Package: rrum (via r-universe)

November 2, 2026

Type Package

Title Bayesian Estimation of the Reduced Reparameterized Unified Model with Gibbs Sampling

Version 0.2.2

Description Implementation of Gibbs sampling algorithm for Bayesian Estimation of the Reduced Reparameterized Unified Model ('rrum'), described by Culpepper and Hudson (2017) <doi:10.1177/0146621617707511>.

URL https://tmsalab.github.io/rrum/, https://github.com/tmsalab/rrum

BugReports https://github.com/tmsalab/rrum/issues

License GPL (>= 2)

Depends R (>= 4.3.0), simcdm (>= 0.1.0)

Imports Rcpp (>= 1.1.0)

LinkingTo Rcpp, RcppArmadillo (>= 15.0.2-2), rgen, simcdm

Suggests testthat, covr

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation yes

Author Steven Andrew Culpepper [aut, cph] (ORCID: <https://orcid.org/0000-0003-4226-6176>), Aaron Hudson [aut, cph] (ORCID: <https://orcid.org/0000-0002-9731-2224>), James Joseph Balamuta [aut, cph, cre] (ORCID: <https://orcid.org/0000-0003-2826-8458>)

Maintainer James Joseph Balamuta <balamut2@illinois.edu>

Repository https://cran.r-universe.dev

Date/Publication 2025-09-27 04:30:02 UTC

RemoteUrl https://github.com/cran/rrum

RemoteRef HEAD

RemoteSha 0181e17b43aa36e7b5e979adf05d0741aac1ca39

Contents

rrum 2

Index **5**

`rrum` *Gibbs sampler to estimate the rRUM*

Description

Obtains samples from posterior distributon for the reduced Reparametrized Unified Model (rRUM).

Usage

```
rrum(
  Y,
  Q,
  chain_length = 10000L,
  as = 1,
  bs = 1,
  ag = 1,
  bg = 1,
  delta0 = rep(1, 2^ncol(Q))
)
```

Arguments

`Y`	A `matrix` with N rows and J columns, where N reperesnts the number of individuals and J the number of items. `Y` indicates the indviduals' responses to each of the items.
`Q`	A `matrix` with J rows and K columns indicating which attributes are required to answer each of the items. An entry of 1 indicates attribute k is required to answer item j. An entry of one indicates attribute k is not required.
`chain_length`	A `numeric` indicating the number of iterations of Gibbs sampler to be run. Default is set to 10000.
`as`	A `numeric`, parameter for the prior distribution of pistar. High values as encourage higher values of pistar and lower values of rstar.
`bs`	A `numeric`, parameter for the prior distribution of pistar. High values as encourage lower values of pistar and higher values of rstar.
`ag`	A `numeric`, parameter for the prior distribution of rstar. High values as encourage higher values of rstar.
`bg`	A `numeric`, parameter for the prior distribution of pistar. High values as encourage lower values of rstar.
`delta0`	A `vector`, parameters for the Dirichlet prior on pi.

Value

A `list` that contains

- `PISTAR`: A `matrix` where each column represents one draw from the posterior distribution of pistar.
- `RSTAR`: A $JxKxchain_length$ `array` where `J` reperesents the number of items, and `K` represents the number of attributes. Each slice represents one draw from the posterior distribution of `rstar`.
- `PI`: A `matrix` where each column reperesents one draw from the posterior distribution of `pi`.
- `ALPHA`: An $NxKxchain_length$ `array` where `N` reperesents the number of individuals, and `K` represents the number of attributes. Each slice represents one draw from the posterior distribution of `alpha`.

Author(s)

Steven Andrew Culpepper, Aaron Hudson, and James Joseph Balamuta

References

Culpepper, S. A. & Hudson, A. (In Press). An improved strategy for Bayesian estimation of the reduced reparameterized unified model. Applied Psychological Measurement.

Hudson, A., Culpepper, S. A., & Douglas, J. (2016, July). Bayesian estimation of the generalized NIDA model with Gibbs sampling. Paper presented at the annual International Meeting of the Psychometric Society, Asheville, North Carolina.

See Also

`simcdm::sim_rrum_items()`

Examples

```
# Set seed for reproducibility
set.seed(217)

## Define Simulation Parameters

N = 1000 # Number of Individuals
J = 6    # Number of Items
K = 2    # Number of Attributes

# Matrix where rows represent attribute classes
As = attribute_classes(K)

# Latent Class probabilities
pis = c(.1, .2, .3, .4)

# Q Matrix
Q = rbind(c(1, 0),
          c(0, 1),
          c(1, 0),
```

```
          c(0, 1),
          c(1, 1),
          c(1, 1)
    )

# The probabiliies of answering each item correctly for individuals
# who do not lack any required attribute
pistar = rep(.9, J)

# Penalties for failing to have each of the required attributes
rstar  = .5 * Q

# Randomized alpha profiles
alpha  = As[sample(1:(K ^ 2), N, replace = TRUE, pis),]

# Simulate data
rrum_items = simcdm::sim_rrum_items(Q, rstar, pistar, alpha)

## Not run:
# Note: This portion of the code is computationally intensive.

# Recover simulation parameters with Gibbs Sampler
Gibbs.out = rrum(rrum_items, Q)

# Iterations to be discarded from chain as burnin
burnin = 1:5000

# Calculate summarizes of posterior distributions
rstar.mean  = with(Gibbs.out, apply(RSTAR[,,-burnin], c(1, 2), mean))
pistar.mean = with(Gibbs.out, apply(PISTAR[,-burnin], 1, mean))
pis.mean    = with(Gibbs.out, apply(PI[,-burnin], 1 ,mean))

## End(Not run)
```

Index

rrum, 2

simcdm::sim_rrum_items(), *3*