Bridging the Grade Gap: Reducing Assessment Bias in a Multi-Grader Class

Manuscript ID: PA-2020-174.R1
Replication ReadMe File

August 16, 2022

This README file covers the contents of the replication materials and how to replicate each Figure and Analysis included in the paper. Running replication code (e.g. generating figures) took 20 minutes to implement on an Intel(R) Core(TM) i5-4590S CPU processor with 16.0 GB RAM. The main analysis and simulation analysis depend upon a large number of models, each of which took 1-7 days to run on the high performance clusters at New York University (Operating System: RedHat Enterprise Linux 8.2). Each of these scripts were implemented using R version 4.0.3 and the following specific packages:

- rstan 2.21.2
- tidyverse 1.3.1
- Rcpp 1.0.5
- ggplot2 3.3.3
- gridExtra 2.3
- ggpubr 0.4.0
- parallel 4.0.3
- coda 0.19-4
- plyr 1.8.5

In addition, it is recommended that those interested in replicating the modeling part of the code below increase the maximum number of DLLs that R allows. There are multiple ways of doing this, but including "R_MAX_NUM_DLLS=256" in the .Renviron file.

There are two main folders in the replication materials, named "Actual Data Analysis" and "Simulation Analysis." These are, broadly:

1. **Actual_Data_Analysis:** this subfolder contains scripts to conduct analysis on the grades produced from the real course taught and overseen by the authors. It also contains this data, the necessary model to conduct the analysis, and the scripts to produce the figures in the paper and appendices. Finally, it contains a subfolder with the results of the analyses in the form necessary to replicate the figures. Each script for analysis explains which Figure it aids

creation, and each visualization script notes which analysis scripts are necessary to produce the results that feed the figure.

2. **Simulation_Analysis** - this subfolder contains scripts to conduct analysis based on simulations that is contained in the paper. It also contains the necessary model to conduct the analysis, and the scripts to produce the figures in the paper and appendices. Finally, it contains two (2) subfolders with the results of the analyses in the form necessary to replicate various figures. Each script for analysis explains which Figure it aids creation, and each visualization script notes which analysis scripts are necessary to produce the results that feed the figure.

We describe the full contents of these two subfolders below.

1 Actual_Data_Analysis

1. **MidtermAnalysis.R, Fig3.R, FigA1.R and FigA2.R**: Implements the Bayesian Aldrich Mckelvey algorithm on the midterm assignment. The figure scripts produce figures 3 (MAE) in the manuscript and A1 (RMSE) and A2 (MAE of scores) in Appendix B.

2. **PaperAnalysis.R and FigA3.R**: Implements the Bayesian Aldrich Mckelvey algorithm on the paper assignment. The figure script produces Fig A3 in Appendix B.

3. **FinalGradeAnalysis.R and FigA4A5.R**: Implements the Bayesian Aldrich Mckelvey algorithm on the the final grade assignment. The figure script produces Figs A4 and A5 in Appendix B.

4. **FinalExamAnalysis.R and FigA6.R**: Implements the Bayesian Aldrich Mckelvey algorithm on the the final exam assignment. Produces Fig A4 and A5 in Appendix B.

5. **Analysis2TA.R and FigA7.R**: Analysis of improvements from bridging for each of the pairwise combinations of the teaching assistants (TAs). Implements the Bayesian Aldrich Mckelvey algorithm after removing one TA at a time. FigA7.R produces Fig A7 in Appendix B.

6. **ExtremeBridgeAnalysis.R and FigA8.R**: Analysis on whether bridge extremity matters. Implements the Bayesian Aldrich Mckelvey algorithm across different types of bridges (a) 1-10 exams from the lowest third of grades (b) 1-10 exams from the highest third of grades and (c) 1-5 exams from both the highest and lowest extremes. FigA8.R produces Fig A8 in Appendix B.

2 Simulation_Analysis

2.1 Varying Parameters

1. **simulation_code_hpc.R**: runs twenty iterations of the main simulation procedure. The procedure generates 27 data sets reflecting various degrees of grader reliability (β) and grader shift (α) parameters, as well as grade-level error (μ). In the second section, we randomly select

5 bridging observations, and implement the Bayesian Aldrich Mckelvey algorithm. Finally, in lieu of storing all the posterior estimates, we store the MAE and RMSE of estimates of the latent trait as both levels and ranks.

2. **simulation_dataset_rank_BAM.RDA** is a dataset that summarizes the MAE and RMSE (on the rank scale) of the latent trait for each iteration of the simulation procedure.
   - *alpha*: Level of variability in the intercept or shift-term - no variability, $\mathcal{N}(0,3)$, or $\mathcal{N}(0,7)$
   - *beta*: Level of variability in the slope-term - no variability, $U(0.5,1)$, or $U(0.8,1)$
   - *tau*: Level of variability in the stochastic error term - no variability, $\mathcal{N}(0,3)$, or $\mathcal{N}(0,7)$
   - *metric*: Whether value column refers to MAE or RMSE
   - *value*: MAE or RMSE of the bridged grade
   - *unbridged_value*: MAE or RMSE of the assigned grade
   - *runs*: Iteration number
   - *Convergence*: Dummy indicator equal to one if the latent trait parameters converged.
3. **figuresA9A10.R** provides code to generate Appendix Figures A9 and A10.

2.2 Comparison Over Size of Classes and Numbers of Graders

1. **ConditionSimulationAnalysis.R**: Analysis on how effectiveness is conditioned by number of students, number of graders, and type of variability. Constructs simulated dataset and implements the Bayesian Aldrich Mckelvey algorithm across all combinations of (a) 12, 30 or 60 classroom sizes (b) two, three, four or five graders and (c) low or high grader variability.
2. **FigA11.R**: produces Fig A8 in Appendix B.

2.3 Comparison with IRT models (Appendix Figure 15)

1. **create_framework_datasets_IRT.R** generates framework datasets based upon the data structure of the midterm and assuming various forms and degrees of DIF and precision. This script replicates the approach used by Pemstein et al. (2015) and has three sections
   (a) First, we load the **bridgereplicdf.rds** dataset and generate the simulation parameters (levels of coder precision and DIF).
   (b) The second section details three different functions for ordinalizing perceptions of the latent ability (each function reflects a different degree of variability in DIF).
   (c) In the third section, we apply these functions to datasets generated under various degrees of variability in coder precision.
2. **datairt** is a folder that contains the 9 framework datasets generated in Step 1:
   (a) 'cons.rel_no.dif' assumes no variability in DIF and constant precision
   (b) 'cons.rel_low.threshold.dif.' assumes medium variability in DIF ($\kappa_{i,k} \sim \mathcal{N}(\kappa_k, 3)$) and constant precision

(c) 'cons.rel_high.threshold.dif.' assumes high variability in DIF ($\kappa_{i,k} \sim \mathcal{N}(\kappa_k, 7)$) and constant precision

(d) 'med.sd.rel_no.dif' assumes no variability in DIF and medium variability in precision ($\tau_i \sim \mathcal{N}(1, .5)$)

(e) 'med.sd.rel_low.dif' assumes medium variability in DIF and medium variability in precision

(f) 'med.sd.rel_high.dif' assumes high variability in DIF and medium variability in precision

(g) 'high.sd.rel_no.dif' assumes no variability in DIF and high variability in precision ($\tau_i \sim \mathcal{N}(1, 1)$)

(h) 'high.sd.rel_low.dif' assumes medium variability in DIF and high variability in precision

(i) 'high.sd.rel_high.dif' assumes high variability in DIF and high variability in precision

3. **simulation_code_hpc_IRT.R** runs one iteration of the simulation procedure described in Appendix C, page 44 (we ran this file 20 times). The procedure randomly selects bridging observations and implements

   (a) "unbridged" simply uses the assigned grader's input i.e., does not adjust for DIF.

   (b) "bam" is the Bayesian Aldrich McKelvey algorithm.

   (c) "irt2" is the IRT model with threshold DIF parameterization.

4. **resultsirt** is a folder that contains results from 20 iterations of the simulation procedure in Step 3. Specifically, each dataset contains nine columns:

   – *dgp*: Form and magnitude of the simulated DIF and precision (nine-levels listed in Step 2).

   – *model*: Model used to correct for bias (three-levels listed in Step 3)

   – *rmse*: Root mean-squared error of the output grades

   – *rmse_rank*: Root mean-squared error of the output ranks

   – *mae*: Mean absolute error of the outputted grades

   – *mae_rank*: Mean absolute error of the outputted ranks

   – *Convergence*: Dummy indicator equal to one if the latent trait parameters converged.

5. **figureA12_IRT.R** provides code to combine simulation results and generate the nine figures in Appendix Figure A12.