Package: IRTM (via r-universe)

October 19, 2026

Title Theory-Driven Item Response Theory (IRT) Models

Version 0.0.1.2

Description IRT-M is a semi-supervised approach based on Bayesian Item Response Theory that produces theoretically identified underlying dimensions from input data and a constraints matrix. The methodology is fully described in 'Morucci et al. (2024), ``Measurement That Matches Theory: Theory-Driven Identification in Item Response Theory Models'". Details are available at <https://www.cambridge.org/core/journals/american-political-science-review/article/measurement-that-matches-theory-theorydriven-identification-in-item-response-theory-models/395DA1DFE3DCD7B866DC053D7554A30B>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends truncnorm, tmvtnorm, utils, RcppProgress, RcppDist, ggplot2, R (>= 3.5.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0), RColorBrewer, fastDummies, ggrepel, tidyverse, spelling

VignetteBuilder knitr

LazyData true

LinkingTo Rcpp, RcppArmadillo, RcppDist, RcppProgress

Imports coda, Rcpp, RcppArmadillo, ggridges, rlang, dplyr, reshape2,

Config/testthat/edition 3

Language en-US

NeedsCompilation yes

Author Marco Morucci [aut], Margaret Foster [aut] (ORCID: <https://orcid.org/0000-0002-6418-8394>), David Siegel [aut, cre] (ORCID: <https://orcid.org/0000-0003-1619-6119>)

Maintainer David Siegel <david.siegel@duke.edu>

Config/pak/sysreqs libicu-dev

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

Date/Publication 2026-04-21 08:12:49 UTC

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

RemoteRef HEAD

RemoteSha a9559886041480ce5785d4ac86f3f0f5bcc30766

Contents

dim_corr 2
get_lambdas 3
Geweke_convergence 4
irt_m 4
irt_vis 5
M_constrained_irt 6
MCodes 7
mse 8
pair_gen_anchors 9
standardize_theta 9
synth_idvs 10
synth_questions 10
theta_av 11
theta_lambda_traceplots 11

Index **12**

`dim_corr` *learned correlations*

Description

Takes as input either the Sigma covariance matrix, if the user has learned the factor covariance, or the Omega covariance matrix, if the user has learned the loading covariance, as well as a vector of dimension names. Returns a correlation matrix with correlations between the dimensions.

Usage

```
dim_corr(cov_array, dim_names = NULL)
```

Arguments

`cov_array`	An array of dimension (d x d x nsamp/thin) containing posterior samples of the relevant covariance matrix.
`dim_names`	Vector of dimension names.

Value

A data frame containing the correlation matrix derived from t input covariance array, with rows and columns labeled according to dim_names (if provided). Each cell represents the correlation between the corresponding dimensions.

`get_lambdas` *get_lambdas*

Description

Takes as input the array of lambdas from the irt list, a vector of item names (can be taken from either Y_in or M_matrix), a vector of dimension names, and, optionally, a vector comprising elaborations about each item. Returns a list containing a data frame with the mean lambdas for each item-dimension pair, possibly attaching elaborations to each item's string, and a data frame with the items with the highest mean values of lambda for each dimension in order

Usage

```
get_lambdas(lambda_array, item_names, dim_names, item_elab = NULL)
```

Arguments

`lambda_array`	An array of dimension (K x d x nsamp/thin) containing posterior samples of item discrimination parameters.
`item_names`	Vector of item names.
`dim_names`	Vector of dimension names.
`item_elab`	A vector comprising elaborations about each item (Default = NULL).

Value

A list containing the following components:

`av_lams`	A data frame of dimension (K x (1+d)) containing averages of item discrimination parameters.
`high_lams`	A data frame of dimension (K x d) containing an ordered list of the items with the highest mean values of lambda for each dimension.

Geweke_convergence *Geweke Convergence*

Description

Runs Geweke tests to assess MCMC convergence

Usage

```
Geweke_convergence(THETA)
```

Arguments

THETA	Matrix of parameter estimates from IRTM

Value

Proportion of values that fail the Geweke convergence test ($p < 0.05$) for each parameter

irt_m *irt_m*

Description

This function is a wrapper to enable easier use of the IRT-M model in M_constrained_irt. It takes as input two data frames: a N x K data frame, and a K x (1+d) M-matrix. The first column of the M-matrix should contain item identifiers that match the K column headers in the N x K data frame. If they do not match, the wrapper exits with an error. The wrapper computes anchors, Y_all (merged data and anchors), and a list of diagonal M-Matrices. The second two are used as inputs to M_constrained_irt, which runs the sampler. Also used as input are nburn (Default = 10^3), nsamp (Default = 10^3), thin (Default = 1), and learn_loadings (Default = FALSE). This last one defaults to having the sampler learn factor covariances. If set to true, it will learn loading covariances instead. Finally, the wrapper removes the anchors and returns an irt list.

Usage

```
irt_m(
  Y_in,
  d,
  M_matrix = NULL,
  nburn = 1000,
  nsamp = 1000,
  thin = 1,
  learn_loadings = FALSE
)
```

Arguments

Y_in	a N x K matrix of responses given by N respondents to K items. Can contain missing values. Column names should match first column in M_matrix.
d	an integer specifying the number of latent dimensions.
M_matrix	a K x (d+1) matrix of theoretical codings used to constrain IRT-M (default=NULL). First column should match column names in Y_in.
nburn	an integer specifying the number of burn-in MCMC iterations.
nsamp	an integer specifying the number of sampling MCMC iterations.
thin	an integer specifying the number of thinning MCMC samples.
learn_loadings	a Boolean specifying whether a covariance matrix for the latent loadings should be learned, instead of the default covariance matrix for latent dimensions.

Value

A list containing the following components:

lambda	An array of dimension (K x d x nsamp/thin) containing posterior samples of item discrimination parameters.
b	A matrix of dimension (K x nsamp/thin) containing posterior samples of item difficulty parameters.
theta	An array of dimension (N x d x nsamp/thin) containing posterior samples of respondent latent trait values.
Sigma	An array of dimension (d x d x nsamp/thin) containing posterior samples of the covariance matrix of latent traits (only if learn_Sigma=TRUE).
Omega	An array of dimension (d x d x nsamp/thin) containing posterior samples of the covariance matrix of item loadings (only if learn_Omega=TRUE).

---

irt_vis *irt_vis*

Description

Takes as input the number of latent dimensions (d), an N x (d+z) data frame with average thetas in the first d columns and variables not included in the calculation of the thetas in the rest (T_out), and, optionally, a variable name (sub_name) taken from T_out, and an output file name (out_file), and returns either unconditional theta distributions or distributions subset by that variable

Usage

```
irt_vis(d, T_out, sub_name = NULL, out_file = NULL)
```

Arguments

d	The number of latent dimensions
T_out	N x (d+z) data frame with average latent dimensions in first d columns
sub_name	The name of a variable in T_out used for levels in the plot (Default = NULL)
out_file	Output file name for plot (Default = NULL)

Value

A ggplot2 object containing density ridge plots of the latent dimensions. When sub_name is NULL, the plot shows the distribution of each theta dimension. When sub_name is provided, the plot shows distributions faceted by theta dimension and grouped by the specified variable.

M_constrained_irt *M_constrained_irt*

Description

This function allows you to run the IRT model.

Usage

```
M_constrained_irt(
  Y,
  d,
  M = NULL,
  theta_fix = NULL,
  which_fix = NULL,
  nburn = 1000,
  nsamp = 1000,
  thin = 10,
  learn_Sigma = TRUE,
  learn_Omega = FALSE,
  hyperparameters = list(),
  display_progress = TRUE
)
```

Arguments

Y	a N x K matrix of responses given by N respondents to K items. Can contain missing values.
d	an integer specifying the number of latent dimensions.
M	a list of K d x d matrices (default=NULL).
theta_fix	a matrix with d columns containing the values of the latent dimensions for respondents that have pre-specified latent factors.

which_fix	a vector containing the indices of the respondents for which latent factors have been fixed.
nburn	an integer specifying the number of burn-in MCMC iterations.
nsamp	an integer specifying the number of sampling MCMC iterations.
thin	an integer specifying the number of thinning MCMC samples.
learn_Sigma	a Boolean specifying whether a covariance matrix for the latent factors should be learned.
learn_Omega	a Boolean specifying whether a covariance matrix for the latent loadings should be learned.
hyperparameters	a list of hyperparameters for the model.
display_progress	a Boolean specifying whether a progress bar should be displayed.

Value

A list containing the following components:

lambda	An array of dimension (K x d x nsamp/thin) containing posterior samples of item discrimination parameters.
b	A matrix of dimension (K x nsamp/thin) containing posterior samples of item difficulty parameters.
theta	An array of dimension (N x d x nsamp/thin) containing posterior samples of respondent latent trait values.
Sigma	An array of dimension (d x d x nsamp/thin) containing posterior samples of the covariance matrix of latent traits (only if learn_Sigma=TRUE).
Omega	An array of dimension (d x d x nsamp/thin) containing posterior samples of the covariance matrix of item loadings (only if learn_Omega=TRUE).

MCodes *Methodological Codes*

Description

Factor loading matrix for IRT-M vignette. This is a 793 row and 9 column dataset. The rows are derived from the binary encoding of the synthetic survey, with a row for every binarized question in the survey. The first 56 rows are retained metadata, and have lots of NA.

The data format is an intermediary processing for IRT-M, and is detailed in the vignette text.

@format A data frame with the following variables: #'

QCode Mapping of the dimension coding key to the underlying question in the original (synthetic) survey data. The first 56 rows are blank because they map to survey and respondent metadata that doesn't relate to dimensions.

QMap Mapping to the question in data processed by the vignette for variable estimation.

SubstantiveNotes Brief human readable comments on the substantive meaning of the coded questions. These are for convenience of reference.

D1-Culture threat Loading vector for the cultural threat dimension. A 1 indicates that the question is expected to load.

D2-ReligionThreat Loading vector for the religious threat dimension.

D3-Economic Threat Loading vector for the economic threat dimension.

D4-HealthThreat Loading vector for the health threat dimension.

O1-OutcomeSupportImmigration Loading vector for the immigration support composite.

O2-OutcomeSupportEU Loading vector for the European Union support composite.

Details

Datasets for IRT-M Package

MCodes, synth_idvs, and synth_questions are included in the vignette

Source

IRT-M vignette walk through.

`mse` *Mean Squared Error*

Description

loss function for benchmarks

Usage

```
mse(ytrue, ypred, aggregate = TRUE, root = FALSE)
```

Arguments

`ytrue`	observed values,
`ypred`	predicted values
`aggregate`	logical for whether to take mean of estimate
`root`	logical for whether to return square root of MSE

Value

mean squared error

pair_gen_anchors *pair_gen_anchors*

Description

This function generates anchor points from the M matrices. It creates d(d-1)*4 fake respondents such that, for every pair of dimensions: The first respondent has an extremely positive value of both dimensions of the pair, The second has an extremely positive value for dim 1 and an extremely negative value for dim 2 of each pair, The third has an extremely negative value for dim 1 and an extremely positive value for dim 2 of each pair, The fourth has an extremely negative value of both dimensions of the pair. These respondents' answers are imputed according to the directions of loadings specified by the M-matrices, i.e., if question k loads positively on dim 1 and positively on dim 2, the first respondent in the dim 1/dim 2 pair will have yes for question k. If question k+1 loads positively on dim 1 and negatively on dim 2 then the second respondent in the dim 1/dim 2 pair will have yes on question k+1 and so on.

Usage

```
pair_gen_anchors(M, A)
```

Arguments

M	a list containing K dxd M-matrices
A	What value should be considered extreme for the latent dimensions.

Value

A list with two elements:

Yfake	A matrix of dimension (d(d-1)*4 x K) containing the imputed answers of the fake anchor respondents, where d is the number of dimensions and K is the number of questions. Values are 0 (no) or 1 (yes).
theta_fake	A list of d(d-1)*4 vectors, each of length d, representing the latent trait values for the fake anchor respondents. Each vector contains mostly zeros, with extreme values (A or -A) at the positions corresponding to the pair of dimensions being considered.

standardize_theta *Standardize Theta*

Description

standardizes theta estimates

Usage

```
standardize_theta(theta, Sigma)
```

Arguments

theta	estimated object
Sigma	covariance matrix

Value

theta divided by sigma param

---

synth_idvs *Synthetic Independent Variables*

Description

A synthetic dataset of independent variables for post-estimate analysis in the vignette. Extraction of the data from the synthetic survey is described in the vignette.

Format

A 3000 row and 27 column dataset of synthetic survey responses. This closely follows the 94.3 Eurobarometer survey in structure. This dataset is a toy that is intended for the IRT-M vignette. For real analysis, see the original Eurobarometer data collection.

---

synth_questions *Questions for the Synthetic European sentiment survey in the vignette*

Description

A synthetic dataset with 3000 rows and 148 questions. This data replicates the structure of questions following Eurobarometer 94.3 (2021). It is not intended to be analyzed independently from the vignette. '

Format

A data frame with 3000 rows and 148 columns representing synthetic survey responses

theta_av *Average Thetas*

Description

Compute matrix of theta means over posterior distributions

Usage

```
theta_av(theta_array)
```

Arguments

theta_array	An array of dimension (N x d x nsamp/thin) containing posterior samples of respondent latent trait values

Value

theta_av N x d matrix of average thetas

theta_lambda_traceplots *Theta Lambda Traceplots*

Description

Creates traceplots for IRT parameter convergence diagnostics

Usage

```
theta_lambda_traceplots(irt, i = NULL, k = NULL)
```

Arguments

irt	An object containing theta and lambda parameters from an IRTM model
i	Index of the respondent to plot (randomly selected if NULL)
k	Index of the item to plot (randomly selected if NULL)

Value

Plots of theta, lambda, and their product across MCMC iterations

Index

∗ **datasets**
MCodes, 7
synth_idvs, 10
synth_questions, 10

dim_corr, 2

get_lambdas, 3
Geweke_convergence, 4

irt_m, 4
irt_vis, 5

M_constrained_irt, 6
MCodes, 7
mse, 8

pair_gen_anchors, 9

standardize_theta, 9
synth_idvs, 10
synth_questions, 10

theta_av, 11
theta_lambda_traceplots, 11