Bayesian Item Response Modeling in R with brms and Stan

Paul Bürkner
Bürkner P. C. (accepted). Bayesian Item Response Modeling in R with brms and Stan. *Journal of Statistical Software*. 


We need:

- A set of parameters $\xi_i$ for item $i$
- A set of parameters $\theta_p$ for person $p$
- A model for the responses $y_{ip}$

$$y_{ip} \sim \text{model}(\xi_i, \theta_p)$$

- Some restrictions on the parameters $(\xi, \theta)$
Bayesian IRT Models

We need:

- A set of parameters $\xi_i$ for item $i$
- A set of parameters $\theta_p$ for person $p$
- A model for the responses $y_{ip}$

$$y_{ip} \sim \text{model}(\xi_i, \theta_p)$$

- Some priors on the parameters $(\xi, \theta)$:

$$\xi_i \sim \text{prior}(.)$$

$$\theta_p \sim \text{prior}(.)$$
Consider the data to be in long format and a (pointwise) likelihood with *distributional parameters* $\psi_1$ to $\psi_K$:

$$y \sim \text{likelihood}(\psi_1, \psi_2, \ldots, \psi_K)$$

Connect the distributional parameters to the item and person parameters via response functions $f_k$:

$$\psi_k = f_k(\xi_i, \theta_p)$$
Binary response $y$ and a single distributional parameter $\psi$:

$$y \sim \text{Bernoulli}(\psi) = \psi^y (1 - \psi)^{1-y},$$

Rasch Model:

$$\psi = f(\xi_i + \theta_p) = \frac{\exp(\theta_p + \xi_i)}{1 + \exp(\theta_p + \xi_i)}$$

2PL Model:

$$\psi = f(\alpha_i(\theta_p + \xi_i))$$

3PL Model:

$$\psi = \gamma_i + (1 - \gamma_i) f(\alpha_i(\theta_p + \xi_i))$$
Non-hierarchical prior for item parameters:

$$\xi_i \sim \text{Normal}(0, 3)$$

Hierarchical prior for single parameter per item:

$$\xi_i \sim \text{Normal}(0, \sigma_\xi)$$

$$\sigma_\xi \sim \text{Normal}_+(0, 1)$$
Hierarchical prior for multiple parameters per item:

$$(\xi_{1i}, \ldots, \xi_{Ki}) \sim \text{MultiNormal}(0, \Sigma_{\xi})$$

Decompose the covariance matrix $\Sigma_{\xi}$ as:

$$\Sigma_{\xi} = D(\sigma_{\xi 1}, \ldots, \sigma_{\xi K}) \Omega_{\xi} D(\sigma_{\xi 1}, \ldots, \sigma_{\xi K})$$

$$\sigma_{\xi k} \sim \text{Normal}_+(0, 1)$$

$$\Omega_{\xi} \sim \text{LKJ}(1)$$
Stan and brms
Model specification in brms: family

General structure:

```r
family = brmsfamily(
    family = "<family>", link = "<link>",
    <more_link_arguments>
)
```

Binary Model:

```r
family = brmsfamily(family = "bernoulli", link = "logit")
```

Gaussian Model:

```r
family = brmsfamily(
    family = "gaussian", link = "identity",
    link_sigma = "log"
)
```
Model Specification in \textit{brms}: \textit{formula}

Item parameters have independent priors, person parameters have hierarchical priors:

\texttt{formula = y \sim 0 + item + (1 | person)}

Both item and person parameters have hierarchical priors:

\texttt{formula = y \sim 1 + (1 | item) + (1 | person)}

Add a covariate:

\texttt{formula = y \sim 1 + x + (1 | item) + (1 | person)}
Model Specification in brms: formula

Linear formulas for multiple distributional parameters:

formula = bf(
    y ~ 1 + (1 | item) + (1 | person),
    par2 ~ 1 + (1 | item) + (1 | person),
    par3 ~ 1 + (1 | item) + (1 | person),
)

Non-linear formula for a single distributional parameter:

formula = bf(
    y ~ fun(x, nlpar1, nlpar2),
    nlpar1 ~ 1 + (1 | item) + (1 | person),
    nlpar2 ~ 1 + (1 | item),
    nl = TRUE
)
Model Specification in brms: formula

Linear formulas for multiple distributional parameters:

```r
formula = bf(
    y ~ 1 + (1 |i| item) + (1 |p| person),
    par2 ~ 1 + (1 |i| item) + (1 |p| person),
    par3 ~ 1 + (1 |i| item) + (1 |p| person),
)
```

Non-linear formula for a single distributional parameter:

```r
formula = bf(
    y ~ fun(x, nlpar1, nlpar2),
    nlpar1 ~ 1 + (1 |i| item) + (1 |p| person),
    nlpar2 ~ 1 + (1 |i| item),
    nl = TRUE
)
```
### Case Study: The VerbAgg Data Set

```r
data("VerbAgg", package = "lme4")
```

<table>
<thead>
<tr>
<th>Anger</th>
<th>Gender</th>
<th>item</th>
<th>resp</th>
<th>id</th>
<th>btype</th>
<th>situ</th>
<th>mode</th>
<th>r2</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>M</td>
<td>S1WantCurse</td>
<td>no</td>
<td>1</td>
<td>curse</td>
<td>other</td>
<td>want</td>
<td>N</td>
</tr>
<tr>
<td>11</td>
<td>M</td>
<td>S1WantCurse</td>
<td>no</td>
<td>2</td>
<td>curse</td>
<td>other</td>
<td>want</td>
<td>N</td>
</tr>
<tr>
<td>17</td>
<td>F</td>
<td>S1WantCurse</td>
<td>perhaps</td>
<td>3</td>
<td>curse</td>
<td>other</td>
<td>want</td>
<td>Y</td>
</tr>
<tr>
<td>21</td>
<td>F</td>
<td>S1WantCurse</td>
<td>perhaps</td>
<td>4</td>
<td>curse</td>
<td>other</td>
<td>want</td>
<td>Y</td>
</tr>
<tr>
<td>17</td>
<td>F</td>
<td>S1WantCurse</td>
<td>perhaps</td>
<td>5</td>
<td>curse</td>
<td>other</td>
<td>want</td>
<td>Y</td>
</tr>
<tr>
<td>21</td>
<td>F</td>
<td>S1WantCurse</td>
<td>yes</td>
<td>6</td>
<td>curse</td>
<td>other</td>
<td>want</td>
<td>Y</td>
</tr>
<tr>
<td>39</td>
<td>F</td>
<td>S1WantCurse</td>
<td>yes</td>
<td>7</td>
<td>curse</td>
<td>other</td>
<td>want</td>
<td>Y</td>
</tr>
<tr>
<td>21</td>
<td>F</td>
<td>S1WantCurse</td>
<td>no</td>
<td>8</td>
<td>curse</td>
<td>other</td>
<td>want</td>
<td>N</td>
</tr>
<tr>
<td>24</td>
<td>F</td>
<td>S1WantCurse</td>
<td>no</td>
<td>9</td>
<td>curse</td>
<td>other</td>
<td>want</td>
<td>N</td>
</tr>
<tr>
<td>16</td>
<td>F</td>
<td>S1WantCurse</td>
<td>yes</td>
<td>10</td>
<td>curse</td>
<td>other</td>
<td>want</td>
<td>Y</td>
</tr>
</tbody>
</table>
Fitting a Rasch Model

formula_va_1pl <- bf(r2 ~ 1 + (1 | item) + (1 | id))

prior_va_1pl <-
  prior("normal(0, 3)", class = "sd", group = "id") +
  prior("normal(0, 3)", class = "sd", group = "item")

fit_va_1pl <- brm(
  formula = formula_va_1pl,
  data = VerbAgg,
  family = brmsfamily("bernoulli", "logit"),
  prior = prior_va_1pl
)
Rasch Model: Investigate the Posterior

```r
plot(fit_va_1pl)
```

![Distribution plots for different parameters](image)

- **b_Intercept**
  - Chain 1: Distribution
  - Chain 2: Trace plot
  - Chain 3: Trace plot
  - Chain 4: Trace plot

- **sd_id__Intercept**
  - Chain 1: Distribution
  - Chain 2: Trace plot
  - Chain 3: Trace plot
  - Chain 4: Trace plot

- **sd_item__Intercept**
  - Chain 1: Distribution
  - Chain 2: Trace plot
  - Chain 3: Trace plot
  - Chain 4: Trace plot

```r
Chain 1 2 3 4
```

```
```
Fitting a 2PL Model

```r
formula_va_2pl <- bf(
  r2 ~ exp(logalpha) * eta,
  eta ~ 1 + (1 |i| item) + (1 | id),
  logalpha ~ 1 + (1 |i| item),
  nl = TRUE
)

prior_va_2pl <-
  prior("normal(0, 5)", class = "b", nlpar = "eta") +
  prior("normal(0, 1)", class = "b", nlpar = "logalpha") +
  prior("constant(1)", class = "sd", group = "id", nlpar = "eta") +
  prior("normal(0, 3)", class = "sd", group = "item", nlpar = "eta") +
  prior("normal(0, 1)", class = "sd", group = "item", nlpar = "logalpha")

fit_va_2pl <- brm(
  formula = formula_va_2pl,
  data = VerbAgg,
  family = brmsfamily("bernoulli", "logit"),
  prior = prior_va_2pl,
)
```
The loo method implements approximate leave-one-out cross-validation via Pareto-Smoothed importance sampling:

```r
loo_compare(loo(fit_va_1pl), loo(fit_va_2pl))
```

## elpd_diff se_diff
## fit_va_2pl 0.0 0.0
## fit_va_1pl -3.0 2.4
Adding Person and Item Covariates

```r
formula_va_1pl_cov2 <- bf(
  r2 ~ btype + situ + mode * Anger + Gender +
  (0 + Gender | item) + (0 + mode | id)
)

fit_va_1pl_cov2 <- brm(
  formula = formula_va_1pl_cov2,
  data = VerbAgg,
  family = brmsfamily("bernoulli", "logit"),
  prior = prior_va_1pl
)
```
Visualizing Covariate Effects

```r
conditional_effects(fit_va_1pl_cov2, "mode:Gender")
```
Case Study: The Rotation Data Set

data("rotation", package = "diffIRT")

<table>
<thead>
<tr>
<th>person</th>
<th>item</th>
<th>time</th>
<th>resp</th>
<th>rotate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>4.444</td>
<td>1</td>
<td>150</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>5.447</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2.328</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3.408</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>5.134</td>
<td>1</td>
<td>150</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>2.653</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>2.607</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>3.126</td>
<td>1</td>
<td>150</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>2.869</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>3.271</td>
<td>1</td>
<td>150</td>
</tr>
</tbody>
</table>
Wiener Drift Diffusion Models

reaction time (RT)

nondecision time

decision time

nondecision time

correct RT distribution

decision boundary (e.g. face)

time

stimulus

(a)

v

z

(correct) response

correct decision
	error decision

decision boundary (e.g. house)

error RT distribution
bform_drift1 <- bf(
  time | dec(resp) ~ rotate + (1 | p| person) + (1 | i| item),
  bs ~ rotate + (1 | p| person) + (1 | i| item),
  ndt ~ rotate + (1 | p| person) + (1 | i| item),
  bias = 0.5
)

fit_drift1 <- brm(
  formula = bform_drift1,
  data = rotation,
  family = brmsfamily(
    "wiener", "log", link_bs = "log",
    link_ndt = "log"
  )
)
Learn More about brms and Stan

- Help within R: `help("brms")`
- Overview of vignettes: `vignette(package = "brms")`
- List of all methods: `methods(class = "brmsfit")`
- Website of brms: [https://github.com/paul-buerkner/brms](https://github.com/paul-buerkner/brms)
- Website of Stan: [http://mc-stan.org/](http://mc-stan.org/)
- Contact me: paul.buerkner@gmail.com
- Twitter: @paulbuerkner