression and Multilevelmodeling in Stan

The BRMS package fits Bayesian models using an extended R formula syntax.

```
fit <- brm(Reaction ~ 1 + Days + (1 + Days|Subject), data = sleepstudy)
```

https://paul-buerkner.github.io/brms/

Why should I use BRMS?

- Simplifies model development:
  - Use extended R formula syntax to specify the likelihood
  - User `set_prior` function to specify priors for all parameters
- Supports Bayesian workflow
  - BRMS package provides prior and posterior predictive checks
  - Works with downstream analysis packages bayesplot, projpred, and loo
- BRMS-generated Stan programs are efficient and robust

Bayesian Workflow

Model development

- Fit data to model (simulated or real)
- Evaluate the fit:
  - How good is the fit?
  - How sensitive are the results to the modeling assumptions?
  - Do the predictions make sense?

Model Comparison

- Some models are too simple
  - Learn what we lose when features are omitted
- Some models are too complex
  - Learn the limits of what we can fit given the data

Modeling Terminology and Notation

- y - data
- θ - parameters
- p(y, θ) - **joint probability distribution** of the data and parameters
- p(θ) - **prior probability distribution** - the probability of the parameters before any data are observed
- p$(\theta \mid y)$ - **posterior probability distribution** - the probability of the parameters conditional on the data (i.e., after seeing the data).
- p$(y \mid \theta)$
  - if y is fixed, this is the **likelihood function**
  - if θ is fixed, this is the **sampling distribution**

Multilevel Regression

McElreath: "Multilevel regression deserves to be the default form of regression."

Statistical Rethinking, 2nd ed, section 1.3.2

Multilevel regression models can handle structured data.

- Almost all data has some structure
  - Observations are repeated or ordered or come from different (nested) groups, e.g.
  - Hierarchical: students in classrooms in schools in districts in states in regions
  - Auto-regressive: time series, spatial data, spatio-temporal data
- With a multilevel models, we can say more about the data
  - Estimate variation on all levels of the model
  - Predict values of new groups not originally present in the data

Regression Models in R

- Pre-existing packages `lm`, `glm`, `lme4`
  - `lm`, `glm` - single-level linear models
  - `lme4` - hierarchal linear model
- Stan (2010) - build a better `lme4`
  - Stan probabilistic programming language based on BUGS
  - NUTS-HMC algorithm more efficient MCMC sampler
- BRMS (2016) - simplify model specification.
  - Use `lme4`-style formulas and R functions to wrap Stan
  - User specifies formula, priors, BRMS generates Stan program
- RStanARM (2015) - precompiled Stan models

Linear Regression

Linear regression relates a scalar outcome (the dependent variable “y”) to one or more predictors (the independent variable “x”). For a single predictor x

- $y_i = \alpha + \beta x_i + \epsilon_i$
  - α is the *intercept*, the offset from zero on the x-axis
  - β is the *slope*, the multiplier applied to x.
  - ϵ_i is the error term

When ϵ_i are independent errors drawn from a normal distribution with mean 0, standard deviation σ, the **linear model** is

- $y_i \sim \mathrm{N}(\alpha + \beta x_i, \sigma)$
  - $\alpha + \beta x_i$ is the *linear predictor*
  - σ is the variance

Generalized Linear Regression

We extend the simple linear model $y_i \sim \mathrm{N}(\alpha + \beta\, x_i, \sigma)$ to a multilevel general linear regression as follows

- Instead of a normal distribution N, we can use any distributional **family** D, (e.g., a Beta distribution), correspondingly, we generalize the variance parameter σ to any family-specific parameter θ
- We generalize $\alpha + \beta\, x_i$ to η, any linear predictor
- The linear predictor can be transformed by any *inverse link function* f
- We use group-level subscripts to allow for group-level parameters.

General Multilevel Model: $y_i \sim \mathrm{D}(\, f(\eta_i), \theta)$

Don't let these definitions obscure the fact we are defining a function comprised of **intercept** and **slope** terms.

Regression Formula Syntax

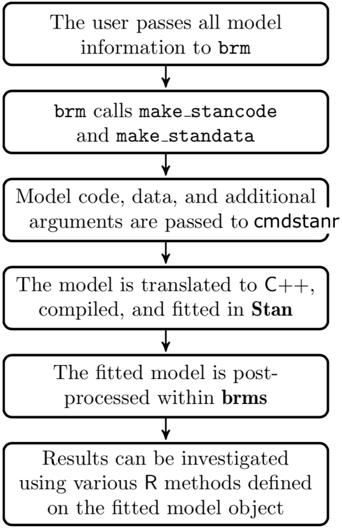
A regression formula has the general form $\mathrm{LHS} \sim \mathrm{RHS}$

```
Reaction ~ 1 + Days + (1 + Days|Subject)
```

- The left-hand side is the outcome, in the simplest case, a single observed value.
- The right-hand side is the linear predictor, consisting of
  - “Population-level” terms (a.k.a. fixed effects)
  - “Group-level” terms (a.k.a. random effects) which vary by grouping factor.

    Group-level terms are of the form `(coefs | group)`, where `group` is a grouping factor and `coefs` refer to the predictors whose effects vary with the levels of the grouping factor.
  - The number `1` corresponds to an intercept term

BRMS Processing



The user passes all model information to brm

brm calls make_stancode and make_standata

Model code, data, and additional arguments are passed to cmdstanr

The model is translated to C++, compiled, and fitted in **Stan**

The fitted model is post-processed within **brms**

Results can be investigated using various R methods defined on the fitted model object

Notebook

link to online notebook and data here

References

- https://paul-buerkner.github.io/brms/articles/index.html
- https://xcelab.net/rm/statistical-rethinking/
- https://journal.r-project.org/archive/2018/RJ-2018-017/RJ-2018-017.pdf
- https://www.barelysignificant.com/slides/RGUG2019#1
- https://ourcodingclub.github.io/tutorials/brms/
- https://onlinelibrary.wiley.com/doi/pdf/10.1111/eth.13225
- https://mc-stan.org/users/documentation/case-studies/tutorial_rstanarm.html