Replication Material for

Bayesian Rule Set: A Quantitative Alternative to Qualitative Comparative Analysis

Albert Chiu and Yiqing Xu*

Please feel free to contact the authors for further assistance.

1. Replication Instructions

To replicate the entirety of our analysis, please follow the following procedure:

1. Specify the path to the replication files `rep_path` in the file `preamble.r`. Optionally, set the number of cores `numCores` (default is 4).

2. Run `1_preamble.r` — this file will install `BRS` from source (provided with the replication materials) and load helper functions

3. Run `2_lipset_script.r` — this file will run BRS on the democracy example and create the corresponding figures and tables in the paper (including Table 2 and Table A4, Figure 1 and Figure 2)

    - Note: By default, the script will not run BRS. We provide the resulting rule sets in `lipset_out_pois.rda`. To reproduce the rule sets, simply run the part of Part II in `lipset_script.r` that is surrounded by `if (FALSE)` {...}. This will a few minutes.

4. Run `3_vote_script.r` — this file will run BRS on the voting example and create the corresponding figures (including Figure 4, Figure 5, Figure A2 and Figure A3). It will also run LASSO on the voting example, which has no corresponding figures or tables, but can be used to verify our analysis in the appendix.

*Albert Chiu, Graduate Student, Department of Political Science, Stanford University. Email: altchiu@stanford.edu. Yiqing Xu, Assistant Professor, Department of Political Science, Stanford University. Email: yiqingxu@stanford.edu.

- Note: By default, the script will not run BRS. We provide the resulting rule sets in `vote_out_pois.rda`. To reproduce the rule sets, simply run the part of Part II in `vote_script.r` that is surrounded by `if (FALSE) {...}`. This will not take as long as the simulations to run, but will still take a nontrivial amount of time.

5. Run `4_sim_script.r` — this file will run the simulations and create the corresponding figures and tables that appear in the paper (including Figure 3, Figure A2, Table A2 and Table A3).

    - Note: By default, the script will not run the simulations, which are computationally expensive to conduct. We provide the simulated data in `simData.rda` and `simIndices.rda`, the simulation results from BRS and QCA respectively in `sim_out_combined.rda` and `sim_out_QCA.rda`, and the statistics (average parsimony and out of sample accuracy) in `performance.rda`. The figures and tables are based on `performance.rda` only.
    - To reproduce these rule sets and calculate their statistics yourself, simply run the section in Part II of `sim_script.r` that is surrounded by an `if (FALSE) {...}`. The BRS simulations are broken into three different files so that you may run them on separate nodes/computers. For reference, they take five hours (one file takes one hour, the other two take two hours) to run sequentially on a single computer with 20 cores.

Notes:

- The resulting figures will be created in the `figures` folder, and tables in the `tables` folder. Intermediary data will be stored in `sim/out`, `lipset/out`, and `vote/out` depending on whether it is from the simulations, the democracy example, or the voting example.
- Explanations of functions used can be found in the documentation for the `BRS` package (for the functions `BRS` and `agg_BRS`) or in comments in the `helper.r` file (all other functions we define).
- Table 1 and Table A1 are a toy example, which does not require replication.
- We did not include a log file because all estimation, simulations, and graphics production work behind the scene.

2. Description of Contents

2.1. Program Files

`BRS_0.0.0.9006.tar.gz`: This is a developmental version of the authors' R package for running BRS. Please install it in R. The source files include code written in Python 3.0 (modified from Wang et al (2016, 2017)) and R for running the BRS algorithm and creating figures. We use the `R` package `reticulate` to run the Python code in R. The supplied replication code (in particular, lines 16-19 and 51-52 of `1_preamble.R`) will create a conda environment with the necessary packages, and the user does not have to separately install Python or any Python packages.

BRS depends on the following R packages:

- `reticulate`
- `circlize`
- `Rtsne`
- `ggplot2`
- `dplyr`
- `magrittr`
- `gtools`

BRS also depends on the following Python packages:

- `scipy`
- `numpy`
- `pandas`
- `scikit-learn`

2.2. General Replication Files

1. `1_preamble.r`: Sets the working directory, specifies the number of cores to use for parallelized tasks, installs packages (including `BRS`), loads helper functions, and creates a conda environment for running BRS. `BRS` is installed from the supplied source files, and the package `ggpattern` is installed from GitHub.

2. `helper.r`: Helper functions for creating graphs and computing statistics of the simulated rule sets.

2.3. Replication Files for Democracy Empirical Example

Code

1. `2_lipset_script.r`: Runs all files (listed below) needed to replicate the first empirical example (democracy in interwar Europe).

2. `lipset_make-data.r`: Loads data and creates all labels and other objects required for replication. The data is included with the `BRS` package.

3. `lipset_pois.r`: Runs BRS on example.

4. `lipset_bar.r`: Makes Figure 1 in paper.

5. `lipset_chord.r`: Makes Figure 2(a) in paper. NOTE: labels (e.g. "high") were added manually using image editing software.

6. `lipset_tsne.r`: Makes Figure 2(b) in paper. NOTE: lines encircling points were added manually using image editing software. Code is provided for identifying the observations covered by each rule.

7. `lipset_tt.r`: Makes Table A.4.1 in paper (Table 2 is the first five rows of this table).

Data

1. `lipset_out_pois.rda`: Rule sets produced by 100 bootstraps of BRS on the data for the democracy example.

2.4. Replication Files for Voter Turnout Empirical Example

Code

1. `3_vote_script.r`: Runs all files (listed below) needed to replicate the second empirical example (voter turnout in the US). By default, does not run BRS. Instead, the resulting rule sets are provided. Can reproduce these rule sets by running the code in Part II contained within the `if (FALSE)` {...} statement.

2. `vote_make-data.r`: Loads data and creates all labels and other objects necessary for second empirical example. Data is downloaded from Harvard Dataverse.

3. `vote_pois.r`: Runs BRS on example.

4. `vote_bar.r`: Makes Figure 4 in paper.

5. `vote_chord.r`: Makes Figure 5(a) in paper. NOTE: labels (e.g. "high") were added manually using image editing software.

6. `vote_tsne.r`: Makes Figure 5(b) in paper. NOTE: dashed lines encircling points were added manually using image editing software. Code is provided for identifying the observations covered by each rule.

7. `vote_trees.r`: Makes Figures A2 and A3 in paper.

8. `vote_lasso.r`: Replicates analysis described in Section A.2.2.

Data

1. `vote_out_pois.rda`: Rule sets produced by 100 bootstraps of BRS on the data for the voter turnout example.

2.5. Replication Files for Simulations

Code

1. `4_sim_script.r`: Runs all files (listed below) needed to replicate simulations. By default, does not run simulations. Instead, the resulting rule sets are provided. Can reproduce these rule sets by running the code in Part II contained within the `if (FALSE)` {...} statement.

2. `sim_make-data.r`: Generates simulated populations and the indices of the samples from them.

3. `sim_pois_param.r`: Parameters for BRS for simulations

4. `sim_pois1-3.5.r`: Runs BRS on all deterministic settings and the probabilistic setting with 5 variables.

5. `sim_pois4.r`: Runs BRS on probabilistic setting with 10 variables.

6. `sim_pois6.r`: Runs BRS on probabilistic setting with 20 variables.

7. `sim_combine.r`: Combines output of `sim_pois1-3.5.r`, `sim_pois4.r`, and `sim_pois6.r`.

8. `sim_QCA.r`: Runs QCA on all settings, except for 20 variables and $N \geq 250$.

9. `sim_stats.r`: Computes average number of rules, length of rules, and out of sample accuracy for all BRS and QCA solutions.

10. `sim_plot.r`: Creates Figures 3 and A1 in the paper, which plots the statistics computed in `sim_stats.r`.

11. `sim_table.r`: Creates Tables A2 and A3 in the paper, which are tables of the statistic computed in `sim_stats.r`.

Data

1. `simData.rda`: Simulated populations.

2. `simIndices.rda`: Indices of samples drawn from simulated populations.

3. `sim_out_combined.rda`: Rule sets produced by BRS on simulated data.

4. `sim_out_QCA.rda`: Rule sets produced by QCA on simulated data, except for 20 variables and $N \geq 250$.

A. Versioning

A.1. BRS

BRS (for both the simulations and the empirical examples) were conducted on Sherlock, the high performance computing cluster at Stanford. This includes the following files:

- `sim_pois_param.r`
- `sim_pois1-3.5.r`
- `sim_pois4.r`
- `sim_pois6.r`
- `lipset_pois.r`
- `vote_pois.r`

This was done using version 4.0.2 and the following versions of R packages:

- `doParallel_1.0.16`
- `iterators_1.0.13`
- `foreach_1.5.1`
- `BRS_0.0.0.9006`
- `compiler_4.0.2`
- `magrittr_2.0.1`
- `codetools_0.2-16`

This was also done using Python 3.6.1 and the following versions of Python packages:

- `scipy/1.4.1`
- `numpy/1.19.2`
- `pandas/1.0.3`
- `scikit-learn/0.24.2`

A.2. All Other Files

All other files were run on the authors' personal computer. This was done using R version 4.0.3 and the following versions of R packages:

- gridExtra 2.3
- gtools 3.9.2
- glmnet 4.1-2
- Matrix 1.3-4
- randomForest 4.6-14
- rpart.plot 3.1.0
- rpart 4.1-15
- kableExtra 1.3.4
- Rtsne 0.15
- prettyGraphs 2.1.6
- RColorBrewer 1.1-2
- circlize 0.4.13
- cowplot 1.1.1
- ggplot2 3.3.5
- QCA 3.13
- admisc 0.16
- BRS 0.0.0.9006
- dplyr 1.0.7
- doParallel 1.0.16
- iterators 1.0.13
- foreach 1.5.1
- lattice 0.20-44
- svglite 2.0.0
- assertthat 0.2.1

- digest 0.6.27
- utf8 1.2.1
- mime 0.11
- R6 2.5.0
- evaluate 0.14
- httr 1.4.2
- pillar 1.6.1
- GlobalOptions 0.1.2
- rlang 0.4.11
- rstudioapi 0.13
- rmarkdown 2.9
- splines 4.0.3
- webshot 0.5.2
- stringr 1.4.0
- munsell 0.5.0
- shiny 1.6.0
- compiler 4.0.3
- httpuv 1.6.1
- xfun 0.24
- pkgconfig 2.0.3
- systemfonts 1.0.2
- shape 1.4.6
- htmltools 0.5.1.1
- tidyselect 1.1.1
- tibble 3.1.2
- venn 1.10
- codetools 0.2-18

- fansi 0.5.0
- viridisLite 0.4.0
- crayon 1.4.1
- withr 2.4.2
- later 1.2.0
- xtable 1.8-4
- gtable 0.3.0
- lifecycle 1.0.0
- DBI 1.1.1
- magrittr 2.0.1
- scales 1.1.1
- stringi 1.6.2
- promises 1.2.0.1
- xml2 1.3.2
- ellipsis 0.3.2
- generics 0.1.0
- vctrs 0.3.8
- tools 4.0.3
- glue 1.4.2
- purrr 0.3.4
- survival 3.2-11
- fastmap 1.1.0
- colorspace 2.0-2
- rvest 1.0.0
- knitr 1.33
- Rcpp 1.0.7
- ggpattern 0.3.2-14