brms: An R Package for Bayesian Multilevel Models using Stan

Paul Bürkner

Institut für Psychologie
Westfälische Wilhelms-Universität Münster

26.02.2016
Agenda

1. Short introduction to Stan

2. The brms package
   - Model Specification
   - Model Fitting
   - Post-Processing

3. Discussion
Stan Overview

- Probabilistic programming language...
- ... to fit open-ended Bayesian Models
- Powerful sampling algorithms: HMC and NUTS
- Automatic differentiation library
- Runs on all major platforms (Windows, OS X, Linux)
- Can be called from R, Python, Julia, Stata, and Matlab
Short introduction to Stan

Stan code for simple linear regression

data {
  int<lower=1> N;    // number of observations
  vector[N] Y;      // response variable
  int<lower=1> K;    // number of fixed effects
  matrix[N, K] X;   // FE design matrix
}

parameters {
  vector[K] b;    // fixed effects
  real<lower=0> sigma;  // residual SD
}

transformed parameters {
  vector[N] eta;  // linear predictor
  eta <- X * b;
}

model {
  // prior specifications
  sigma ~ student_t(3, 0, 100);
  // likelihood contribution
  Y ~ normal(eta, sigma);
}
The idea of brms

Make Stan usable for a wider range of audience

- use R formula syntax
- internally write Stan code that is readable yet fast
- provide an easy interface for defining priors
- facilitate post-processing
Some features of brms

Basic model types:
- (Robust multivariate) linear models
- Count data models
- Categorical and ordinal models
- Survival models
- Zero-inflated and hurdle models
- Non-linear models

Other modeling options:
- Group specific terms (random effects) using **lme4** syntax
- Residual autocorrelation
- Censored / truncated data
Example: Censored survival regression

```r
brm(time | cens(censored) ~ age * sex + (1 + age|patient),
     data = kidney, family = exponential())
```

Let’s add some priors:

```r
brm(time | cens(censored) ~ age * sex + (1 + age|patient),
     data = kidney, family = exponential(),
     prior = c(set_prior("normal(0,5)", class = "b"),
               set_prior("cauchy(0,2)", class = "sd"),
               set_prior("lkj(2)", class = "cor")))
```
Flow chart of the model fitting procedure

1. The user passes all model information to `brm`.
2. `brm` calls `make_stancode` and `make_standata`.
3. Model code, data, and additional arguments are passed to `rstan`.
4. The model is translated to C++, compiled, and fitted in `Stan`.
5. The fitted model is post-processed within `brms`.
6. Results can be investigated using various R methods defined on the fitted model object.
### Example: Summary

- **Family:** exponential (log)
- **Formula:** `time | cens(censored) ~ age * sex + (1 + age | patient)`
- **Data:** kidney (Number of observations: 76)
- **Samples:** 4 chains, each with iter = 2000; warmup = 1000; thin = 1; total post-warmup samples = 4000
- **WAIC:** 668.22

#### Random Effects:

<table>
<thead>
<tr>
<th>~patient (Number of levels: 38)</th>
<th>Estimate</th>
<th>Est.Error</th>
<th>l-95% CI</th>
<th>u-95% CI</th>
<th>Eff.Sample</th>
<th>Rhat</th>
</tr>
</thead>
<tbody>
<tr>
<td>sd(Intercept)</td>
<td>0.45</td>
<td>0.30</td>
<td>0.02</td>
<td>1.07</td>
<td>1172</td>
<td>1.00</td>
</tr>
<tr>
<td>sd(age)</td>
<td>0.01</td>
<td>0.01</td>
<td>0.00</td>
<td>0.03</td>
<td>450</td>
<td>1.01</td>
</tr>
<tr>
<td>cor(Intercept,age)</td>
<td>-0.12</td>
<td>0.45</td>
<td>-0.87</td>
<td>0.77</td>
<td>1133</td>
<td>1.00</td>
</tr>
</tbody>
</table>

#### Fixed Effects:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Est.Error</th>
<th>l-95% CI</th>
<th>u-95% CI</th>
<th>Eff.Sample</th>
<th>Rhat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>2.91</td>
<td>0.93</td>
<td>1.22</td>
<td>4.85</td>
<td>1636</td>
<td>1</td>
</tr>
<tr>
<td>age</td>
<td>0.02</td>
<td>0.02</td>
<td>-0.02</td>
<td>0.06</td>
<td>1527</td>
<td>1</td>
</tr>
<tr>
<td>sexfemale</td>
<td>2.76</td>
<td>1.07</td>
<td>0.59</td>
<td>4.77</td>
<td>1578</td>
<td>1</td>
</tr>
<tr>
<td>age:sexfemale</td>
<td>-0.03</td>
<td>0.02</td>
<td>-0.08</td>
<td>0.02</td>
<td>1478</td>
<td>1</td>
</tr>
</tbody>
</table>
Example: Trace and density plots

**Trace**
- `b_age`
- `sd_patient_Intercept`
- `cor_patient_Intercept_age`

**Density**
- `b_age`
- `sd_patient_Intercept`
- `cor_patient_Intercept_age`
Example: Marginal plots
Example: Hypothesis Testing

```r
> hypothesis(fit, "Intercept - age > 0", class = "sd", group = "patient")

Hypothesis Tests for class sd_patient:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Est.Error</th>
<th>l-95% CI</th>
<th>u-95% CI</th>
<th>Evid.Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept-age &gt; 0</td>
<td>0.44</td>
<td>0.3</td>
<td>0.03</td>
<td>Inf</td>
</tr>
</tbody>
</table>
```

Hypothesis for class sd_patient

![Graph showing posterior distribution for Intercept-age > 0]
Plans for the future

- Full support for multivariate models
- Spatial correlation structures
- Generalized additive multilevel models
- Mixture distributions

Your ideas are always welcome!
Learn more about brms and Stan

- Help within R: `vignette("brms")` or `help("brms")`
- List of all applicable methods: `methods(class = "brmsfit")`
- Website of brms: https://github.com/paul-buerkner/brms
- Website of Stan: http://mc-stan.org/
- Contact me: paul.buerkner@gmail.com
Thank you for your attention!

Questions?