Package: cIRT (via r-universe)

September 28, 2024

Type Package

Title Choice Item Response Theory

Version 1.3.2

Description Jointly model the accuracy of cognitive responses and item choices within a Bayesian hierarchical framework as described by Culpepper and Balamuta (2015) <doi:10.1007/s11336-015-9484-7>. In addition, the package contains the datasets used within the analysis of the paper.

License GPL (>= 2)

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

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

Depends R (>= 4.0.0)

Imports Rcpp (>= 1.0.8)

LinkingTo Rcpp, RcppArmadillo (>= 0.10.8.1.0)

Suggests knitr, rmarkdown, covr

LazyData true

Encoding UTF-8

RoxygenNote 7.1.2

Language en-US

VignetteBuilder knitr

NeedsCompilation yes

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

Maintainer James Joseph Balamuta <balamut2@illinois.edu>

Repository CRAN

Date/Publication 2022-02-21 20:50:02 UTC

Contents

`cIRT-package` *cIRT: Choice Item Response Theory*

Description

Jointly model the accuracy of cognitive responses and item choices within a Bayesian hierarchical framework as described by Culpepper and Balamuta (2015) <doi:10.1007/s11336-015-9484-7>. In addition, the package contains the datasets used within the analysis of the paper.

Author(s)

Maintainer: James Joseph Balamuta <`balamut2@illinois.edu`> (ORCID) [copyright holder]

Authors:

- Steven Andrew Culpepper <`sculpepp@illinois.edu`> (ORCID) [copyright holder]

See Also

Useful links:

- `https://tmsalab.github.io/cIRT/`
- `https://github.com/tmsalab/cIRT`
- Report bugs at `https://github.com/tmsalab/cIRT/issues`

center_matrix — *Center a Matrix*

Description

Obtains the mean of each column of the matrix and subtracts it from the given matrix in a centering operation.

Usage

```
center_matrix(x)
```

Arguments

x	A `matrix` with any dimensions

Details

The application of this function to a matrix mimics the use of a centering matrix given by:

$$C_n = I_n - \frac{1}{n} 1 1^T$$

Value

A `matrix` with the same dimensions of X that has been centered.

Author(s)

James Joseph Balamuta

See Also

`cIRT()`

Examples

```
nobs = 500
nvars = 20
x = matrix(rnorm(nobs * nvars), nrow = nobs, ncol = nvars)
r_centered = scale(x)
arma_centered1 = center_matrix(x)
```

choice_matrix *Choice Matrix Data*

Description

This data set contains the subject's choices and point values for the difficult questions.

Usage

```
choice_matrix
```

Format

A data frame with 3780 observations on the following 5 variables.

- `subject_id` Research Participant Subject ID. There are 102 IDs and each ID has 15 observations.
- `hard_q_id` The item ID of the hard question assigned to the student (16-30)
- `easy_q_id` The item ID of the easy question assigned to the student (1-15)
- `choose_hard_q` Selected either: Difficult Question (1) or Easy Question (0)
- `high_value` Range of values associated with Difficult Question that span from 12 to 16, repeated three times per subject
- `low_value` Range of values associated with Easy Question that span from 4 to 6, repeated five times per subject
- `is_correct_choice` Did the user select an item that was answered correctly?

Author(s)

Steven Andrew Culpepper and James Joseph Balamuta

Source

Choice38 Experiment at UIUC during Spring 2014 - Fall 2014

cIRT *Generic Implementation of Choice IRT MCMC*

Description

Builds a model using MCMC

Usage

```
cIRT(
  subject_ids,
  fixed_effects,
  B_elem_plus1,
  rv_effects,
  trial_matrix,
  choices_nk,
  burnit,
  chain_length = 10000L
)
```

Arguments

- `subject_ids` A `vector` that contains subject IDs for each line of data in the choice vector (e.g. For 1 subject that made 5 choices, we would have the number 1 appear five times consecutively.)
- `fixed_effects` A `matrix` with NK x P1 dimensions that acts as the design matrix for terms WITHOUT theta.
- `B_elem_plus1` A `V[[1]]` dimensional column `vector` indicating which zeta_i relate to theta_i.
- `rv_effects` A `matrix` with NK x V dimensions for random effects design matrix.
- `trial_matrix` A `matrix` with N x J dimensions, where J denotes the number of items presented. The matrix MUST contain only 1's and 0's.
- `choices_nk` A `vector` with NK length that contains the choice value e.g. 0 or 1.
- `burnit` An `int` that describes how many MCMC draws should be discarded.
- `chain_length` An `int` that controls how many MCMC draws there are. (> 0)

Value

A `list` that contains:

- `as` A `matrix` of dimension chain_length x J
- `bs` A `matrix` of dimension chain_length x J
- `gs` A `matrix` of dimension chain_length x P_1
- `Sigma_zeta_inv` An `array` of dimension V x V x chain_length
- `betas` A `matrix` of dimension chain_length x P_2

Author(s)

Steven Andrew Culpepper and James Joseph Balamuta

See Also

`TwoPLChoicemcmc()`, `probitHLM()`, `center_matrix()`, `rmvnorm()`, `rwishart()`, and `riwishart()`

Examples

```
## Not run:
# Variables
# Y = trial matix
# C = KN vector of binary choices
# N = #of subjects
# J = # of items
# K= # of choices
# atrue = true item discriminations
# btrue = true item locations
# thetatrue = true thetas/latent performance
# gamma = fixed effects coefficients
# Sig = random-effects variance-covariance
# subid = id variable for subjects

# Load the Package
library(cIRT)

# Load the Data
data(trial_matrix)
data(choice_matrix)

# Thurstone design matrices
all_nopractice = subset(all_data_trials, experiment_loop.thisN > -1)
hard_items = choice_matrix$hard_q_id
easy_items = choice_matrix$easy_q_id

D_easy = model.matrix( ~ -1 + factor(easy_items))
D_hard = -1 * model.matrix( ~ -1 + factor(hard_items))[, -c(5, 10, 15)]

# Defining effect-coded contrasts
high_contrasts = rbind(-1, diag(4))
rownames(high_contrasts) = 12:16
low_contrasts = rbind(-1, diag(2))
rownames(low_contrasts) = 4:6

# Creating high & low factors
high = factor(choice_matrix[, 'high_value'])
low = factor(choice_matrix[, 'low_value'])
contrasts(high) = high_contrasts
contrasts(low) = low_contrasts

fixed_effects = model.matrix( ~ high + low)
fixed_effects_base = fixed_effects[, 1]
fixed_effects_int = model.matrix( ~ high * low)


# Model with Thurstone D Matrix
system.time({
 out_model_thurstone = cIRT(
   choice_matrix[, 'subject_id'],
   cbind(fixed_effects[, -1], D_easy, D_hard),
```

```
   c(1:ncol(fixed_effects)),
   as.matrix(fixed_effects),
   as.matrix(trial_matrix),
   choice_matrix[, 'choose_hard_q'],
   20000,
   25000
 )
})


vlabels_thurstone = colnames(cbind(fixed_effects[, -1], D_easy, D_hard))
G_thurstone = t(apply(
 out_model_thurstone$gs0,
 2,
 FUN = quantile,
 probs = c(.5, .025, .975)
))

rownames(G_thurstone) = vlabels_thurstone
B_thurstone = t(apply(
 out_model_thurstone$beta,
 2,
 FUN = quantile,
 probs = c(.5, 0.025, .975)
))

rownames(B_thurstone) = colnames(fixed_effects)

S_thurstone = solve(
  apply(out_model_thurstone$Sigma_zeta_inv, c(1, 2), FUN = mean)
)

inv_sd = diag(1 / sqrt(diag(solve(
 apply(out_model_thurstone$Sigma_zeta_inv, c(1, 2),
       FUN = mean)
))))

inv_sd %*% S_thurstone %*% inv_sd
apply(out_model_thurstone$as, 2, FUN = mean)
apply(out_model_thurstone$bs, 2, FUN = mean)

## End(Not run)
```

---

direct_sum *Direct Sum of Matrices*

Description

Computes the direct sum of all matrices passed in via the list.

Usage

```
direct_sum(x)
```

Arguments

x	A `field<matrix>` or `list` containing matrices

Details

Consider matrix A ($M \times N$) and B ($K \times P$). A direct sum is a diagonal matrix $A(+)B$ with dimensions $(m + k)x(n + p)$.

Value

Matrix containing the direct sum of all matrices in the list.

Author(s)

James Joseph Balamuta

Examples

```
x = list(matrix(0, nrow = 5, ncol = 3),
         matrix(1, nrow = 5, ncol = 3))
direct_sum(x)

x = list(matrix(rnorm(15), nrow = 5, ncol = 3),
         matrix(rnorm(30), nrow = 5, ncol = 6),
         matrix(rnorm(18), nrow = 2, ncol = 9))
direct_sum(x)
```

Generate_Choice *Generate Observed Data from choice model*

Description

Generates observed cognitive and choice data from the IRT-Thurstone model.

Usage

```
Generate_Choice(
  N,
  J,
  K,
  theta,
  as,
  bs,
  zeta,
```

```
  gamma,
  X,
  W,
  subject_ids,
  unique_subject_ids
)
```

Arguments

N	An integer for the number of observations.
J	An integer for the number of items.
K	An integer for the number of paired comparisons.
theta	A vector of latent cognitive variables.
as	A vector of length J with item discriminations.
bs	A vector of length J with item locations.
zeta	A matrix with dimensions N x V containing random parameter estimates.
gamma	A vector with dimensions P x 1 containing fixed parameter estimates, where $P = P_1 + P_2$
X	A matrix with dimensions N*K x P_1 containing fixed effect design matrix without theta.
W	A matrix with dimensions N*K x V containing random effect variables.
subject_ids	A vector with length NK x 1 containing subject-choice IDs.
unique_subject_ids	A vector with length N x 1 containing unique subject IDs.

Value

A list that contains:

- Y A matrix of dimension N by J
- C A vector of length NK

Author(s)

Steven Andrew Culpepper and James Joseph Balamuta

payout_matrix *Payout Matrix Data*

Description

This data set contains the payout information for each subject.

Usage

```
payout_matrix
```

Format

A data frame with 252 observations on the following 4 variables.

`Participant` Subject ID

`cum_sum` Sum of all payouts

`num_correct_choices` Total number of correct choices (out of 15)

`num_correct_trials` Total number of correct trials (out of 30)

Author(s)

Steven Andrew Culpepper and James Joseph Balamuta

Source

Choice38 Experiment at UIUC during Spring 2014 - Fall 2014

probitHLM *Probit Hierarchical Level Model*

Description

Performs modeling procedure for a Probit Hierarchical Level Model.

Usage

```
probitHLM(
  unique_subject_ids,
  subject_ids,
  choices_nk,
  fixed_effects_design,
  rv_effects_design,
  B_elem_plus1,
  gamma,
```

```
    beta,
    theta,
    zeta_rv,
    WtW,
    Z_c,
    Wzeta_0,
    inv_Sigma_gamma,
    mu_gamma,
    Sigma_zeta_inv,
    S0,
    mu_beta,
    sigma_beta_inv
  )
```

Arguments

`unique_subject_ids`	A `vector` with length N x 1 containing unique subject IDs.
`subject_ids`	A `vector` with length N*K x 1 containing subject IDs.
`choices_nk`	A `vector` with length N*K x 1 containing subject choices.
`fixed_effects_design`	A `matrix` with dimensions N*K x P containing fixed effect variables.
`rv_effects_design`	A `matrix` with dimensions N*K x V containing random effect variables.
`B_elem_plus1`	A `V[[1]]` dimensional column `vector` indicating which zeta_i relate to theta_i.
`gamma`	A `vector` with dimensions P_1 x 1 containing fixed parameter estimates.
`beta`	A `vector` with dimensions P_2 x 1 containing random parameter estimates.
`theta`	A `vector` with dimensions N x 1 containing subject understanding estimates.
`zeta_rv`	A `matrix` with dimensions N x V containing random parameter estimates.
`WtW`	A `field<matrix>` P x P x N contains the caching for direct sum.
`Z_c`	A `vector` with dimensions N*K x 1
`Wzeta_0`	A `vector` with dimensions N*K x 1
`inv_Sigma_gamma`	A `matrix` with dimensions P x P that is the prior inverse sigma matrix for gamma.
`mu_gamma`	A `vector` with length P x 1 that is the prior mean vector for gamma.
`Sigma_zeta_inv`	A `matrix` with dimensions V x V that is the prior inverse sigma matrix for zeta.
`S0`	A `matrix` with dimensions V x V that is the prior sigma matrix for zeta.
`mu_beta`	A `vector` with dimensions P_2 x 1, that is the mean of beta.
`sigma_beta_inv`	A `matrix` with dimensions P_2 x P_2, that is the inverse sigma matrix of beta.

Details

The function is implemented to decrease the amount of vectorizations necessary.

Value

A `list` that contains:

`zeta_1` A `vector` of length N

`sigma_zeta_inv_1` A `matrix` of dimensions V x V

`gamma_1` A `vector` of length P

`beta_1` A `vector` of length V

`B` A `matrix` of length V

Author(s)

Steven Andrew Culpepper and James Joseph Balamuta

See Also

`rwishart()` and `TwoPLChoicemcmc()`

`riwishart` *Generate Random Inverse Wishart Distribution*

Description

Creates a random inverse wishart distribution when given degrees of freedom and a sigma matrix.

Usage

```
riwishart(df, S)
```

Arguments

`df`	An `integer` that represents the degrees of freedom. (> 0)
`S`	A `matrix` with dimensions m x m that provides Sigma, the covariance matrix.

Value

A `matrix` that is an inverse wishart distribution.

Author(s)

James Joseph Balamuta

See Also

`rwishart()` and `TwoPLChoicemcmc()`

Examples

```
#Call with the following data:
riwishart(3, diag(2))
```

rmvnorm — *Generate Random Multivariate Normal Distribution*

Description

Creates a random Multivariate Normal when given number of obs, mean, and sigma.

Usage

```
rmvnorm(n, mu, S)
```

Arguments

n	An `integer`, which gives the number of observations. (> 0)
mu	A `vector` length m that represents the means of the normals.
S	A `matrix` with dimensions m x m that provides Sigma, the covariance matrix.

Value

A `matrix` that is a Multivariate Normal distribution.

Author(s)

James Joseph Balamuta

See Also

`TwoPLChoicemcmc()` and `probitHLM()`

Examples

```
# Call with the following data:
rmvnorm(2, c(0,0), diag(2))
```

rwishart — *Generate Random Wishart Distribution*

Description

Creates a random wishart distribution when given degrees of freedom and a sigma matrix.

Usage

```
rwishart(df, S)
```

Arguments

- `df` An `integer`, which gives the degrees of freedom of the Wishart. (> 0)
- `S` A `matrix` with dimensions m x m that provides Sigma, the covariance matrix.

Value

A `matrix` that is a Wishart distribution, aka the sample covariance matrix of a Multivariate Normal Distribution

Author(s)

James Joseph Balamuta

See Also

`riwishart()` and `probitHLM()`

Examples

```
# Call with the following data:
rwishart(3, diag(2))

# Validation
set.seed(1337)
S = toeplitz((10:1)/10)
n = 10000
o = array(dim = c(10,10,n))
for(i in 1:n){
o[,,i] = rwishart(20, S)
}
mR = apply(o, 1:2, mean)
Va = 20*(S^2 + tcrossprod(diag(S)))
vR = apply(o, 1:2, var)
stopifnot(all.equal(vR, Va, tolerance = 1/16))
```

`survey_data` *Survey Data*

Description

This data set contains the subject's responses survey questions administered using Choice38.

Usage

```
survey_data
```

Format

A data frame with 102 observations on the following 2 variables.

`id` Subject's Assigned Research ID

`sex` Subject's sex:

- Male
- Female

Author(s)

Steven Andrew Culpepper and James Joseph Balamuta

Source

Choice38 Experiment at UIUC during Spring 2014 - Fall 2014

`Total_Tabulate` *Calculate Tabulated Total Scores*

Description

Internal function to -2LL

Usage

```
Total_Tabulate(N, J, Y)
```

Arguments

`N`	An `integer`, which gives the number of observations. (> 0)
`J`	An `integer`, which gives the number of items. (> 0)
`Y`	A N by J `matrix` of item responses.

Value

A vector of tabulated total scores.

Author(s)

Steven Andrew Culpepper

trial_matrix *Trial Matrix Data*

Description

This data set contains the subject's responses to items. Correct answers are denoted by 1 and incorrect answers are denoted by 0.

Usage

```
trial_matrix
```

Format

A data frame with 252 observations on the following 30 variables.

`t1` Subject's Response to Item 1.

`t2` Subject's Response to Item 2.

`t3` Subject's Response to Item 3.

`t4` Subject's Response to Item 4.

`t5` Subject's Response to Item 5.

`t6` Subject's Response to Item 6.

`t7` Subject's Response to Item 7.

`t8` Subject's Response to Item 8.

`t9` Subject's Response to Item 9.

`t10` Subject's Response to Item 10.

`t11` Subject's Response to Item 11.

`t12` Subject's Response to Item 12.

`t13` Subject's Response to Item 13.

`t14` Subject's Response to Item 14.

`t15` Subject's Response to Item 15.

`t16` Subject's Response to Item 16.

`t17` Subject's Response to Item 17.

`t18` Subject's Response to Item 18.

`t19` Subject's Response to Item 19.

`t20` Subject's Response to Item 20.

`t21` Subject's Response to Item 21.

`t22` Subject's Response to Item 22.

`t23` Subject's Response to Item 23.

`t24` Subject's Response to Item 24.

t25 Subject's Response to Item 25.

t26 Subject's Response to Item 26.

t27 Subject's Response to Item 27.

t28 Subject's Response to Item 28.

t29 Subject's Response to Item 29.

t30 Subject's Response to Item 30.

Author(s)

Steven Andrew Culpepper and James Joseph Balamuta

Source

Choice38 Experiment at UIUC during Spring 2014 - Fall 2014

TwoPLChoicemcmc *Two Parameter Choice IRT Model MCMC*

Description

Performs an MCMC routine for a two parameter IRT Model using Choice Data

Usage

```
TwoPLChoicemcmc(
  unique_subject_ids,
  subject_ids,
  choices_nk,
  fixed_effects,
  B,
  rv_effects_design,
  gamma,
  beta,
  zeta_rv,
  Sigma_zeta_inv,
  Y,
  theta0,
  a0,
  b0,
  mu_xi0,
  Sig_xi0
)
```

Arguments

`unique_subject_ids`
: A `vector` with length $N \times 1$ containing unique subject IDs.

`subject_ids`
: A `vector` with length $NK \times 1$ containing subject IDs.

`choices_nk`
: A `vector` with length $NK \times 1$ containing subject choices.

`fixed_effects`
: A `matrix` with dimensions $NK \times P_1$ containing fixed effect design matrix without theta.

`B`
: A V dimensional column `vector` relating θ_i and ζ_i.

`rv_effects_design`
: A `matrix` with dimensions $NK \times V$ containing random effect variables.

`gamma`
: A `vector` with dimensions $P \times 1$ containing fixed parameter estimates, where $P = P_1 + P_2$

`beta`
: A `vector` with dimensions P_2 containing random parameter estimates.

`zeta_rv`
: A `matrix` with dimensions $N \times V$ containing random parameter estimates.

`Sigma_zeta_inv`
: A `matrix` with dimensions $P_2 \times P_2$.

`Y`
: A `matrix` of dimensions $N \times J$ for Dichotomous item responses

`theta0`
: A `vector` of length $N \times 1$ for latent theta.

`a0`
: A `vector` of length J for item discriminations.

`b0`
: A `vector` of length J for item locations.

`mu_xi0`
: A `vector` of dimension 2 (i.e. c(0,1)) that is a prior for item parameter means.

`Sig_xi0`
: A `matrix` of dimension 2x2 (i.e. diag(2)) that is a prior for item parameter vc matrix.

Value

A `list` that contains:

`ai1` A `vector` of length J

`bi1` A `vector` of length J

`theta1` A `vector` of length N

`Z_c` A `matrix` of length NK

`Wzeta_0` A `matrix` of length NK

Author(s)

Steven Andrew Culpepper and James Joseph Balamuta

See Also

`cIRT()`, `rmvnorm()`, and `riwishart()`

Examples

```
## Not run:
# Call with the following data:
TwoPLChoicemcmc(cogDAT, theta0, a0, b0, mu_xi0, Sig_xi0)

## End(Not run)
```

Index