

Detecting misfitting items in Rasch models

Magnus Johansson

2025-01-03

Psychometrics in general have long relied on rule-of-thumb cutoff values for various goodness of fit metrics. With more powerful personal computers it is both feasible and desirable to use simulation/bootstrap methods to determine appropriate cutoff values. This paper illustrates and evaluates the use of an R package for Rasch psychometrics that has implemented functions to simplify the process of identifying misfitting items using simulation based cutoff values. Through a series of simulation studies a comparison is made between information weighted conditional item fit and item-restscore using Goodman and Kruskal's gamma.

1 Introduction

This paper presents a series of simulations conducted to evaluate methods to detect of item misfit in Rasch models. First, conditional item infit and outfit will be under scrutiny. Second, item infit will be compared to item-restscore (Kreiner 2011; Mueller and Santiago 2022). Third, a bootstrap method for item-restscore will be presented and tested.

Müller (2020) showed how the range of critical values for conditional item infit varies with sample size. The expected average item conditional infit range was described by Müller as fairly well captured by Smith's rule-of-thumb formula $1 \pm 2/\sqrt{n}$ (Smith, Schumacker, and Bush 1998). However, the average range does not apply for all items, since item location also affects model expected item fit which means that some items may have plausible item fit values outside Smith's average value range.

It is here proposed that by using parametric bootstrapping one can establish item fit critical cutoff values that are sample and item specific. The R package **easyRasch** (Johansson 2024) includes a function to determine item infit and outfit cutoff values, which will be tested in the simulations in this paper.

It is important to note that the conditional item fit described by Müller (2020) and implemented in the **iarm** R package (Mueller and Santiago 2022) should not be confused with the unconditional item fit implemented in software such as Winsteps and RUMM2030, as well as

all R packages except `iarm`. Unconditional item fit can result in unreliable item fit in sample sizes as small as 250 with increasing likelihood of problems as sample size increases. Readers are strongly recommended to read Müller’s paper to fully understand the problems with unconditional item fit.

2 Methods

Source: [Article Notebook](#)

A fully reproducible manuscript with R code and data is available

The simulation of response data used three steps: first, a vector of theta values (person scores on the latent variable) were generated using `rnorm(mean = 0, sd = 1.5)`. Second, a set of item locations ranging from -2 to 2 logits were generated for dichotomous items, using `runif(n = 20, min = -2, max = 2)`. Third, the theta values were used to simulate item responses for participants, using `sim.xdim()` from the `eRm` package (Mair and Hatzinger 2007), which allows simulation of multidimensional response data. Multiple datasets were generated using the same item and person parameters, varying the targeting and number of the misfitting item(s). More details are described under the separate studies below.

The parametric bootstrapping procedure was implemented using random samples from the 10 000 simulated responses. The sample size variations tested are described under each study.

1. Estimation of item locations based on simulated item responses using conditional maximum likelihood (CML, Mair and Hatzinger 2007).
2. Estimation of sample theta values using weighted maximum likelihood (Warm 1989)
3. Simulation of new response data, fitting the Rasch model, using the estimated item locations and theta values.
4. Estimation of the dichotomous Rasch model for the new response data using CML.
5. Based on step 4, calculation of conditional item fit (Müller 2020; Mueller and Santiago 2022) and/or item-restscore metrics (Kreiner 2011; Mueller and Santiago 2022).

Steps three and four were iterated over, using resampling with replacement from the estimated theta values as a basis for simulating the response data in step three.

Summary statistics were created with focus on the percentage of correct detection of misfit and false positives.

Source: [Article Notebook](#)

Source: [Article Notebook](#)

Source: [Article Notebook](#)

3 Study 1

Item mean square standardized residuals are either unweighted, which is referred to as “outfit”, or information weighted, which we call “infit” (Ostini and Nering 2006, 86–87). For details on conditional item fit we refer to the previously mentioned paper by Müller (2020).

Main lines of inquiry:

1. How does the number of iterations used in `RIgetfit()` impact the indicated cutoff values?
2. How useful are the cutoff values in detecting misfitting items (and false positives), when using the optimal number of iterations?
3. Müller (2020) hints at outfit being less useful than infit. We will investigate this by comparing them.

20 dichotomous items are used, with one item misfitting. Item locations are the same throughout. The location of the misfitting item relative to the sample theta mean was selected to be approximately 0, -1, and -2 logits. Three separate datasets were generated with these variations, each with 10 000 simulated respondents. One dataset with all three misfitting items was also generated, using the same sample size.

The function `RIgetfit()` from the `easyRasch` R package is tested here. It’s source code can be accessed on GitHub. The function offers the user a choice of the number of bootstrap iterations to use to calculate the cutoff values for infit and outfit.

Then the `RIitemfit()` function is used to summarize the bootstrap results and also calculates the infit and outfit for each item in the observed data and highlights items with infit/outfit values outside of the cutoff values. `RIitemfit()` has a default (user modifiable) setting to slightly truncate the distribution of values using `stats::quantile()` at 0.001 and 0.999 to remove extreme values. An example is demonstrated in Table 1, using a subset of the items used in the simulations. Figure 1 provides a visualization of the distribution of bootstrapped infit and outfit values, together with the infit/outfit values from the observed data illustrated using an orange diamond shape. Note the variation between items in plausible values of infit and outfit based on the bootstrap, and that Smith’s rule-of-thumb regarding infit ($1 \pm 2/\sqrt{n}$) would be 0.9-1.1 for a sample size of 400.

This study was rather computationally demanding since each simulation run entailed 100-400 underlying simulations. The sample sizes used were 150, 250, 500, and 1000. The number of iterations to determine cutoff values were 100, 200, and 400. Sample size and iteration conditions were fully crossed with each other and the three different targeting variations of the one misfitting item, resulting in $433 = 36$ conditions. Each combination used 200 simulation runs. The simulations took about 12 hours to run on a Macbook Pro Max M1 using 9 CPU cores.

Table 1: Conditional item fit with simulation based cutoff values

Item	Infit		Outfit		Infit diff	Outfit diff	Location
	InfitMSQ	thresholds	OutfitMSQ	thresholds			
V1	1.017	[0.862, 1.162]	1.061	[0.647, 1.423]	no misfit	no misfit	-1.37
V11	1.000	[0.839, 1.21]	1.032	[0.742, 1.289]	no misfit	no misfit	-0.66
V3	1.022	[0.902, 1.098]	1.050	[0.662, 1.498]	no misfit	no misfit	0.46
V12	0.966	[0.827, 1.19]	0.793	[0.783, 1.306]	no misfit	no misfit	1.58

Source: [Article Notebook](#)

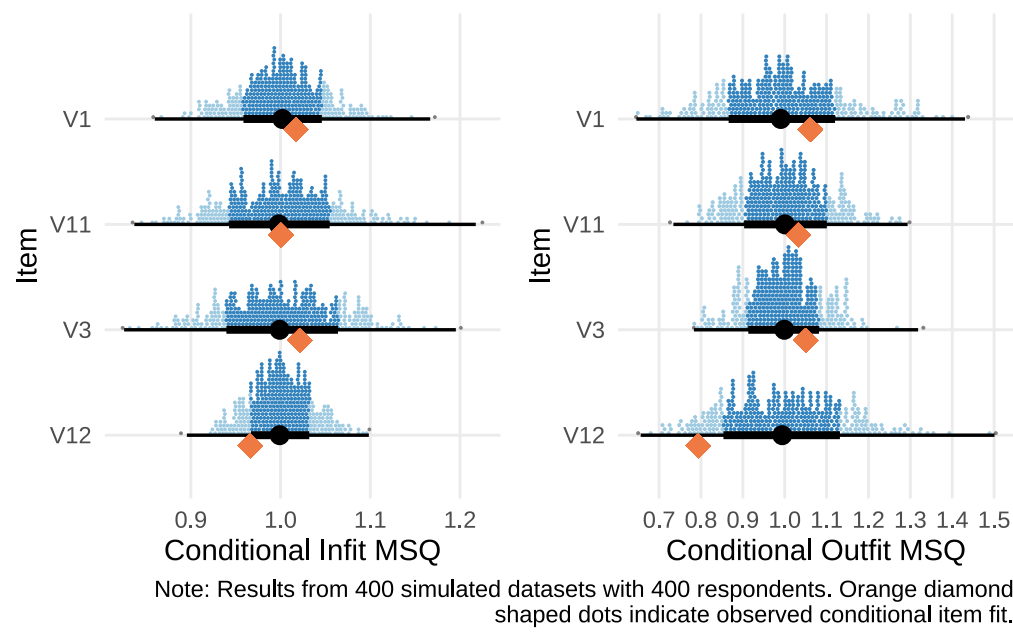


Figure 1: Distribution of simulation based item fit and estimated item fit from observed data

Source: [Article Notebook](#)

Source: [Article Notebook](#)

Source: [Article Notebook](#)

3.1 Results

Source: [Article Notebook](#)

Figures show the percent of simulation runs that have identified an item as misfitting. Items with more than 5% are colored in light red. A number representing the detection rate is shown adjacent to the bar representing the misfitting item. The figure grid columns are labelled with the number of iterations used by `Rlgetfit()` to determine cutoff values, and grid rows are labelled with the sample size.

3.1.1 Infit

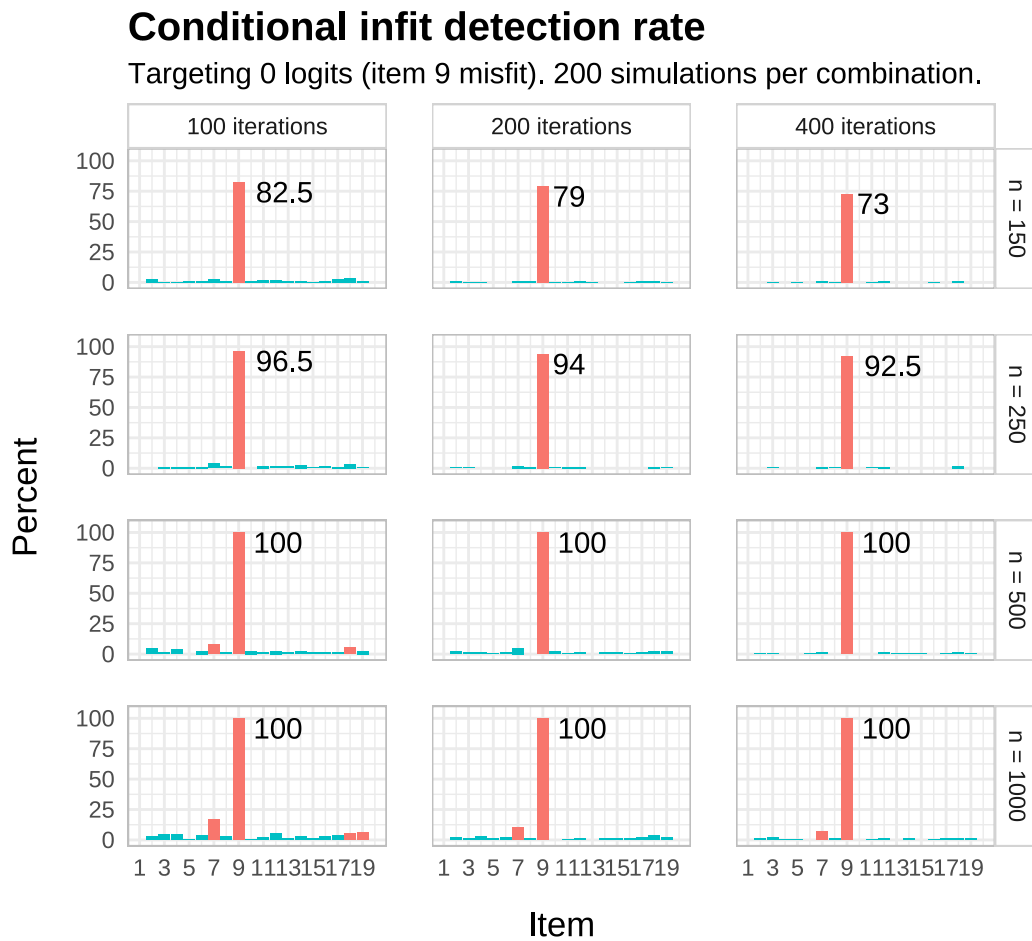


Figure 2

Source: [Article Notebook](#)

Figure 2 shows the detection rate when the misfitting item is located at the sample mean. Detection rate is highest for the condition with 100 iterations with sample size 100 and 250, but it also shows higher levels of false positives when sample size increases to 500 or more.

Conditional infit detection rate

Targeting -1 logits (item 18 misfit). 200 simulations per combination.

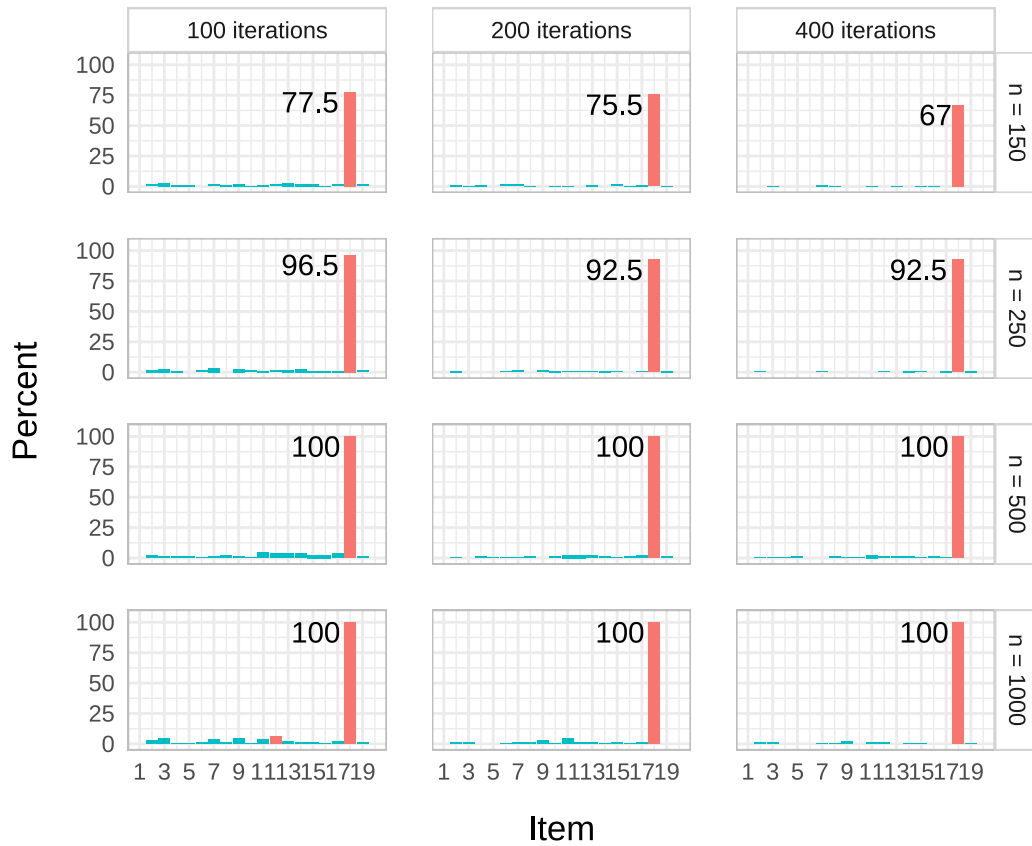


Figure 3

Source: [Article Notebook](#)

When the misfitting item is offset in targeting by -1 logits compared to the sample mean (see Figure 3), the smallest sample size has less power to detect misfit compared to the on-target misfitting item. There are lower rates of false positives across all sample sizes and iterations.

Conditional infit detection rate

Targeting -2 logits (item 13 misfit). 200 simulations per combination.



Figure 4

Source: [Article Notebook](#)

Finally, when the misfitting item is located at -2 logits compared to the sample mean (see Figure 4), we see a stronger reduction in power for sample sizes 150 and 250. No false positives are identified.

3.1.2 Outfit

Conditional outfit detection rate

Targeting 0 logits (item 9 misfit), 200 simulations per combination.

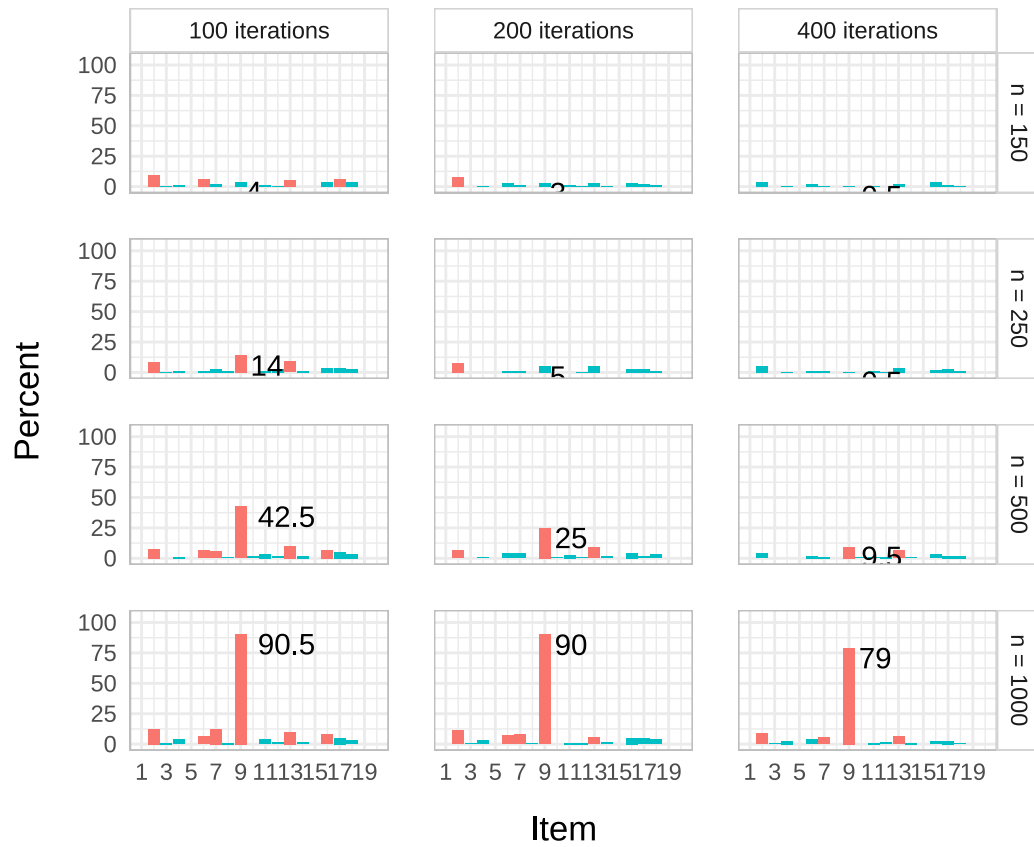


Figure 5

Source: [Article Notebook](#)

Conditional outfit detection rate

Targeting -1 logits (item 18 misfit). 200 simulations per combination.

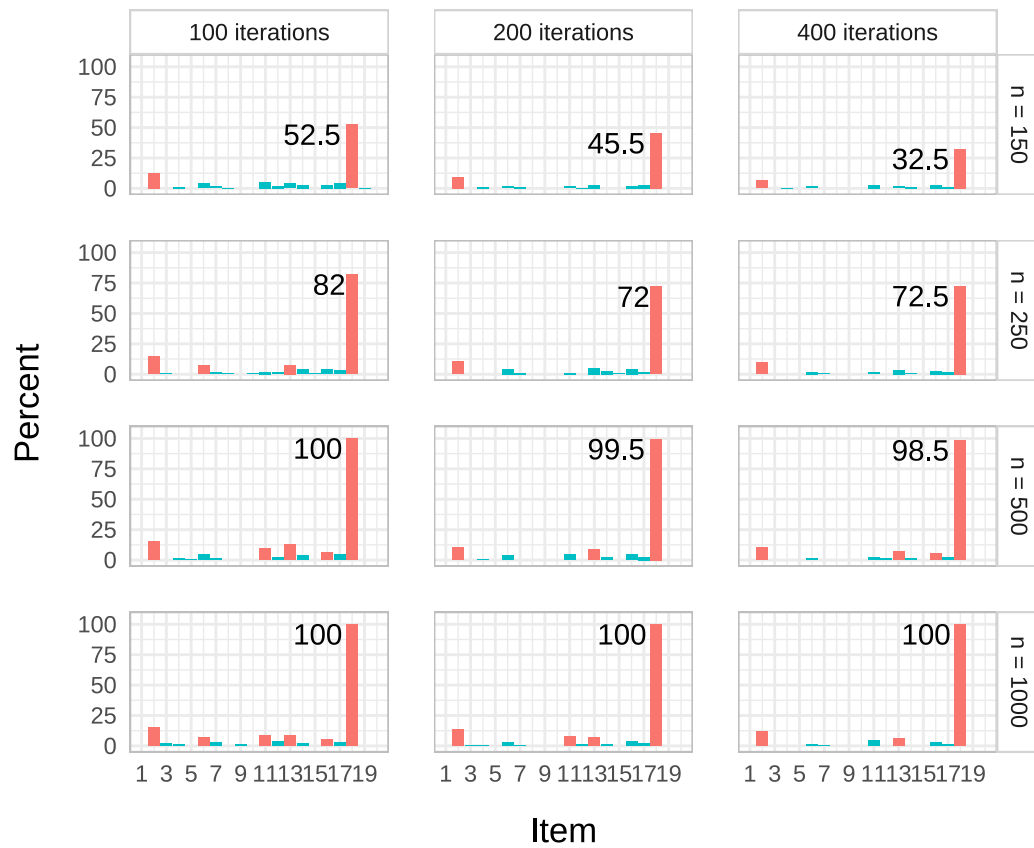


Figure 6

Source: [Article Notebook](#)

Conditional outfit detection rate

Targeting -2 logits (item 13 misfit). 200 simulations per combination.

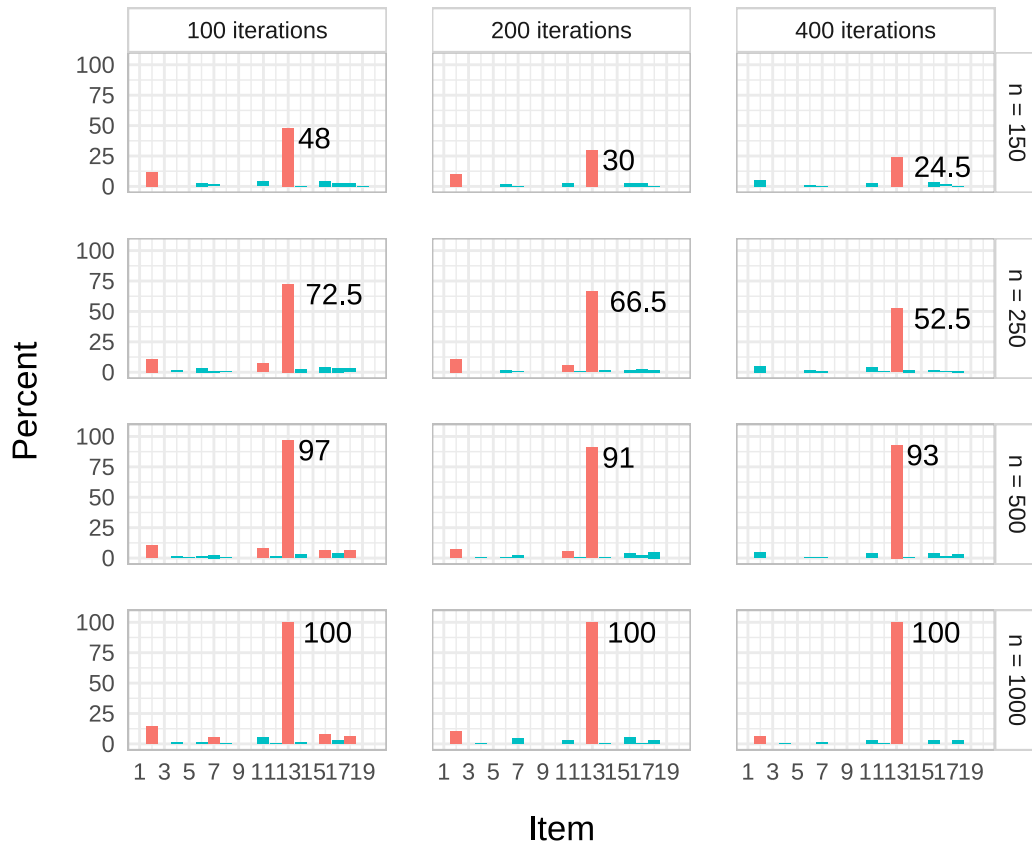


Figure 7

Source: [Article Notebook](#)

As shown in Figure 5, Figure 6, and Figure 7, outfit is performing much worse than infit across the board.

3.1.3 Comments

Based on these simulation, it seems reasonable to recommend the use of infit in determining item fit over outfit. The performance of outfit calls to question whether it is useful at all.

Regarding infit and the use of parametric bootstrapping with `Rlgetfit()`, it looks like 100 iterations are to recommend to determine cutoff values when the sample size is 250 or lower, while 200 or 400 iterations reduce the risk for false positives at sample sizes of 500 or larger.

False positives are found at sample sizes 500 and 1000 only. The risk for false positives is notably higher when the misfitting item is located at the sample mean compared to when the misfitting item is off-target by -1 logits or more.

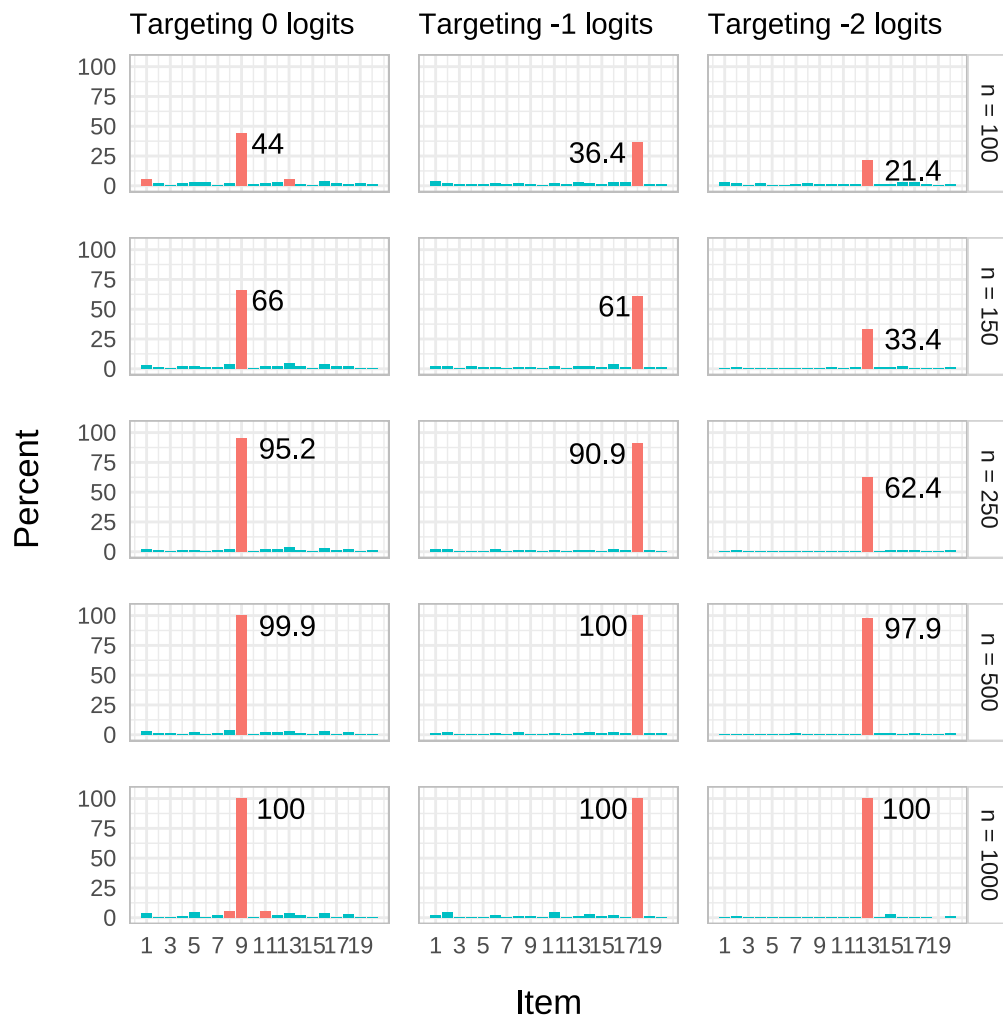
4 Study 2

Source: [Article Notebook](#)

Source: [Article Notebook](#)

4.1 Results

Item-restscore detection rate across targeting and sample size



Note: 1000 simulated datasets for each combination.

Figure 8

Source: [Article Notebook](#)

This simulation includes an additional condition with 100 respondents, which results in significantly lower detection rates than with 150 respondents. Compared to infit at 250 respondents, item-restscore has detection rates of 95.2%, 90.9%, and 62.4% for targeting 0, -1, and -2,

while infit has 96.5%, 96.5%, and 71%. For sample size 500 and 1000, performance is similar, including the increased tendency for false positives at $n = 1000$.

Similarly to infit, item-restscore has decreased detection rate for off-target misfitting items. The false positive rate is lower for item-restscore than infit for sample sizes below 1000.

5 Study 3

We will now compare the performance of infit and item-restscore when all three items are misfitting at the same time. This simulation will also include a condition with 2000 respondents, to examine if the false positive rate increases with more respondents. For infit, we will only use 200 iterations with `Rlgetfit()` since that condition seemed to strike a balance between detection rate and false positives.

Source: [Article Notebook](#)

Source: [Article Notebook](#)

Source: [Article Notebook](#)

5.0.1 Results

Conditional infit detection rate

3 items misfitting. 500 simulations per combination.

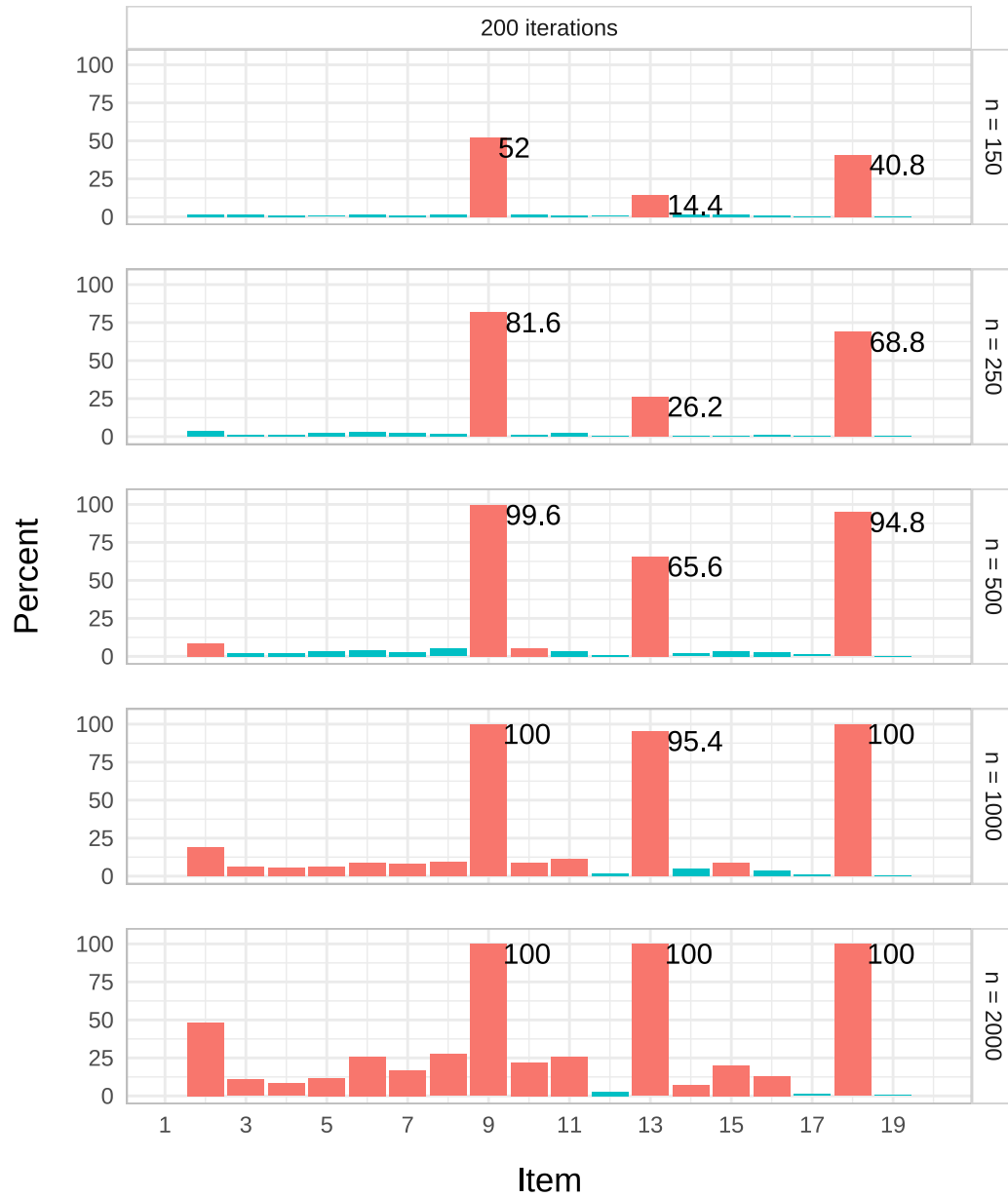


Figure 9

Conditional outfit detection rate

3 items misfitting. 500 simulations per combination.



Figure 10

Source: [Article Notebook](#)

Looking at the performance of infit with three misfitting items (Figure 9), we can see that the detection rate is markedly worse for item 13 in sample sizes 500 and below, compared to when single items were misfitting. The false positive rate has increased for sample size of 1000, and we can see this escalate strongly for $n = 2000$. Outfit (Figure 10) again performs worse than infit.

Item-restscore detection rate

3 items misfitting. 500 simulations per combination.



Figure 11

Source: [Article Notebook](#)

Item-restscore has higher detection rate than infit (see Figure 11), but also higher levels of

false positives.

6 Study 4

For our final set of simulations, we will use a non-parametric bootstrap procedure with item-restscore. First, based on the above problematic sample size of 2000, we let the bootstrap function sample with replacement using $n = 800$. 250 bootstrap samples will be used, then we summarize the percentage indicating misfit for each item. Second, we will also apply this to samplesizes 150, 250, and 500, and use the complete sample for the same bootstrap procedure to see if this produces more useful information for identifying misfitting items.

7 Discussion

8 Conclusion

These findings make a good argument for removing one item at a time when the analysis indicates misfitting items, starting with the most misfitting item. This is especially relevant for sample sizes at 500 or above and when misfitting items are located close to the sample mean.

References

- Johansson, Magnus. 2024. *easyRasch: Psychometric Analysis in r with Rasch Measurement Theory*. <https://github.com/pgmj/easyRasch>.
- Kreiner, Svend. 2011. “A Note on Item–Restscore Association in Rasch Models.” *Applied Psychological Measurement* 35 (7): 557–61. <https://doi.org/10.1177/0146621611410227>.
- Mair, Patrick, and Reinhold Hatzinger. 2007. “Extended Rasch Modeling: The eRm Package for the Application of IRT Models in R.” *Journal of Statistical Software* 20 (1): 1–20. <https://doi.org/10.18637/jss.v020.i09>.
- Mueller, Marianne, and Pedro Henrique Ribeiro Santiago. 2022. “Iarm: Item Analysis in Rasch Models.” <https://cran.r-project.org/web/packages/iarm/index.html>.
- Müller, Marianne. 2020. “Item Fit Statistics for Rasch Analysis: Can We Trust Them?” *Journal of Statistical Distributions and Applications* 7 (1): 5. <https://doi.org/10.1186/s40488-020-00108-7>.
- Ostini, Remo, and Michael Nering. 2006. *Polytomous Item Response Theory Models*. SAGE Publications, Inc. <https://doi.org/10.4135/9781412985413>.
- Smith, R. M., R. E. Schumacker, and M. J. Bush. 1998. “Using Item Mean Squares to Evaluate Fit to the Rasch Model.” *Journal of Outcome Measurement* 2 (1): 66–78.

Warm, Thomas A. 1989. “Weighted Likelihood Estimation of Ability in Item Response Theory.” *Psychometrika* 54 (3): 427–50. <https://doi.org/10.1007/BF02294627>.