*Journal of Statistical Software*

*MMMMMM YYYY, Volume VV, Issue II.* doi: 10.18637/jss.v000.i00

# Bayesian Item Response Modelling in R Using brms and Stan

**Paul-Christian Bürkner**
University of Münster

**Abstract**

Item Response Theory (IRT) is widely applied in the social sciences to model persons' responses on a set of items measuring one or more dimensions. While several R packages have been developed that implement IRT models, they are each limited to a few subclasses of models. Unfortunately, model specification syntax and post-processing methods are rather inconsistent across packages making it hard for users to switch between models implemented in the different packages. Further, most implementations are frequentist while the availibility of Bayesian methods remains comparably limited. We demonstrate how to use the R package brms and Stan to specify and fit a wide range of Bayesian IRT models using well known multilevel formula syntax. Further, item and person parameters can be related in both a linear or non-linear manner. Discrete, ordinal, and continuous responses are supported. Common IRT model classes of the presented framework include 1PL and 2PL logistic models optionally also containing guessing parameters, graded response and partial credit ordinal models, as well as drift diffusion models of response times coupled with binary decisions. Posterior distributions of item and person parameters can be conveniently extracted and summarized. Model fit can be evaluated and compared using Bayes factors and approximate cross-validation procedures.

*Keywords*: Item Response Theory, Bayesian Statistics, R, Stan, **brms**.

## 1. Introduction

## 2. Model description and formula syntax

## 3. Parameter estimation and post-processing

The **brms** package does not fit models itself but uses **Stan** (Carpenter, Gelman, Hoffman, Lee, Goodrich, Betancourt, Brubaker, Guo, Li, and Ridell 2017) on the back-end. Accordingly, all samplers implemented in **Stan** can be used to fit **brms** models. The flagship algorithm of Stan is an adaptive Hamiltonian Monte-Carlo (HMC) sampler (Betancourt, Byrne, Livingstone, and Girolami 2014; Betancourt 2017; Stan Development Team 2019), which progressed from the No-U-Turn Sampler (NUTS) by Hoffman and Gelman (2014). HMC-like algorithms produce posterior samples that are much less autocorrelated than those of other samplers such as the random-walk Metropolis algorithm (Hoffman and Gelman 2014; Creutz 1988). What is more, consequtive samples may even be anti-correlated leading to higher efficiency than completely independent samples (Gelman, Carlin, Stern, and Rubin 2014). The main drawback of this increased efficiency is the need to calculate the gradient of the log-posterior, which can be automated using algorithmic differentiation (Griewank and Walther 2008) but is still a time-consuming process for more complex models. Thus, using HMC leads to higher quality samples but takes more time per sample than other algorithms typically applied. Another drawback of HMC is the need to pre-specify at least two parameters, which are both critical for the performance of HMC. The adaptive HMC Sampler of Stan allows setting these parameters automatically thus eliminating the need for any hand-tuning, while still being at least as efficient as a well tuned HMC (Hoffman and Gelman 2014). For more details on the sampling algorithms applied in **Stan**, see the **Stan** user's manual (Stan Development Team 2019) as well as Hoffman and Gelman (2014).

After the estimation of the parameters' joint posterior distribution, **brms** offers a wide range of post-processing options of which several are helpful in an IRT context. Below, we introduce the most important post-processing options. We will show their usage in hands-on examples in the upcoming sections. For a quick numerical and graphical summary, respectively, of the central model parameters, we recommend the `summary` and `plot` methods. The posterior distribution of person parameters (and, if also modeled as varying effects, item parameters) can be extracted with the `coef` method. The `hypothesis` method can be used to easily compute and evaluate parameter contrasts, for instance, when the goal is to compare the difficulty of two items or the ability of two persons. A visualization of the effects of item or person covariates is readily available via the `marginal_effects` method.

With the help of the `posterior_predict` method, **brms** allows drawing samples from the posterior predictive distribution. This not only allows to make predictions for existing and new data, but also enables the comparison between the actual response $y$ and the response $\hat{y}$ predicted by the model. Such comparisons can be visualized in the form of posterior-predictive checks by means of the `pp_check` method (Gabry, Simpson, Vehtari, Betancourt, and Gelman 2019). Further, via the `log_lik` method, the pointwise log-likelihood can be obtained, which can be used, among others, for various cross-validation methods. Possibly the most widely known cross-validation approach is leave-one-out cross-validation (LOO-CV; Vehtari, Gelman, and Gabry 2017b), for which an approximate version is available via the `loo` method of the **loo** package (Vehtari *et al.* 2017b; Vehtari, Gelman, and Gabry 2017a). If LOO-CV is not an option or if the approximation fails, exact k-fold cross-valdiation is available via the `kfold` method. The cross-validation results can be further post-processed for the purpose of comparison, selection, or averaging of models. In these contexts, the `loo_compare`, `model_weights`, and `pp_average` methods are particularly helpful.

In addition to cross-validation based fit measures, the marginal likelihood (i.e., the denomintor in Bayes' theorem) and marginal likelihood ratios, commonly known as Bayes factors, can be used for model comparison, selection, or averaging as well (Kass and Raftery 1995). In general, obtaining the marginal likelihood of a model is a computationally demanding task (Kass and Raftery 1995). In **brms** this is realized via bridgesampling (Meng and Wong 1996; Meng and Schilling 2002) as implemented in the **bridgesampling** package (Gronau and Singmann 2018). The corresponding methods are called `bridge_sampler` to obtain (log) marginal likelihood estimates, `bayes_factor` to obtain Bayes factors and `post_prob` to obtain posterior model probabilities based on prior model probabilities and marginal likelihood estimates.

# 4. Binary Models

# 5. Ordinal Models

# 6. Response Times Models

# Conclusion

Betancourt M (2017). "A Conceptual Introduction to Hamiltonian Monte Carlo." *arXiv preprint*. URL `https://arxiv.org/pdf/1701.02434.pdf`.

Betancourt M, Byrne S, Livingstone S, Girolami M (2014). "The Geometric Foundations of Hamiltonian Monte Carlo." *arXiv preprint arXiv:1410.5110*.

Carpenter B, Gelman A, Hoffman M, Lee D, Goodrich B, Betancourt M, Brubaker MA, Guo J, Li P, Ridell A (2017). "Stan: A Probabilistic Programming Language." *Journal of Statistical Software*.

Creutz M (1988). "Global Monte Carlo Algorithms for Many-Fermion Systems." *Physical Review D*, **38**(4), 1228.

Gabry J, Simpson D, Vehtari A, Betancourt M, Gelman A (2019). "Visualization in Bayesian workflow." *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, **182**(2), 389–402.

Gelman A, Carlin JB, Stern HS, Rubin DB (2014). *Bayesian Data Analysis*, volume 2. Taylor & Francis.

Griewank A, Walther A (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Siam.

Gronau QF, Singmann H (2018). *bridgesampling: Bridge Sampling for Marginal Likelihoods and Bayes Factors*. R package version 0.6-0, URL `https://CRAN.R-project.org/package=bridgesampling`.

Hoffman MD, Gelman A (2014). "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo." *The Journal of Machine Learning Research*, **15**(1), 1593–1623.

Kass RE, Raftery AE (1995). "Bayes factors." *Journal of the american statistical association*, **90**(430), 773–795.

Meng XL, Schilling S (2002). "Warp bridge sampling." *Journal of Computational and Graphical Statistics*, **11**(3), 552–586.

Meng XL, Wong WH (1996). "Simulating ratios of normalizing constants via a simple identity: a theoretical exploration." *Statistica Sinica*, pp. 831–860.

Stan Development Team (2019). *Stan Modeling Language: User's Guide and Reference Manual*. URL http://mc-stan.org/manual.html.

Vehtari A, Gelman A, Gabry J (2017a). "Pareto smoothed importance sampling." *arXiv preprint*. URL https://arxiv.org/abs/1507.02646.

Vehtari A, Gelman A, Gabry J (2017b). "Practical Bayesian Model Evaluation Using Leave-One-Out Cross-Validation and WAIC." *Statistics and Computing*, **27**(5), 1413–1432. URL http://link.springer.com/article/10.1007/s11222-016-9696-4.

**Affiliation:**

Paul-Christian Bürkner
University of Münster
Fliednerstr. 21, 48149 Münster, Germany
E-mail: paul.buerkner@gmail.com
URL: https://paul-buerkner.github.io/

*Journal of Statistical Software* http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd