Journal of
Intelligence

Article

Analysing Standard Progressive Matrices (SPM-LS) with Bayesian Item Response Models

Paul-Christian Bürkner

Department of Computer Science, Aalto University, Konemiehentie 2, 02150 Espoo, Finland;
mailto:paul.buerkner@gmail.com

Received: 21 November 2019; Accepted: 29 January 2020; Published: 4 February 2020

Abstract: Raven's Standard Progressive Matrices (SPM) test and related matrix-based tests are widely applied measures of cognitive ability. Using Bayesian Item Response Theory (IRT) models, I reanalyzed data of an SPM short form proposed by Myszkowski and Storme (2018) and, at the same time, illustrate the application of these models. Results indicate that a three-parameter logistic (3PL) model is sufficient to describe participants dichotomous responses (correct vs. incorrect) while persons' ability parameters are quite robust across IRT models of varying complexity. These conclusions are in line with the original results of Myszkowski and Storme (2018). Using Bayesian as opposed to frequentist IRT models offered advantages in the estimation of more complex (i.e., 3–4PL) IRT models and provided more sensible and robust uncertainty estimates.

Keywords: Standard Progressive Matrices; Item Response Theory; Bayesian statistics; brms; Stan; R

1. Introduction

Raven's Standard Progressive Matrices (SPM) test ([1]) matrix-based tests are widely applied measures of cognitive ability (e.g., [2,3]). Due to their non-verbal content, which reduces biases due to language and cultural differences, they are considered one of the purest measures of fluid intelligence ([4]). However, a disadvantage of the original SPM is that its administration takes considerable time as 60 items have to be answered and time limits are either very loose or not imposed at all (e.g., [3]). Thus, using it as part of a bigger procedure involving the administration of the SPM as part of a battery of tests and/or experiments may be problematic. This is not only due to direct time restrictions but also because participants' motivation and concentration tends to decline over the course of the complete procedure, potentially leading to less valid measurements (e.g., [5]).

Recently, Myszkowski and Storme ([4]) proposed a short version of the original SPM test, called SPM-LS, comprising only the last block of the 12 most complex SPM items. They evaluated the statistical properties of the SPM-LS using methods of Item Response Theory (IRT). IRT is widely applied in the human sciences to model persons' responses on a set of items measuring one or more latent constructs (for a comprehensive introduction see [6–8]). Due to its flexibility compared to Classical Test Theory (CTT), IRT provides the formal statistical basis for most modern psychological measurement. The best known IRT models are likely those for binary responses, which predict the probability of a correct answer depending on item's properties and the participant's latent abilities. As responses on SPM items can be categorized as either right or wrong, I focus on these binary models in the present paper (although other models for these data are possible as well; see [4]). Myszkowski and Storme ([4]), whose data I sought to reanalyze, used frequenstist IRT models for inference. In this paper, I apply Bayesian IRT models instead and investigate potential differences to the original results. In doing so, I hope to improve our understanding of the robustness of the inference obtainable from the SPM-LS test and to illustrate the application of Bayesian IRT methods.

2. Bayesian IRT Models

In Bayesian statistics applied to IRT, we aim to estimate the posterior distribution $p(\theta, \xi | y)$ of the person and item parameters (θ and ξ, respectively, which may vary in number depending on the model) given the data y. We may be either interested in the posterior distribution directly, or in quantities that can be computed on its basis. The posterior distribution for an IRT model is defined as

$$p(\theta, \xi | y) = \frac{p(y|\theta, \xi)\, p(\theta, \xi)}{p(y)}. \tag{1}$$

In the above equation, $p(y|\theta, \xi)$ is the likelihood, $p(\theta, \xi)$ is the prior distribution, and $p(y)$ is the marginal likelihood. The likelihood $p(y|\theta, \xi)$ is the distribution of the data given the parameters and thus relates the data to the parameters. tThe prior distribution $p(\theta, \xi)$ describes the uncertainty in the person and item parameters before having seen the data. It thus allows explicitly incorporating prior knowledge into the model and/or helping to identify the model. In practice, we factorize the joint prior $p(\theta, \xi)$ into the product of $p(\theta)$ and $p(\xi)$ so that we can specify priors on person and items parameters independently. I provide more details on likelihoods and priors for Bayesian IRT models in the next section. The marginal likelihood $p(y)$ serves as a normalizing constant so that the posterior is an actual probability distribution. Except in the context of specific methods (e.g., Bayes factors), $p(y)$ is rarely of direct interest.

Obtaining the posterior distribution analytically is only possible in certain cases of carefully chosen combinations of prior and likelihood, which may considerably limit modelling flexibility but yield a computational advantage. However, with the increased power of today's computers, Markov-Chain Monte-Carlo (MCMC) sampling methods constitute a powerful and feasible alternative to obtaining posterior distributions for complex models in which the majority of modeling decisions is made based on theoretical and not computational grounds. Despite all the computing power, these sampling algorithms are computationally very intensive and thus fitting models using full Bayesian inference is usually much slower than in point estimation techniques. If using MCMC to fit a Bayesian model turns out to be infeasible, an alternative is to perform optimization over the posterior distribution to obtain Maximum A-Posteriori (MAP) estimates, a procedure similar to maximum likelihood estimation just with additional regularization through priors. MCMC and MAP estimates differ in at least two aspects. First, MCMC allows obtaining point estimates (e.g., means or medians) from the unidimensional marginal posteriors of the quantities of interest, which tend to be more stable than MAP estimates obtained from the multidimensional posterior over all parameters. Second, in contrast to MAP, MCMC provides a set of random draws from the model parameters' posterior distribution. After the model fitting, the posterior distribution of any quantity that is a function of the original parameters can be obtained by applying the function on a draw by draw basis. As such, the uncertainty in the posterior distribution naturally propagates to new quantities, a highly desirable property that is difficult to achieve using point estimates alone.

In the present paper, I apply Bayesian binary IRT models to the SPM-LS data using both MCMC and MAP estimators. Their results are compared to those obtained by frequentist maximum likelihood estimation. For a comprehensive introduction to Bayesian IRT modeling see, for example, the works of Fox ([9]), Levy and Mislevy ([10]), and Rupp, Dey, and Zumbo ([11]).

2.1. Bayesian IRT Models for Binary Data

In this section, I introduce a set of Bayesian IRT models for binary data and unidimensional person traits. Suppose that, for each person j ($j = 1, \ldots, J$) and item i ($i = 1, \ldots, I$), we have observed a binary response y_{ji}, which is coded as 1 for a correct answer and 0 otherwise. With binary IRT models, we aim to model $p_{ji} = P(y_{ji} = 1)$, that is, the probability the person j answers item i correctly. In other words, we assume a Bernoulli distribution for the responses y_{ji} with success probability p_{ji}:

$$y_{ji} \sim \text{Bernoulli}(p_{ji}) \tag{2}$$

Across all models considered here, we assume that all items measure a single latent person trait θ_j. For the present data, we can expect θ_j to represent something closely related to fluid intelligence ([4]). The most complex model I consider in this paper is the four-parameter logistic (4PL) model and all other simpler models result from this model by fixing some item parameters to certain values. In recent years, the 4PL model has received much attention in IRT research due to its flexibility in modeling complex binary response processes (e.g., [12–15]). Under this model, we express $P(y_{ji} = 1)$ via the equation

$$P(y_{ji} = 1) = \gamma_i + (1 - \gamma_i - \psi_i) \frac{1}{1 + \exp(-(\beta_i + \alpha_i \theta_j))}. \tag{3}$$

In the 4PL model, each item has four associated item parameters. The β_i parameter describes the location of the item, that is, how easy or difficult it is in general. In the above formulation of the model, higher values of β_i imply higher success probabilities and hence β_i can also be called the "easiness" parameter. The α_i parameter describes how strongly item i is related to the latent person trait θ_j. We can call α_i "factor loading", "slope", or "discrimination" parameter, but care must be taken that none of these terms is used uniquely and their exact meaning can only be inferred in the context of a specific model (e.g., see Bürkner, 2019 for a somewhat different use of the term "discrimination" in IRT models). For our purposes, we assume α_i to be positive as we expect answering the items correctly implies higher trait scores than when answering incorrectly. In addition, if we did not fix the sign of α_i, we may run into identification issues as changing the sign of α_i could be compensated by changing the sign of θ_j without a change in the likelihood.

The γ_i parameter describes the guessing probability, that is, the probability of any person answering item i correctly even if they do not know the right answer and thus have to guess. For obvious reasons, guessing is only relevant if the answer space is reasonably small. In the present data, participants saw a set of 8 possible answers of which exactly one was considered correct. Thus, guessing cannot be ruled out and would be equal to $\gamma_i = 1/8$ for each item if all answer alternatives had a uniform probability to be chosen given that a person guesses. Lastly, the ψ_i parameter enables us to model the possibility that a participant makes a mistake even though they know the right answer, perhaps because of inattention or simply misclicking when selecting the chosen answer. We may call ψ_i the "lapse", "inattention", or "slipping" parameter. Usually, these terms can be used interchangeably but, as always, the exact meaning can only be inferred in the context of the specific model. As the answer format in the present data (i.e., "click on the right answer") is rather simple and participants have unlimited time for each item, mistakes due to lapses are unlikely to appear. However, by including a lapse parameter into our model, we are able to explicitly check whether lapses played a substantial role in the answers.

We can now simplify the 4PL model in several steps to yield the other less complex models. The 3PL model results from the 4PL model by additionally fixing the lapse probability to zero, that is, $\psi_i = 0$ for all items. In the next step, we can obtain the 2PL model from the 3PL model by also fixing the guessing probabilities to zero, that is, $\gamma_i = 0$ for all items. In the last simplification step, we obtain the 1PL model (also known as Rasch model [16]) from the 2PL model by assuming the factor loadings to be one, that is, $\alpha_i = 1$ for all items. Even though didactically I find it most intuitive and helpful to introduce the models from most to least complex, I recommend the inverse order in applications, that is, starting from the simplest (but still sensible) model. The reason is that more complex models tend to be more complicated to fit in the sense that they both take longer (especially when using MCMC estimation) and yield more convergence problems (e.g., [17,18]). If we started by fitting the most complex model and, after considerable waiting time, found the model to not have converged, we may have no idea which of the several model components were causing the problem(s). In contrast, by starting simple and gradually building towards more complex models, we can make sure that each model component is reasonably specified and can be reliably estimated before we move further.

As a result, when a problem occurs, we are likely to have much clearer understanding of why/where it occurred and how to fix it.

With the model likelihood fully specified by Equations (2) and (3) (potentially with some fixed item parameters), we are, in theory, already able to obtain estimates of person and item parameters via maximum likelihood (ML) estimation. However, there are multiple potential issues that can get into our way at this point. First, we simply may not have enough data to obtain sensible parameter estimates. As a rule of thumb, the more complex a model, the more data we need to obtain the same estimation precision. Second, there may be components in the model which will not be identified no matter how much data we add. An example would be binary IRT models from 2PL upwards as (without additional structure) we cannot identify the scale of both θ_j and α_i at the same time. This is because, due to the multiplicative relationship, multiplying one of the two by a constant can be adjusted for by dividing the other by the same constant without changing the likelihood. Third, we need to have software that is able to do the model fitting for us, unless we want to hand code every estimation algorithm on our own. Using existing software requires (re)expressing our models in a way the software understands. I will focus on the last issue first and then address the former two.

2.2. IRT Models as Regression Models

There are a lot of IRT specific software packages available, in particular in the programming language R ([19]), for example, mirt ([20]), sirt ([21]), or TAM ([22]; see [17] for a detailed comparison). In addition to these more specialized packages, general purpose probabilistic programming languages can be used to specify and fit Bayesian IRT models, for example, BUGS ([23]; see also [24]), JAGS ([25]; see also [26,27]), or Stan ([28]; see also [29,30]). In this paper, I use the brms package ([31,32]), a higher level interface to Stan, which is not focused specifically on IRT models but more generally on (Bayesian) regression models. Accordingly, we need to rewrite our IRT models in a form that is understandable for brms or other packages focussed on regression models.

The first implication of this change of frameworks is that we now think of the data in long format, with all responses from all participants on all items in the same data column coupled with additional columns for person and item indicators. That is, y_{ji} is now formally written as y_n where n is the observation number ranging from 1 to $N = J \times I$. If we needed to be more explicit we could also use $y_{j_n i_n}$ to indicate that each observation number n has specific indicators j and i associated with it. The same goes for item and person parameters. For example, we may write θ_{n_j} to refer to the ability parameter of the person j to whom the nth observation belongs.

One key aspect of regression models is that we try to express parameters on an unconstrained space that spans the whole real line. This allows for using linear (or more generally additive) predictor terms without having to worry about whether these predictor terms fulfill certain boundaries, for instance, are positive or within the unit interval $[0, 1]$. In the considered binary IRT models, we need to ensure that the factor loadings α are positive and that guessing and lapse parameters, γ and ψ, respectively, are within $[0, 1]$ as otherwise the interpretation of the latter two as probabilities would not be sensible. To enforce these parameter boundaries within a regression, we apply (inverse-)link functions. That is, for α, we use the log-link function (or equivalently the exponential response function) so that

$$\alpha = \exp(\eta_\alpha) \tag{4}$$

where η_{α_n} is unconstrained. Similarly, for γ and ψ, we use the logit-link (or equivalently the logistic response function) so that

$$\gamma = \text{logistic}(\eta_\gamma) = \frac{1}{1 + \exp(-\eta_\gamma)}, \tag{5}$$

$$\psi = \text{logistic}(\eta_\psi) = \frac{1}{1 + \exp(-\eta_\psi)} \tag{6}$$

where η_γ and η_ψ are unconstrained. The location parameters β are already unbounded and as such do not need an additional link function so that simply $\beta = \eta_\beta$. The same goes for the ability parameters θ. On the scale of the linear predictors, we can perform the usual regression operations, perhaps most importantly modeling predictor variables or including multilevel structure. In the present data, we do not have any additional person or item variables available so there are no such predictors in our models (but see Bürkner, 2019 for examples if you are interested in this option). However, there certainly is multilevel structure as we have both multiple observations per item and per person, which we seek to model appropriately, as detailed in the next section.

2.3. Model Priors and Identification

When it comes to the specification of priors on item parameters, we typically distinguish between non-hierarchical and hierarchical priors ([9,10,17]) with the former being applied more commonly (e.g., [10,32]). When applying non-hierarchical priors, we directly equate the linear predictor η (for any of the item parameter classes) with item-specific parameters b_i, so that

$$\eta_n = b_{i_n} \tag{7}$$

for each observation n and corresponding item i. Since η is on an unconstrained scale so are the b_i parameters and we can apply location-scale priors such as the normal distribution with mean μ and standard deviation σ:

$$b_i \sim \text{normal}(\mu, \sigma) \tag{8}$$

In non-hierarchical priors, we fix μ and σ to sensible values. In general, priors can only be understood in the context of the model as a whole, which renders general recommendation for prior specification difficult ([33]). If we only use our understanding of the scale of the modeled parameters without any data-specific knowledge, we arrive at weakly-informative prior distributions. By weakly-informative I mean penalizing a-priori implausible values (e.g., a location parameter of 1000 on the logit-scale) without affecting the a-priori plausible parameter space too much (e.g., location parameters within the interval $[-3, 3]$ on the logit-scale). Weakly informative normal priors are often centered around $\mu = 0$ with σ appropriately chosen so that the prior covers the range of plausible parameter values but flattens out quickly outside of that space. For more details on priors for Bayesian IRT models, see the works of Bürkner ([17]), Fox ([9]), and Levy and Mislevy ([10]).

A second class of priors for item parameters are hierarchical priors. For this purpose, we apply the non-centered parameterization of hierarchical models (Gelman et al., 2013) as detailed in the following. We split the linear predictor η (for any of the item parameter classes) into an overall parameter, $\overline{b}$, and an item-specific deviation from the overall parameter, $\tilde{b}_i$, so that

$$\eta_n = \overline{b} + \tilde{b}_{i_n} \tag{9}$$

Without additional constraints, this split is not identified as adding a constant to the overall parameter can be compensated by subtracting the same constant from all $\tilde{b}_i$ without changing the likelihood. In Bayesian multilevel models, we approach this problem by specifying a hierarchical prior on $\tilde{b}_i$ via

$$\tilde{b}_i \sim \text{normal}(0, \sigma) \tag{10}$$

where σ is the standard deviation parameter over items on the unconstrained scale. Importantly, not only $\tilde{b}_i$ but also the hyperparameters $\overline{b}$ and σ are estimated during the model fitting.

Using the prior distribution from (10), we would assume the item parameters of the same item to be unrelated but, in practice, it is quite plausible that they are intercorrelated ([17]). To account for such (linear) dependency, we can extend Equation (10) to the multivariate case, so that we can model the vector $\tilde{b}_i = (\tilde{b}_{\beta_i}, \tilde{b}_{\alpha_i}, \tilde{b}_{\gamma_i}, \tilde{b}_{\psi_i})$ jointly via a multivariate normal distribution:

$$\tilde{b}_i \sim \text{multinormal}(0, \boldsymbol{\sigma}, \Omega) \tag{11}$$

where $\boldsymbol{\sigma} = (\sigma_\beta, \sigma_\alpha, \sigma_\gamma, \sigma_\psi)$ is the vector of standard deviations and Ω is the correlation matrix of the item parameters (see also [17,31,34]). To complete the prior specification for the item parameters, we need to set priors on $\bar{b}$ and σ. For this purpose, weakly-informative normal prior on $\bar{b}$ and half-normal priors on σ are usually fine but other options are possible as well (see [17] for details).

A decision between hierarchical and non-hierarchical priors is not always easy. If in doubt, one can try out both kinds of priors and investigate whether they make a relevant difference. Personally, I prefer hierarchical priors as they imply some data-driven shrinkage due to their scale being learned by the model on the fly. In addition, they naturally allow item parameters to share information across parameter classes via the correlation matrix Ω.

With respect to the person parameters, it is most common to apply hierarchical priors of the form

$$\theta_j \sim \text{normal}(0, \sigma_\theta) \tag{12}$$

where, similar as for hierarchical priors on item parameters, σ_θ is a standard deviation parameter estimated as part of the model on which we put a weakly-informative prior. To give the reader intuition: With the overall effects in our model, we model the probability that an average person (with an ability of zero, thus imagine the ability to be centered) answers an average item (with all item parameters at their average values which we estimate). The varying effects then give us the deviations from the average person or item, so that we can "customize" our prediction of the solution probability to more or less able persons, more or less easy items, more or less discriminatory items, etc.

In 2PL or more complex models, we can also fix σ_θ to some value (usually 1) as the scale is completely accounted for by the scale of the factor loadings σ_α. However, when using weakly-informative priors on both θ and α as well as on their hyperparameters, estimating σ_θ actually poses no problem for model estimation. Importantly, however, we do not include an overall person parameter $\bar{\theta}$ as done for item parameters in (9) as this would conflict with the overall location parameter $\bar{b}_\beta$ leading to substantial convergence problems in the absence very informative priors. This does not limit the model's usefulness as only differences of person parameters are of relevance, not their absolute values on an (in principal) arbitrary latent scale.

3. Analysis of the SPM-LS Data

The Bayesian IRT models presented above were applied to the SPM data of Myszkowski and Storme ([4]). The analyzed data consist of responses from 499 participants on the 12 most difficult SPM items and are freely available online (https://data.mendeley.com/datasets/h3yhs5gy3w/1). The data gathering procedure was described in detail by Myszkowski and Storme ([4]). Analyses were performed in R ([19]) using brms ([31]) and Stan ([28]) for model specification and estimation via MCMC. To investigate potential differences between hierarchical and non-hierarchical priors on the item parameters, models were estimated for both of these priors. Below, I refer to these approaches as hierarchical MCMC (MCMC-H) and non-hierarchical MCMC (MCMC-NH). Priors on person parameters were always hierarchical and weakly informative priors were imposed on the remaining parameters. All considered models converged well according to sample-agnostic ([35]) and sampler-specific ([36]) diagnostics. In the presentation of the results below, I omit details of prior distributions and auxiliary model fitting arguments. All details and the fully reproducible analysis are available on GitHub (https://github.com/paul-buerkner/SPM-IRT-models).

In addition to estimating the IRT models using MCMC, I also fitted the models via optimization as implemented in the mirt package ([20]). Here, I considered two options: (1) a fully frequentist approach maximizing the likelihood under the same settings as in the original analysis of Myszkowski and Storme ([4]); and (2) a Bayesian optimization approach where I imposed the same priors on item parameters as in MCMC-NH. I refer to these two methods as maximum likelihood (ML) and maximum a-posteriori (MAP), respectively. For models involving latent variables, such as IRT models, ML or MAP optimization have to be combined with numerical integration over the latent variables as the mode of the joint distribution of all parameters including latent variables does not exist in general (e.g., see [37]). Such a combination of optimization and integration is commonly referred to as expectation-maximization (EM). A thorough discussion on EM methods is outside the scope of the present paper but the interested reader is referred to the work of Do and Batzoglou ([38]).

3.1. Model Estimation

For estimation in a multilevel regression framework such as the one of brms, the data need to be represented in long format. In the SPM-LS data, the relevant variables are the binary response of the participants (variable `response2`) coded as either correct (`1`) or incorrect (`0`) as well as `person` and `item` identifiers. Following the principal of building models bottom-up, I start with the estimation of the most simple sensible model, that is, the 1PL model. When both person and item parameters are modeled hierarchically, the brms formula for the 1PL model can be specified as

```r
formula_1pl <- bf(
  formula = response2 ~ 1 + (1 | item) + (1 | person),
  family = brmsfamily("bernoulli", link = "logit")
)
```

To apply non-hierarchical item parameters, we have to use the formula `response2 ~ 0 + item + (1 | person)` instead (see the code on Github for more details). For a thorough introduction and discussion of the brms formula syntax, see [17,31,32]. As displayed in Figure 1, item parameter estimates of all methods are very similar for the 1PL model. In addition, their uncertainty estimates align closely as well. The brms formula for the 2PL model looks as follows:

```r
formula_2pl <- bf(
  response2 ~ beta + exp(logalpha) * theta,
  nl = TRUE,
  theta ~ 0 + (1 | person),
  beta ~ 1 + (1 |i| item),
  logalpha ~ 1 + (1 |i| item),
  family = brmsfamily("bernoulli", link = "logit")
)
```

When comparing the formulas for the 1PL and 2PL models, we see that the structure has changed considerably as a result of going from a generalized linear model to a generalized non-linear model (see [17] for more details). As displayed in Figure 2, item parameter point and uncertainty estimates of all methods are rather similar for the 2PL model but not as close as for the 1PL model. In particular, we see that the slope estimates of Items 4 and 5 vary slightly, presumably due to different amounts of regularization implied by the priors. The brms formula for the 3PL model looks as follows:

```r
formula_3pl <- bf(
  response2 ~ gamma + (1 - gamma) *
    inv_logit(beta + exp(logalpha) * theta),
  nl = TRUE,
  theta ~ 0 + (1 | person),
  beta ~ 1 + (1 |i| item),
  logalpha ~ 1 + (1 |i| item),
  logitgamma ~ 1 + (1 |i| item),
  nlf(gamma ~ inv_logit(logitgamma)),
  family = brmsfamily("bernoulli", link = "identity"),
)
```

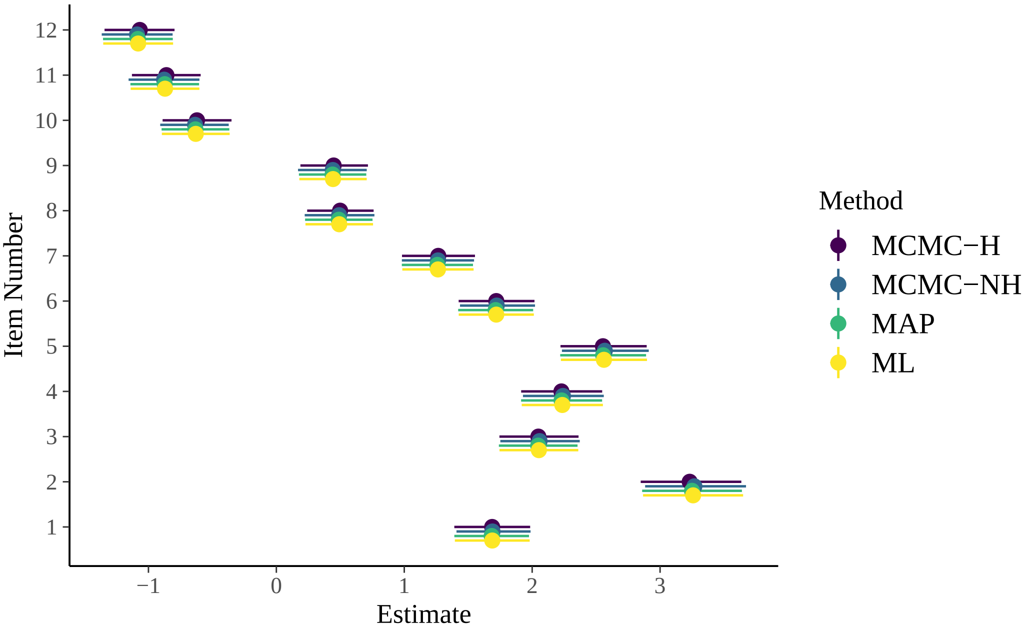

Figure 1. Item parameters of the 1PL model. Horizontal lines indicate 95% uncertainty intervals.

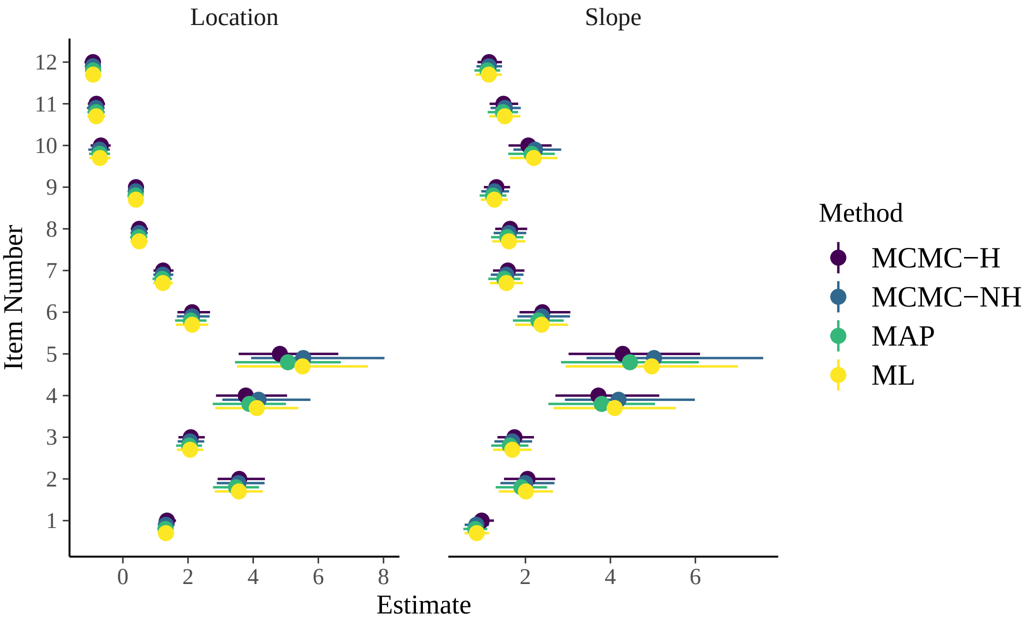

Figure 2. Item parameters of the 2PL model. Horizontal lines indicate 95% uncertainty intervals.

Note that, in the `family` argument, we now use `link = "identity"` instead of `link = "logit"` and build the logit link directly into the formula via `inv_logit(beta + exp(logalpha) * theta)`. This is necessary to correctly include guessing parameters ([17]). As displayed in Figure 3, item parameter estimates of all methods are still quite similar when it comes to locations and slopes of the 3PL model. However, guessing parameter estimates are quite different: ML obtains point estimates of 0 for all but three items with uncertainty intervals ranging the whole definition space from 0 to 1. This is caused by an artifact in the computation of the approximate standard errors because point estimates are located at the boundary of the parameter space at which maximum likelihood theory does not hold. In contrast, point estimates of guessing parameters as obtained by all regularized models are close to but not exactly zero for most items and corresponding uncertainty estimates appear more realistic (i.e., much narrower) than those obtained by pure ML.

On Github, I also report results for the 3PL model with guessing probabilities fixed to 1/8 derived under the assumptions that, in the case of guessing, all alternatives are equally likely. According to Figure 3 and model comparisons shown on GitHub, this assumption does not seem to hold for the present data.

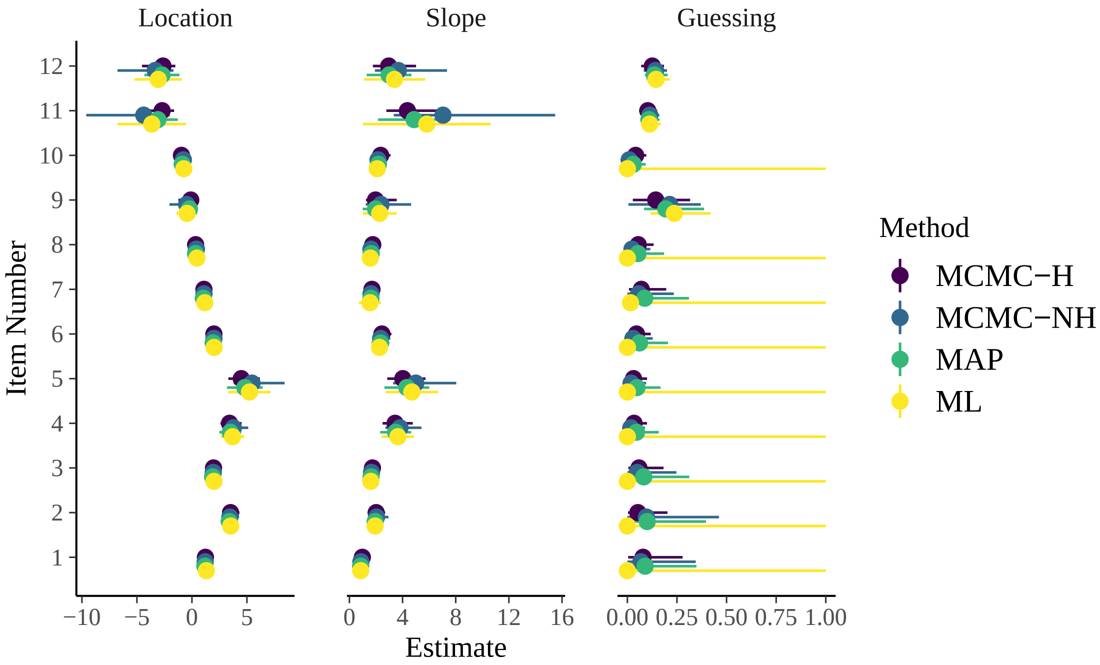

Figure 3. Item parameters of the 3PL model. Horizontal lines indicate 95% uncertainty intervals.

In Figure 4, I display person parameter estimates of the 3PL model. As we can see on the left-hand side of Figure 4, ML and MCMC-H point estimates align very closely. However, as displayed on the right-hand side of Figure 4, uncertainty estimates show some deviations, especially for more extreme point estimates (i.e., particularly good or bad performing participants). The brms formula for the 4PL model looks as follows:

```
formula_4pl <- bf(
  response2 ~ gamma + (1 - gamma - psi) *
    inv_logit(beta + exp(logalpha) * theta),
  nl = TRUE,
  theta ~ 0 + (1 | person),
  beta ~ 1 + (1 |i| item),
  logalpha ~ 1 + (1 |i| item),
  logitgamma ~ 1 + (1 |i| item),
  nlf(gamma ~ inv_logit(logitgamma)),
  logitpsi ~ 1 + (1 |i| item),
  nlf(psi ~ inv_logit(logitpsi)),
  family = brmsfamily("bernoulli", link = "identity")
)
```

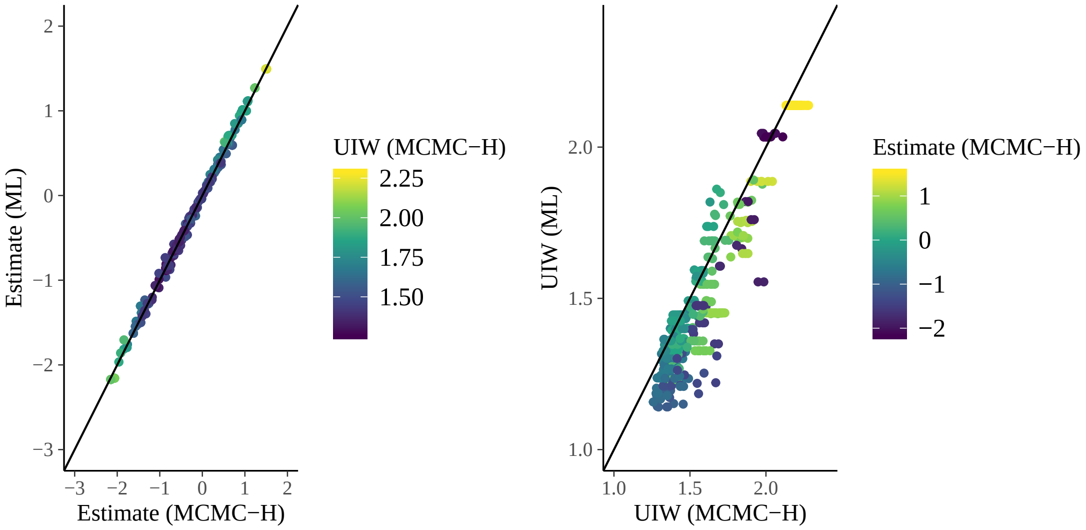

Figure 4. Comparison of 3PL person parameters: (Left) scatter plot of point estimates; and (Right) scatter plot of the associated 95% uncertainty interval widths (UIW).

As displayed in Figure 5, item parameter estimates of the 4PL model differ strongly from each other for different methods. In particular, ML point estimates were more extreme and no uncertainty estimates could be obtained due to singularity of the information matrix. It is plausible that the 4PL model is too difficult to be estimated based on the given data via ML without further regularization. Moreover, the estimates obtained by MCMC-H and MCMC-NH differ noticeably for some item parameters in the way that MCMC-NH estimates tend to be more extreme and uncertain as compared to MCMC-H. This suggests that, for these specifically chosen hierarchical and non-hierarchical priors, the former imply stronger regularization.

3.2. Model Comparison

Next, I investigate the required model complexity to reasonably describe the SPM data. For this purpose, I apply Bayesian approximate leave-one-out cross-validation (LOO-CV; ref. [39–41]) as a method for model comparison, which is closely related to information criteria ([40]). I only focus on the MCMC-H models here. Results for the MCMC-NH models are similar (see Github for details). As shown in Table 1, 3PL and 4PL models fit substantially better than the 1PL and 2PL models, while there was little difference between the former two. Accordingly, in the interest of parsimony, I would tend to prefer the 3PL model if a single model needed to be chosen. This coincides with the conclusions of Myszkowski and Storme ([4]).

Table 1. Bayesian Model comparison based on the leave-one-out cross-validation.

Model	ELPD	SE(ELPD)	ELPD-Difference	SE(ELPD-Difference)
4PL	−2544.7	42.6	0.0	0.0
3PL	−2547.8	42.8	−3.1	5.1
2PL	−2588.7	42.9	−44.0	9.5
1PL	−2655.0	43.8	−110.3	15.0

Note. ELPD, expected log posterior density; SE, standard error. Higher ELPD values indicate better model fit. ELPD differences are in comparison to the 4PL model.

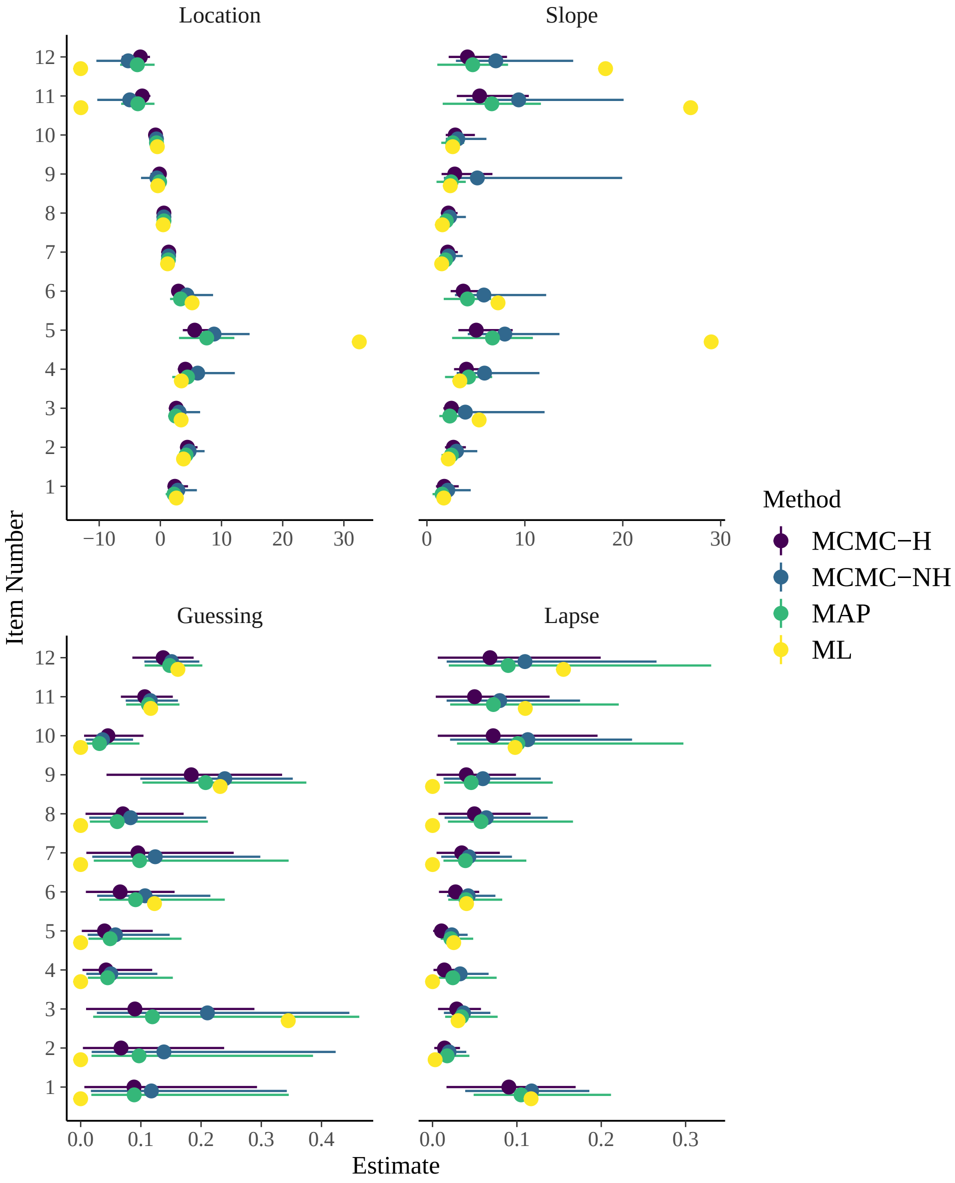

Figure 5. Item parameters of the 4PL model. Horizontal lines indicate 95% uncertainty intervals.

We can also investigate model fit using Bayesian versions of frequentist item or person fit statistics such as log-likelihood values ([42]). Independently of which statistic T is chosen, a Bayesian version of the statistic can be constructed as follows ([42]): First, the fit statistic is computed for the observed responses y. We denote it by $T(y, p)$, where $p = p(\theta, \xi)$ is the model implied response probability defined in Equation (3). As p depends on the model parameters, the posterior distribution over the parameters implies a posterior distribution over p, which in turn implies a posterior distribution over $T(y, p)$. Second, the fit statistic is computed for posterior predicted responses y_{rep} and we denote it by $T(y_{\text{rep}}, p)$. Since y_{rep} reflects the (posterior distribution of) responses that would be predicted if the model was true, $T(y_{\text{rep}}, p)$ provides a natural baseline for $T(y, p)$. Third, by comparing the posterior distributions of $T(y, p)$ and $T(y_{\text{rep}}, p)$, we can detect item- or person-specific model misfit. In Figure 6, we show item-specific log-likelihood differences between predicted and observed responses for the 1PL model. It is clearly visible that the assumptions of the 1PL model are violated for almost half of the items. In contrast, the corresponding results for the 3PL model look much more reasonable (see Figure 7).

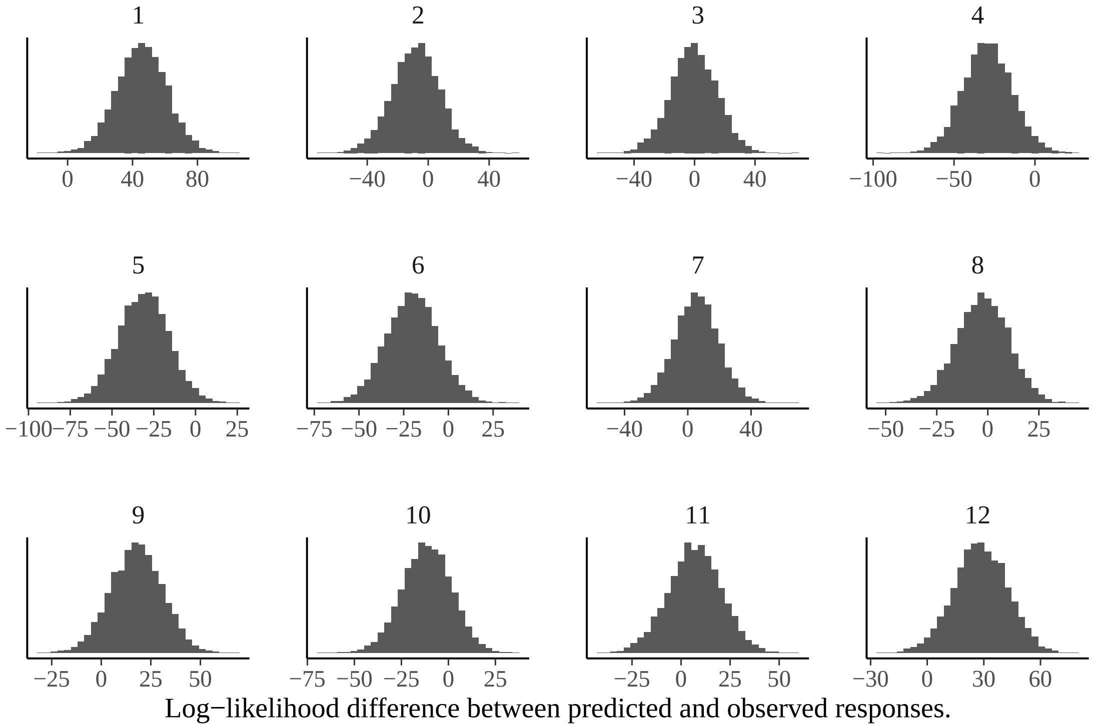

Figure 6. Item-specific posterior distributions of log-likelihood differences between predicted and observed responses for the 1PL model estimated via MCMC-H. If the majority of the posterior distribution is above zero, this indicates model misfit for the given item.

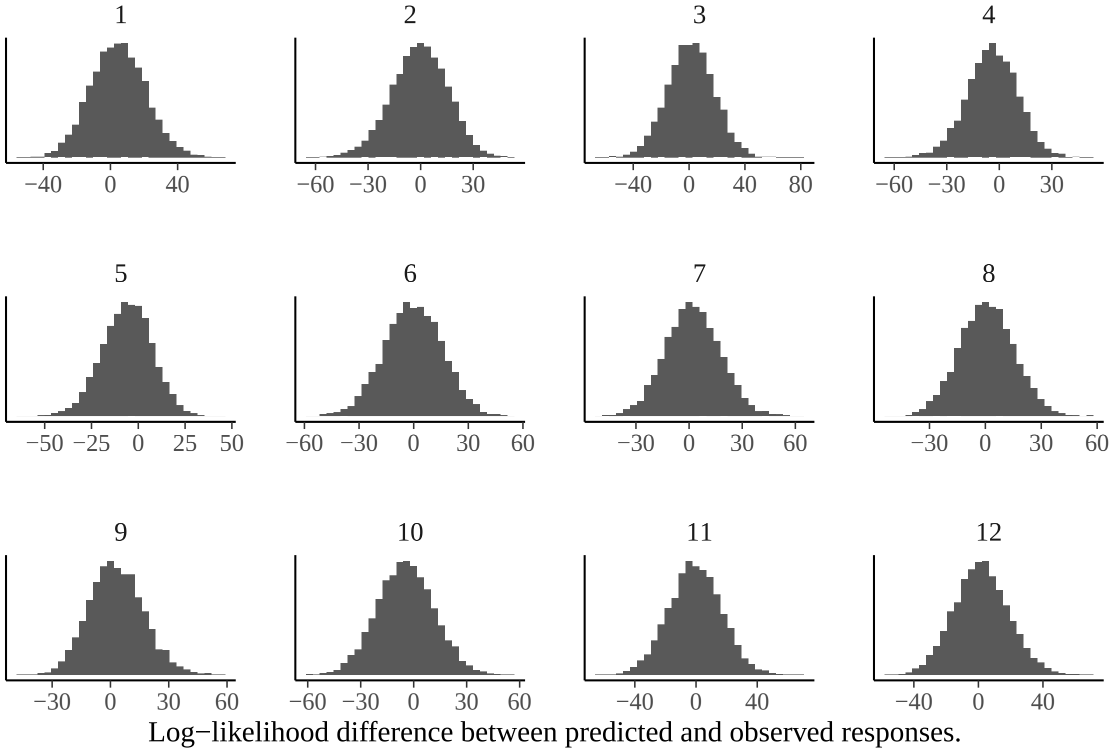

Figure 7. Item-specific posterior distributions of log-likelihood differences between predicted and observed responses for the 3PL model estimated via MCMC-H. If the majority of the posterior distribution is above zero, this indicates model misfit for the given item.

We can use the same logic to investigate person-specific model fit to find participants for whom the models do not make good predictions. In Figure 8, we show the predicted vs. observed log-likelihood differences of the 192nd person with response pattern $(0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1)$. None of the models performs particularly well as this person did not answer some of the easiest items correctly (i.e., Items 2, 4, and 5) but was correct on some of the most difficult items (i.e., Items 8, 9, and 12). It is unclear what was driving such a response pattern. However, one could hypothesize that training effects over the course of the test played a role, which are not accounted for by all models presented here. To circumvent this in future test administrations, one could add more unevaluated practice items at the beginning of the test so that participants have the opportunity to become more familiar with the response format. Independently of the difference in model fit, person parameter estimates correlated quite strongly between different models and estimation approaches, with pairwise correlations exceeding $r = 0.97$ in all cases (see Figure 9 for an illustration).

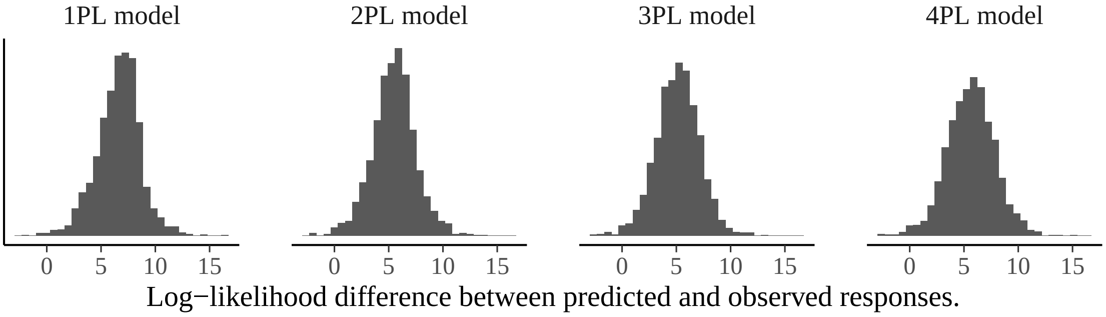

Figure 8. Person-specific posterior distributions of log-likelihood differences between predicted and observed responses for the 192nd person and different models estimated via MCMC-H. If the majority of the posterior distribution is above zero, this indicates model misfit for the given person.

The time required for estimation of the Bayesian models with brms via MCMC ranged from a couple of minutes for the 1PL model to roughly half an hour for the 4PL model (exact timings vary according to several factors, for instance, the number of iterations and chains, applied computing machines, or the amount of parallelization). In contrast, the corresponding optimization methods (ML and MAP) required only a few seconds for estimation in mirt. This speed difference of multiple orders of magnitude is typical for comparisons between MCMC and optimization methods (e.g., [31]). Clearly, if speed is an issue for the given application, full Bayesian estimation methods via MCMC should be applied carefully.

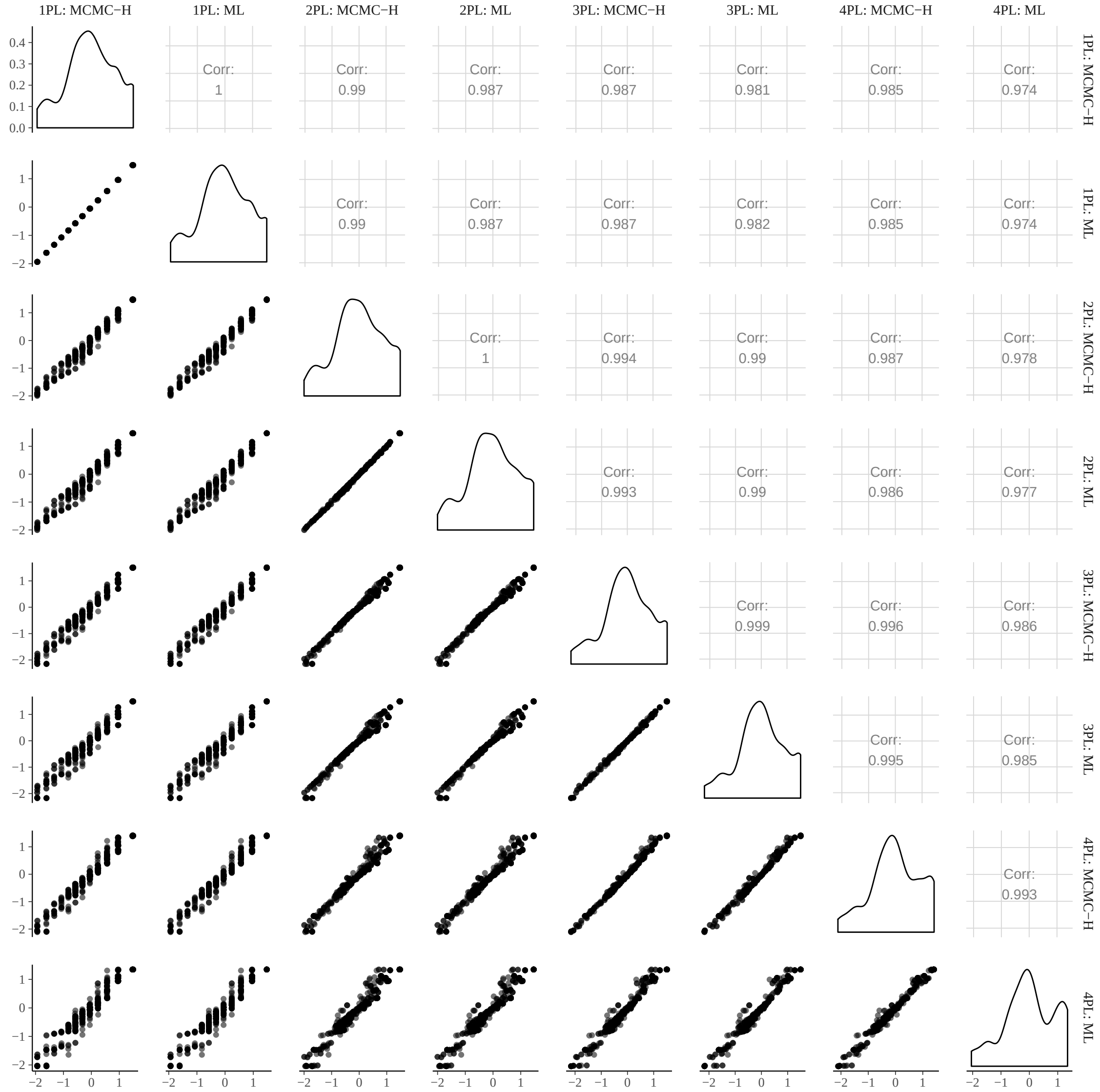

Figure 9. Scatter plots, bivariate correlations, and marginal densities of person parameters from MCMC-NH and ML models.

4. Discussion

In the present paper, I reanalyze data to validate a short version of Standard Progressive Matrices (SPM-LS; [4]) using Bayesian IRT models. By comparing out-of-sample predictive performance, I found evidence that the 3PL model with estimated guessing parameters outperformed simpler models and performed similarly well as the 4PL model, which additionally estimated lapse parameters. As specifying and fitting the 4PL model is substantially more involved than the 3PL model without apparent gains in out-of-sample predictive performance, I argue that the 3PL model should probably be the model of choice within the scope of all models considered here. That is, I come to a similar conclusion as Myszkowski and Storme ([4]) in their original analysis despite using different frameworks for model specification and estimation (Bayesian vs. frequentist) as well as predictive performance (approximate leave-one-out cross-validation ([40]) vs. corrected AIC and χ^2-based measures ([43]).

Even though I reach the same conclusions as Myszkowski and Storme ([4]) reached with conventional frequentist methods, I would still like to point out some advantages of applying Bayesian methods that we have seen in this application. With regard to item parameters, Bayesian and frequentist estimates showed several important differences for the most complex 3PL and 4PL IRT models. First, point estimates of items with particularly high difficulty or slope were more extreme in the frequentist maximum likelihood estimation. One central reason is the use weakly informative priors in the Bayesian models which effectively shrunk extremes a little towards the mean thus providing more conservative and robust estimates ([18]). Specifically, for the 4PL model, the model structure was also too complex to allow for reasonable maximum likelihood estimation in the absence of any additional regularization to stabilize inference. The latter point also becomes apparent because no standard errors of the ML estimated items parameters in the 4PL model could be computed due to singularity of the information matrix. Even when formally computable, uncertainty estimates provided by the frequentist IRT models were not always meaningful. For instance, in the 3PL model, the confidence intervals of guessing parameters estimated to be close to zero were ranging the whole definition space between zero and one. This is clearly an artifact as maximum likelihood theory does not apply at the boundary of the parameter space and hence computation of standard errors is likely to fail. As such, these uncertainty estimates should not be trusted. Robust alternatives to computing approximate standard errors via maximum likelihood theory are bootstrapping or other general purpose data resampling methods (e.g., [44–46]). These resampling methods come with additional computational costs as the model has to be repeatedly fitted to different datasets but can be used even in problematic cases where standard uncertainty estimators fail.

In contrast, due to the use of weakly informative priors, the Bayesian models provided sensible uncertainty estimates for all item parameters of every considered IRT model. MCMC and MAP estimates provided quite similar results for the item parameters in the context of the SPM-LS data and applied binary IRT models. However, there is no guarantee that this will be generally the case and thus it is usually safer to apply MCMC methods when computationally feasible. In addition, for the propagation of uncertainty to new quantities, for instance, posterior predictions, MCMC or other sampling-based methods are required. In the case study, I demonstrated this feature in the context of item and person fit statistics, which revealed important insides into the assumptions of the applied IRT models.

With regard to point estimates of person parameters, I found little differences between all considered Bayesian and frequentist IRT models. Pairwise correlations between point estimates of two different models were all exceeding $r = 0.97$ and often even larger than $r = 0.99$. This should not imply, however, that the model choice does not matter in the context of person parameter estimation ([14]). Although point estimates were highly similar, uncertainty estimates of person parameters varied substantially across model classes. Thus, it is still important to choose an appropriately complex model for the data (i.e., the 3PL model in our case) in order to get sensible uncertainty estimates. The latter are not only relevant for individual diagnostic purposes, which is undoubtedly a major application of intelligence tests, but also when using person parameters as predictors in other models while taking their estimation uncertainty into account. In addition, uncertainty estimates of Bayesian and frequentist models varied substantially even within the same model class, in particular for 3PL and 4PL models. Without a known ground truth, we have no direct evidence which of the uncertainty estimates are more accurate (with respect to some Bayesian and/or frequentist criteria), but I would argue in favor of the Bayesian results as they should have benefited from the application of weakly informative priors and overall more robust inference procedures for the considered classes of models. Overall, it is unsurprising that Bayesian methods have an easier time estimating uncertainty as it is more natural to do so in a Bayesian framework. We have also seen the important advantage of Bayesian methods that is their ability to more easily accommodate more complex models. However, we have also seen that, for simpler models, Bayesian and frequentist methods provide very similar results, which really speaks in favor of both methods and should highlight for the reader that both choices are valid options in this case and neither should be attacked. Developing this understanding seems necessary with the increased application of Bayesian methods and the accompanying arguments of whether this is a valid option.

The analysis presented here could be extended in various directions. First, one could fit polytomous IRT models that take into account potential differences between distractors and thus use more information than binary IRT models. Such polytomous IRT models were also fitted by Myszkowski and Storme ([4]) and demonstrated some information gain as compared to their binary counterparts. Fitting these polytomous IRT models in a Bayesian framework is possible as well, but currently not supported by brms in the here required form. Instead, one would have to use Stan directly, or another probabilistic programming language, whose introduction is out of scope of the present paper. Second, one could consider multiple person traits/latent variables to investigate the unidimensionality of the SPM-LS test. Currently, this cannot be done in brms in an elegant manner but will be possible in the future once formal measurement models have been implemented. For the time being, one has to fall back to full probabilistic programming languages such as Stan or more specialized IRT software that supports multidimensional Bayesian IRT models. According to Myszkowski and Storme ([4]), the SPM-LS test is sufficiently unidimensional to justify the application of unidimensional IRT models. Accordingly, the lack of multidimensional models does not constitute a major limitation for the present analysis.

In summary, I was able to replicate several key findings of Myszkowski and Storme ([4]). Additionally, I demonstrated that Bayesian IRT models have some important advantages over their frequentist counterparts when it comes to reliably fitting more complex response processes and providing sensible uncertainty estimates for all model parameters and other quantities of interest.

Funding: This research received no external funding.

Acknowledgments: I want to thank Marie Beisemann and two anonymous reviewers for their valuable comments on earlier versions of the paper. To conduct the presented analyses and create this paper, I used the programming language R ([19]) through the interface RStudio ([47]). Further, the following R packages were crucial (in alphabetical order): brms ([31]), dplyr ([48]), GGally ([49]), ggplot2 ([50]), kableExtra ([51]), knitr ([52]), loo ([40]), mirt ([20]), papaja ([53]), patchwork ([54]), rmarkdown ([55]), rstan ([28]), and tidyr ([56]).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Raven, J.C. Standardization of progressive matrices, 1938. *Br. J. Med Psychol.* **1941**, *19*, 137–150.
2. Jensen, A.R.; Saccuzzo, D.P.; Larson, G.E. Equating the standard and advanced forms of the Raven progressive matrices. *Educ. Psychol. Meas.* **1988**, *48*, 1091–1095.
3. Pind, J.; Gunnarsdóttir, E.K.; Jóhannesson, H.S. Raven's standard progressive matrices: New school age norms and a study of the test's validity. *Personal. Individ. Differ.* **2003**, *34*, 375–386.
4. Myszkowski, N.; Storme, M. A snapshot of g? Binary and polytomous item-response theory investigations of the last series of the standard progressive matrices (SPM-LS). *Intelligence* **2018**, *68*, 109–116.
5. Ackerman, P.L.; Kanfer, R. Test length and cognitive fatigue: An empirical examination of effects on performance and test-taker reactions. *J. Exp. Psychol. Appl.* **2009**, *15*, 163.
6. Embretson, S.E.; Reise, S.P. *Item Response Theory*; Psychology Press: Hove, UK, 2013.
7. van der Linden, W.J.; Hambleton, R.K. *Handbook of Modern Item Response Theory*; Springer: Berlin/Heidelberg, Germany, 1997.
8. Lord, F.M. *Applications of Item Response Theory to Practical Testing Problems*; Routledge: Milton Park, UK, 2012.
9. Fox, J.-P. *Bayesian Item Response Modeling: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2010.
10. Levy, R.; Mislevy, R.J. *Bayesian Psychometric Modeling*; Chapman; Hall/CRC: Boca Raton, FL, USA, 2017.
11. Rupp, A.A.; Dey, D.K.; Zumbo, B.D. To Bayes or not to Bayes, from whether to when: Applications of Bayesian methodology to modeling. *Struct. Equ. Model.* **2004**, *11*, 424–451.
12. Culpepper, S.A. Revisiting the 4-parameter item response model: Bayesian estimation and application. *Psychometrika* **2016**, *81*, 1142–1163.
13. Culpepper, S.A. The prevalence and implications of slipping on low-stakes, large-scale assessments. *J. Educ. Behav. Stat.* **2017**, *42*, 706–725.
14. Loken, E.; Rulison, K.L. Estimation of a four-parameter item response theory model. *Br. J. Math. Stat. Psychol.* **2010**, *63*, 509–525.

15. Waller, N.G.; Feuerstahler, L. Bayesian modal estimation of the four-parameter item response model in real, realistic, and idealized datasets. *Multivar. Behav. Res.* **2017**, *52*, 350–370.
16. Rasch, G. On general laws and the meaning of measurement in psychology. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*; University of California Press: Berkeley, CA, USA, 1961; Volume 4, pp. 321–333.
17. Bürkner, P.-C. Bayesian item response modelling in R with brms and Stan. *arXiv4* **2019**, arXiv:1905.09501.
18. Gelman, A.; Carlin, J.B.; Stern, H.S.; Dunson, D.B.; Vehtari, A.; Rubin, D.B. *Bayesian Data Analysis*, 3rd ed.; Chapman; Hall/CRC: Boca Raton, FL, USA, 2013; doi:10.1201/b16018.
19. R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing: Vienna, Austria, 2019. Available online: https://www.R-project.org/ (accessed on 3 February 2020).
20. Chalmers, R.P. mirt: A multidimensional item response theory package for the R environment. *J. Stat. Softw.* **2012**, *48*, 1–29. doi:10.18637/jss.v048.i06.
21. Robitzsch, A. Sirt: Supplementary Item Response Theory Models. 2019. Available online: https://CRAN.R-project.org/package=sirt (accessed on 3 February 2020).
22. Robitzsch, A.; Kiefer, T.; Wu, M. TAM: Test Analysis Modules. 2019. Available online: https://CRAN.R-project.org/package=TAM (accessed on 3 February 2020).
23. Lunn, D.; Spiegelhalter, D.; Thomas, A.; Best, N. The BUGS project: Evolution, critique and future directions. *Stat. Med.* **2009**, *28*, 3049–3067.
24. Curtis, S.M. BUGS code for item response theory. *J. Stat. Softw.* **2010**, *36*, 1–34.
25. Plummer, M. JAGS: Just Another Gibbs Sampler. 2013. Available online: http://mcmc-jags.sourceforge.net/ (accessed on 03 February 2020).
26. Depaoli, S.; Clifton, J.P.; Cobb, P.R. Just another Gibbs sampler (JAGS) flexible software for MCMC implementation. *J. Educ. Behav. Stat.* **2016**, *41*, 628–649.
27. Zhan, P.; Jiao, H.; Man, K.; Wang, L. Using JAGS for Bayesian cognitive diagnosis modeling: A tutorial. *J. Educ. Behav. Stat.* **2019**, doi:1076998619826040.
28. Carpenter, B.; Gelman, A.; Hoffman, M.D.; Lee, D.; Goodrich, B.; Betancourt, M.; Brubaker, M.; Guo, J.; Li, P.; Riddell, A. Stan: A probabilistic programming language. *J. Stat. Softw.* **2017**, *76*, 1–32. doi:10.18637/jss.v076.i01.
29. Ames, A.J.; Au, C.H. Using Stan for item response theory models. *Meas. Interdiscip. Res. Perspect.* **2018**, *16*, 129–134.
30. Luo, Y.; Jiao, H. Using the Stan program for bayesian item response theory. *Educ. Psychol. Meas.* **2018**, *78*, 384–408.
31. Bürkner, P.-C. brms: An R package for bayesian multilevel models using Stan. *J. Stat. Softw.* **2017**, *80*, 1–28. doi:10.18637/jss.v080.i01.
32. Bürkner, P.-C. Advanced Bayesian multilevel modeling with the R package brms. *R J.* **2018**, *10*, 395–411. doi:10.32614/RJ-2018-017.
33. Gelman, A.; Simpson, D.; Betancourt, M. The prior can often only be understood in the context of the likelihood. *Entropy* **2017**, *19*, 555–567. doi:10.3390/e19100555.
34. Nalborczyk, L.; Batailler, C.; Lœvenbruck, H.; Vilain, A.; Bürkner, P.-C. An introduction to bayesian multilevel models using brms: A case study of gender effects on vowel variability in standard indonesian. *J. Speech Lang. Hear. Res.* **2019**, *62*, 1225–1242.
35. Vehtari, A.; Gelman, A.; Simpson, D.; Carpenter, B.; Bürkner, P.-C. Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC. *arXiv* **2019**, arXiv:1903.08008.
36. Betancourt, M. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv4* **2017**, arXiv:1701.02434.
37. Bates, D.; Mächler, M.; Bolker, B.; Walker, S. Fitting linear mixed-effects models using lme4. *J. Stat. Softw.* **2015**, *67*, 1–48. doi:10.18637/jss.v067.i01.
38. Do, C.B.; Batzoglou, S. What is the expectation maximization algorithm? *Nat. Biotechnol.* **2008**, *26*, 897.
39. Vehtari, A.; Simpson D.; Gelman, A.; Yao, Y.; Gabry, J. Pareto smoothed importance sampling. *arXiv* **2019**. arxiv:1507.02646.
40. Vehtari, A.; Gelman, A.; Gabry, J. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Stat. Comput.* **2017**, *27*, 1413–1432. doi:10.1007/s11222-016-9696-4.

41. Vehtari, A.; Gelman, A.; Gabry, J. Loo: Efficient Leave-One-Out Cross-Validation and WAIC for Bayesian models. 2018. Available online: https://github.com/stan-dev/loo (accessed on 3 February 2020).
42. Glas, C.A.; Meijer, R.R. A Bayesian approach to person fit analysis in item response theory models. *Appl. Psychol. Meas.* **2003**, *27*, 217–233.
43. Maydeu-Olivares, A. Goodness-of-fit assessment of item response theory models. *Meas. Interdiscip. Res. Perspect.* **2013**, *11*, 71–101.
44. Freedman, D.A. Bootstrapping regression models. *Ann. Stat.* **1981**, *9*, 1218–1228.
45. Junker, B.W.; Sijtsma, K. Nonparametric item response theory in action: An overview of the special issue. *Appl. Psychol. Meas.* **2001**, *25*, 211–220.
46. Mooney, C.Z.; Duval, R.D. *Bootstrapping: A Nonparametric Approach to Statistical Inference*; Sage: Thousand Oaks, CA, USA, 1993.
47. RStudio Team. *RStudio: Integrated development for R*; RStudio, Inc.; Boston, MA, USA, 2018; Volume 42.
48. Wickham, H.; François, R.; Henry, L.; Müller, K. Dplyr: A Grammar of Data Manipulation. 2019. Available online: https://CRAN.R-project.org/package=dplyr (accessed on 3 February 2020).
49. Schloerke, B.; Crowley, J.; Cook, D.; Hofmann, H.; Wickham, H.; Briatte, F.; Marbach, M.; Thoen, E.; Elberg, A.; Larmarange, J. GGally: Extension to 'ggplot2'. 2018. Available online: https://CRAN.R-project.org/package=GGally (accessed on 3 February 2020).
50. Wickham, H. *Ggplot2: Elegant Graphics for Data Analysis*; Springer-Verlag: New York, NY, USA, 2016. Available online: http://ggplot2.org (accessed on 3 February 2020).
51. Zhu, H. kableExtra: Construct Complex Table with 'Kable' and Pipe Syntax. 2019. Available online: https://CRAN.R-project.org/package=kableExtra (accessed on 3 February 2020).
52. Xie, Y. *Dynamic Documents with R and Knitr*, 2nd ed.; Chapman; Hall/CRC: Boca Raton, FL, USA, 2015. Available online: https://yihui.name/knitr/ (accessed on 3 February 2020).
53. Aust, F.; Barth, M. Papaja: Create APA Manuscripts with R Markdown. 2018. Available online: https://github.com/crsh/papaja (accessed on 3 February 2020).
54. Pedersen, T.L. *patchwork: The composer of ggplots*. 2017. Available online: https://github.com/thomasp85/patchwork (accessed on 3 February 2020).
55. Xie, Y.; Allaire, J.J.; Grolemund, G. *R Markdown: The Definitive Guide*; Chapman; Hall/CRC: Boca Raton, FL, USA, 2018. Available online: https://bookdown.org/yihui/rmarkdown (accessed on 3 February 2020).
56. Wickham, H.; Henry, L. Tidyr: Tidy Messy Data. 2019. Available online: https://CRAN.R-project.org/package=tidyr (accessed on 3 February 2020).