

Package: errum (via r-universe)

September 7, 2024

Title Exploratory Reduced Reparameterized Unified Model Estimation

Version 0.0.3

Description Perform a Bayesian estimation of the exploratory reduced reparameterized unified model (ErRUM) described by Culpepper and Chen (2018) <[doi:10.3102/1076998618791306](https://doi.org/10.3102/1076998618791306)>.

License GPL (>= 2)

URL <https://github.com/tmsalab/errum>

BugReports <https://github.com/tmsalab/errum/issues>

Depends R (>= 3.5.0)

Imports Rcpp (>= 1.0.0)

LinkingTo Rcpp, RcppArmadillo (>= 0.9.200)

Suggests simcdm

LazyData true

RoxygenNote 7.1.0

Roxygen list(markdown = TRUE)

Encoding UTF-8

Language en-US

Repository <https://tmsalab.r-universe.dev>

RemoteUrl <https://github.com/tmsalab/errum>

RemoteRef HEAD

RemoteSha 63460e94616b60fd15324089a14739220b8f716d

Contents

errum	2
Index	5

errum

Exploratory reduced Reparameterized Unified Model (ErRUM)

Description

Obtains samples from posterior distribution for the Exploratory reduced Reparameterized Unified Model (ErRUM).

Usage

```
errum(
  y,
  k = 3,
  burnin = 1000,
  chain_length = 10000,
  verbose = FALSE,
  X = matrix(1, nrow = ncol(y)),
  v0 = 4,
  v1 = 2,
  cv0 = 0.1,
  cv1 = 10,
  bnu = 16
)
```

Arguments

y	Binary responses to assessments in <i>matrix</i> form with dimensions $N \times J$.
k	Number of Attribute Levels as a positive integer.
burnin	Number of Observations to discard on the chain.
chain_length	Length of the MCMC chain
verbose	Display estimation progress updates.
X, v0, v1, cv0, cv1, bnu	Additional tuning parameters

Value

An *errum* object that has:

- PISTAR
- RSTAR
- PIs
- QS
- m_Delta
- Delta_biject

- M2
- M1
- NUS

See Also

`simcdm::attribute_bijection()`, `simcdm::sim_rrum_items()`

Examples

```
# Setup Simulation Parameters
N = 5
K = 3
J = 30
# Note:
# Sample size has been reduced to create a minimally
# viable example that can be run during CRAN's automatic check.
# Please make sure to have a larger sample size of around 3,000.

# Sample true attribute profiles
Z      = matrix(rnorm(N * K), N, K)
Sig    = matrix(.5, K, K)
diag(Sig) = 1
theta  = Z %%% chol(Sig)

thvals = matrix(qnorm((1:K) / (K + 1)),
                N, K, byrow = TRUE)

Alphas = 1 * (theta > thvals)

# Defining matrix of possible attribute profiles
As = as.matrix(expand.grid(c(0, 1), c(0, 1), c(0, 1)))
Q = rbind(As[rep(c(2, 3, 5), 4),],
          As[rep(c(4, 6, 7), 4),],
          As[rep(8, 6),])

# Use simulation functions available in simcdm
if (requireNamespace("simcdm", quietly = TRUE)) {

  a = As %%% simcdm::attribute_bijection(K)
  As = As[a + 1,]

  # Setting item parameters
  pistar = rep(.9, J)
  rstar = matrix(.6, J, K) * Q

  # Simulate data under rRUM model
  Y = simcdm::sim_rrum_items(Q, rstar, pistar, Alphas)

  # Estimation Settings
  chainLength = 10000 # Run with 20000
  burnin = chainLength / 2
```

```
# Gibbs Estimation
model = errum(Y, K, burnin, chainLength)
}
```

Index

`errum`, [2](#)

`simcdm::attribute_bijection()`, [3](#)

`simcdm::sim_rrum_items()`, [3](#)