Package: GMLTM (via r-universe)

June 30, 2026

Type Package

Title Generalized Multicomponent Latent Trait Model for Diagnosis

Version 0.1.0

Description Provides Bayesian estimation of Item Response Theory models that decompose item difficulty into cognitive operations or rules. Implements the Linear Logistic Test Model (LLTM; Fischer (1973) <doi:10.1016/0001-6918(73)90003-6>), the Multicomponent Latent Trait Model for Diagnosis (MLTM-D; Embretson and Yang (2013) <doi:10.1007/s11336-012-9296-y>), and the Generalized Multicomponent Latent Trait Model for Diagnosis (GMLTM-D; Ramirez et al. (2024) <doi:10.3390/jintelligence12070067>). All models are estimated via Hamiltonian Monte Carlo using 'Stan' through the 'rstan' interface. Includes tools for model validation, reliability estimation, and visualization of item characteristic curves. Supports user-defined prior distributions for all model parameters.

License GPL (>= 3)

URL https://github.com/Eduar-Ramirez/GMLTM-D

BugReports https://github.com/Eduar-Ramirez/GMLTM-D/issues

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports rstan (>= 2.21.0), ggplot2, gridExtra, grid, utils, parallel, loo, RColorBrewer

Suggests testthat (>= 3.0.0), knitr, rmarkdown

SystemRequirements C++17, GNU make

VignetteBuilder knitr

Language en-US

RoxygenNote 7.3.2

NeedsCompilation no

Author Eduar Ramirez [aut, cre], Marcos Jimenez [aut], Vithor R. Franco [aut], Jesus Alvarado [aut]

Maintainer Eduar Ramirez <edrami02@ucm.es>

Config/pak/sysreqs make

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

Date/Publication 2026-06-30 11:40:07 UTC

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

RemoteRef HEAD

RemoteSha eb15116465800017bc1d8a0a93914a08e0895328

Contents

`analogy` *Analogy items dataset*

Description

Binary item response data from a figural analogies test used to illustrate the LLTM, MLTM-D, and GMLTM-D models.

Usage

```
analogy
```

Format

A matrix with 149 rows (subjects) and 27 columns (items), where each cell contains a binary response (0 = incorrect, 1 = correct).

Source

Blum, D., Holling, H., Galibert, M. S., & Forthmann, B. (2016). Task difficulty prediction of figural analogies. *Intelligence*, 56, 72–81. doi:10.1016/j.intell.2016.03.001

References

Ramirez, E. S., Jimenez, M., Franco, V. R., & Alvarado, J. M. (2024). Delving into the complexity of analogical reasoning: A detailed exploration with the Generalized Multicomponent Latent Trait Model for Diagnosis. *Journal of Intelligence*, 12, 67. doi:10.3390/jintelligence12070067

`check_reliability_data_quality` *Check Data Quality for Reliability Analysis*

Description

Verifies if the data is suitable for robust reliability analysis.

Usage

```
check_reliability_data_quality(fit)
```

Arguments

`fit`	Fitted model

Value

List with data quality diagnostics

compare_conditional_reliability
Compare Conditional Reliability Between Components at Specific Theta Values

Description

Compare Conditional Reliability Between Components at Specific Theta Values

Usage

```
compare_conditional_reliability(cond_rel_obj, theta_points = c(-1, 0, 1))
```

Arguments

cond_rel_obj	An object of class conditional_reliability_tif returned by conditional_reliability_tif.
theta_points	Numeric vector of theta values at which to compare components. Default is c(-1, 0, 1).

Value

Invisibly returns a data frame with columns theta, reliability, and se evaluated at the requested theta_points. Called primarily for its side effect of printing a formatted summary table to the console.

compute_model_validation
Compute LOO and WAIC for GMLTM models

Description

This function extracts the log-likelihood from a GMLTM model and computes the Leave-One-Out Cross-Validation (LOO) and the Widely Applicable Information Criterion (WAIC). LOO is a Bayesian model comparison metric based on Pareto-smoothed importance sampling, while WAIC is a fully Bayesian criterion that estimates predictive accuracy.

Usage

```
compute_model_validation(fit)
```

Arguments

fit	A fitted GMLTM model or a list of fitted models.

Value

If a single model is provided, returns a list with LOO and WAIC results. If multiple models are provided, returns a summary table with key LOO and WAIC indices.

References

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using LOO-CV and WAIC. Statistics and Computing, 27(5), 1413–1432. doi:10.1007/s1122201696964

Examples

```
if (!requireNamespace("rstan", quietly = TRUE)) return()
data(analogy)
Q <- structure(
  c(0,0,1,0,1,0,1,0,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,0,1,0,1,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,1,0,1,1,0,1,0,0,0,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,
    1,0,0,0,1,1,0,1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,0,1),
  dim = c(27L, 5L),
  dimnames = list(NULL, c("rot_fig","rot_trap","reflection",
                          "subt_seg","mov_point")))
components <- list(global = c(1, 2, 3), local = c(4, 5))
fit1 <- GMLTM(data = analogy, Q = Q, components = components,
              iters = 200, iter_warmup = 100, chains = 1)
compute_model_validation(fit1)
```

`conditional_reliability_tif` *Conditional Reliability based on Test Information Function (TIF)*

Description

Calculates conditional reliability using Test Information Function for 3-parameter MLTM-D models. This approach is more precise than quantile-based partitioning methods.

Usage

```
conditional_reliability_tif(
  fit,
  theta_range = seq(-3, 3, 0.2),
  component = NULL,
  n_samples = 1000
)
```

Arguments

`fit`	Fitted model with α, β, guessing parameters
`theta_range`	Vector of θ values where to evaluate reliability
`component`	Integer or character. Specific component to analyze
`n_samples`	Integer. Number of posterior samples to use

Value

A list of class `"conditional_reliability_tif"` with elements:

`theta` Numeric vector of theta values.

`reliability` Numeric vector of reliability estimates at each theta value.

`information` Numeric vector of test information values at each theta value.

`component` Integer indicating the model component.

`fit` The original fitted model object.

---

demo_reliability_analysis

Step-by-step Usage Example

Description

Demonstration function to show how to use the optimized functions.

Usage

```
demo_reliability_analysis(fit)
```

Arguments

`fit`	Fitted model

Value

Invisibly returns a list with the reliability estimates computed at each step of the analysis. Called primarily for its side effect of printing a step-by-step explanation to the console.

Examples

```
if (!requireNamespace("rstan", quietly = TRUE)) return()
data(analogy)
Q <- structure(
  c(0,0,1,0,1,0,1,0,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,0,1,0,1,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,1,0,1,1,0,1,0,0,0,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,
    1,0,0,0,1,1,0,1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,0,1),
  dim = c(27L, 5L),
  dimnames = list(NULL, c("rot_fig","rot_trap","reflection",
                          "subt_seg","mov_point")))
components <- list(global = c(1, 2, 3), local = c(4, 5))
fit <- GMLTM(analogy, Q, components, iters = 200, iter_warmup = 100, chains = 1)
demo_reliability_analysis(fit)
```

---

enhanced_mltm_reliability

Enhanced Reliability Analysis for GMLTM-D Models

Description

Provides comprehensive reliability analysis for General Multicomponent Latent Trait Models for Diagnosis (GMLTM-D) using Bayesian posterior distributions. Optimized for speed and minimal dependencies.

Usage

```
enhanced_mltm_reliability(
  fit,
  include_conditional = FALSE,
  include_hierarchical = TRUE,
  include_comparisons = TRUE,
  n_samples = NULL
)
```

Arguments

`fit` A fitted GMLTM, MLTM, or LLTM model object containing posterior samples of theta parameters.

`include_conditional` Logical. Whether to compute conditional reliability estimates across ability levels. Default is FALSE for speed.

`include_hierarchical` Logical. Whether to compute hierarchical reliability for the general factor. Default is TRUE.

`include_comparisons`
: Logical. Whether to perform Bayesian comparisons between components. Default is TRUE.

`n_samples`
: Integer. Number of posterior samples to use (for speed control). If NULL, uses all available samples.

Value

An object of class `enhanced_mltm_reliability`.

`export_reliability_results` *Export Reliability Results*

Description

Exports results in tabular format for publications.

Usage

```
export_reliability_results(reliability_obj, file_name = NULL)
```

Arguments

`reliability_obj`
: Object of class enhanced_mltm_reliability

`file_name`
: File name (optional)

Value

data.frame with tabulated results

`generate_Q_with_interactions` *Generate an Extended Q-matrix with Rule Interactions and Collinearity Diagnostics*

Description

This function generates interaction terms between rules within the same component, extends the Q-matrix, and evaluates the resulting matrix for collinearity issues using eigenvalues, condition indices, and variance inflation factors (VIF). If severe collinearity is detected, it attempts to iteratively remove problematic interaction terms while keeping the original rules untouched.

Usage

```
generate_Q_with_interactions(
  Q,
  M_list,
  max_condition_index = 30,
  min_eigenvalue = 0.1,
  plot_diagnostics = TRUE,
  verbose = TRUE,
  save_to_global = TRUE
)
```

Arguments

`Q` A binary matrix of items by rules (original Q-matrix). Each row represents an item and each column represents a rule. Values should be 0 or 1.

`M_list` A list where each element contains the indices of rules that belong to the same component/dimension. For example, list(c(1,2,3), c(4,5)) indicates that rules 1,2,3 belong to component 1 and rules 4,5 belong to component 2.

`max_condition_index` Numeric. Maximum acceptable condition index. Default is 30. Values above this threshold indicate severe collinearity.

`min_eigenvalue` Numeric. Minimum acceptable eigenvalue. Default is 0.1. Values below this threshold may indicate linear dependence.

`plot_diagnostics` Logical. Whether to generate diagnostic plots. Default is TRUE.

`verbose` Logical. Whether to print detailed diagnostic information. Default is TRUE.

`save_to_global` Logical. Whether to save results to global environment. Default is TRUE.

Details

The function performs the following steps:

1. Validates input parameters
2. Generates interaction terms for rules within the same component
3. Performs collinearity diagnostics using multiple methods
4. Attempts to resolve severe collinearity by removing problematic interactions
5. Generates diagnostic plots and summaries

Collinearity is assessed using:

- Condition indices (based on eigenvalues of correlation matrix)
- Variance Inflation Factors (VIF)
- Matrix rank assessment
- Eigenvalue analysis

Value

A list containing:

`Q_extended` The extended Q-matrix with interaction terms

`M_list_extended`
Updated component list including interaction terms

`diagnostics` List with collinearity diagnostics

`removed_interactions`
Vector of removed interaction names (if any)

`plots` List of diagnostic plots (if plot_diagnostics = TRUE)

References

Belsley, D. A., Kuh, E., & Welsch, R. E. (1980). Regression diagnostics: Identifying influential data and sources of collinearity. John Wiley & Sons. O'Brien, R. M. (2007). A caution regarding rules of thumb for variance inflation factors. Quality & Quantity, 41(5), 673-690. Hair, J. F., Anderson, R. E., Tatham, R. L., & Black, W. C. (1995). Multivariate data analysis. Prentice Hall.

Examples

```
# Create a sample Q-matrix (5 items, 4 rules)
Q <- matrix(c(1,1,0,0,0,
              1,0,1,0,0,
              0,1,1,0,0,
              0,0,0,1,1), nrow=5, ncol=4, byrow=FALSE)

# Define components (rules 1-2 in component 1, rules 3-4 in component 2)
M_list <- list(c(1,2), c(3,4))

# Generate extended Q-matrix with interactions
result <- generate_Q_with_interactions(Q, M_list)

# Access results
extended_Q <- result$Q_extended
diagnostics <- result$diagnostics
plots <- result$plots
```

`GMLTM` *The Generalized Multicomponent Latent Trait Model for Diagnosis*

Description

Estimate the parameters of the GMLTM-D via Bayesian Hamiltonian Monte Carlo.

Usage

```
GMLTM(
  data,
  Q,
  components,
  iters = 2000,
  chains = 2,
  iter_warmup = 1000,
  quantiles = c(0.025, 0.5, 0.975),
  cores = parallel::detectCores() - 1,
 priors = list(theta = list(mu = 0, sigma = 1), eta = list(mu = 0, sigma = 1), alpha =
    list(mu = 0, sigma = 1), c = list(shape1 = 3, shape2 = 20)),
  ...
)
```

Arguments

`data`	An $n \times p$ matrix or data.frame of binary responses (rows = subjects, columns = items).
`Q`	A $p \times K$ matrix specifying which cognitive rules each item requires (Q-matrix).
`components`	A named list grouping rules into components. Each element is a numeric vector of rule indices belonging to that component. Example: `list(global = c(1,2,3), local = c(4,5))`.
`iters`	Number of post-warmup MCMC iterations per chain. Default is 2000.
`chains`	Number of Markov chains. Default is 2.
`iter_warmup`	Number of warmup iterations per chain. Default is 1000.
`quantiles`	Numeric vector of probabilities for posterior quantiles. Default is `c(0.025, 0.50, 0.975)`.
`cores`	Number of CPU cores for parallel chains. Default is `parallel::detectCores() - 1`.
`priors`	A named list of prior hyperparameters with elements `theta`, `eta`, `alpha`, and `c`. For Normal parameters supply `mu` and `sigma`; for the guessing parameter supply `shape1` and `shape2` (Beta prior). Unspecified elements retain defaults.
`...`	Additional arguments passed to `rstan::sampling()`.

Details

`GMLTM` estimates the Generalized Multicomponent Latent Trait Model for Diagnosis (GMLTM-D; Ramirez et al., 2024) in its Bayesian version. This model analyses items composed of cognitive rules or operations, incorporating three IRT parameters. Rules can be grouped into distinct components.

Prior distributions: Ability (θ) and rule difficulty (η) receive Normal priors. Discrimination (α) receives a half-Normal prior. Guessing (c) receives a Beta prior.

Value

A list of class "GMLTM" with elements:

`EAP` Posterior mean estimates: `theta`, `alpha`, `eta`, `beta`, `guessing`.

`quantiles` Posterior credible intervals for each parameter.

`posterior` Full posterior samples and derived quantities.

`fit` The `stanfit` object from `rstan::sampling`.

`data` The original data matrix.

`priors` The prior hyperparameters used.

References

Ramirez, E.S.; Jimenez, M.; Franco, V.R.; Alvarado, J.M. (2024). Delving into the Complexity of Analogical Reasoning: A Detailed Exploration with the Generalized Multicomponent Latent Trait Model for Diagnosis. *J. Intell.*, 12, 67. doi:10.3390/jintelligence12070067

Examples

```
if (!requireNamespace("rstan", quietly = TRUE)) return()
data(analogy)
Q <- structure(
  c(0,0,1,0,1,0,1,0,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,0,1,0,1,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,1,0,1,1,0,1,0,0,0,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,
    1,0,0,0,1,1,0,1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,0,1),
  dim = c(27L, 5L),
  dimnames = list(NULL, c("rot_fig","rot_trap","reflection",
                          "subt_seg","mov_point")))
components <- list(global = c(1, 2, 3), local = c(4, 5))
fit <- GMLTM(analogy, Q, components, iters = 200, iter_warmup = 100, chains = 1)
fit$EAP$eta
reliability(fit)
```

`integrate_with_enhanced_reliability`
Integration with Enhanced Reliability Analysis

Description

Integrates conditional TIF analysis with existing reliability functions.

Usage

```
integrate_with_enhanced_reliability(
  fit,
  include_conditional_tif = TRUE,
  theta_range = seq(-3, 3, 0.1),
  ...
)
```

Arguments

`fit` A fitted GMLTM, MLTM, or LLTM model object.

`include_conditional_tif`
Logical. Whether to compute conditional TIF-based reliability. Default is `TRUE`.

`theta_range` Numeric vector of theta values for conditional reliability evaluation. Default is `seq(-3, 3, 0.1)`.

`...` Additional arguments passed to `enhanced_mltm_reliability`.

Value

A list combining enhanced reliability results and, optionally, conditional TIF-based reliability.

LLTM *The Linear Logistic Test Model*

Description

Estimate the parameters of the LLTM via Bayesian Hamiltonian Monte Carlo.

Usage

```
LLTM(
  data,
  Q,
  iters = 2000,
  chains = 2,
  iter_warmup = 1000,
  quantiles = c(0.025, 0.5, 0.975),
  cores = parallel::detectCores() - 1,
  priors = list(theta = list(mu = 0, sigma = 1), eta = list(mu = 0, sigma = 1)),
  ...
)
```

Arguments

- `data` — An $n \times p$ matrix or data.frame of binary responses (rows = subjects, columns = items).
- `Q` — A $p \times K$ matrix specifying which cognitive rules each item requires (Q-matrix).
- `iters` — Number of post-warmup MCMC iterations per chain. Default is 2000.
- `chains` — Number of Markov chains. Default is 2.
- `iter_warmup` — Number of warmup iterations per chain. Default is 1000.
- `quantiles` — Numeric vector of probabilities for posterior quantiles. Default is `c(0.025, 0.50, 0.975)`.
- `cores` — Number of CPU cores for parallel chains. Default is `parallel::detectCores() - 1`.
- `priors` — A named list of prior hyperparameters. Each element is a named list with `mu` and `sigma`. Available parameters: `theta` (ability, Normal prior) and `eta` (rule difficulty, Normal prior). Unspecified elements retain defaults. Example: `priors = list(eta = list(sigma = 3))`.
- `...` — Additional arguments passed to `rstan::sampling()`.

Details

`LLTM` estimates the Bayesian version of the Linear Logistic Test Model (Fischer, 1973), which extends the Rasch model by decomposing item difficulty into cognitive rules. Item difficulty is expressed as $\beta_i = \mathbf{q}_i^\top \boldsymbol{\eta}$, where $\mathbf{q}_i$ is the i-th row of Q and $\boldsymbol{\eta}$ is the vector of rule difficulty parameters.

Prior distributions: Ability (θ) and rule difficulty (η) receive Normal priors. Prior sensitivity analysis is recommended.

Value

A list of class `"LLTM"` with elements:

`EAP` Posterior mean estimates: `theta`, `eta`, `beta`.

`quantiles` Posterior credible intervals for each parameter.

`posterior` Full posterior samples and derived quantities.

`fit` The `stanfit` object from `rstan::sampling`.

`data` The original data matrix.

`priors` The prior hyperparameters used.

References

Fischer, G. H. (1973). The linear logistic test model as an instrument in educational research. *Acta Psychologica*, 37(6), 359–374.

Ramirez, E.S.; Jimenez, M.; Franco, V.R.; Alvarado, J.M. (2024). Delving into the Complexity of Analogical Reasoning. *J. Intell.*, 12, 67. doi:10.3390/jintelligence12070067

Examples

```
if (!requireNamespace("rstan", quietly = TRUE)) return()
data(analogy)
Q <- structure(
  c(0,0,1,0,1,0,1,0,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,0,1,0,1,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,1,0,1,1,0,1,0,0,0,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,
    1,0,0,0,1,1,0,1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,0,1),
  dim = c(27L, 5L),
  dimnames = list(NULL, c("rot_fig","rot_trap","reflection",
                          "subt_seg","mov_point")))
fit <- LLTM(analogy, Q, iters = 200, iter_warmup = 100, chains = 1)
fit$EAP$eta
reliability(fit)
```

`marginal_Pchecks` *Marginal Proportions Predictive Checks*

Description

Computes and visualizes marginal success proportions, including predicted values, confidence intervals, RMSR, SRMR, and bias estimation.

Usage

```
marginal_Pchecks(fit, interval = 0.95)
```

Arguments

`fit`	MLTM object containing model results.
`interval`	Probability associated with the credibility intervals (default = 0.95).

Details

`marginal_Pchecks` calculates marginal prediction intervals and observed success proportions. It prints a table with observed vs. predicted values, generates a forest plot for visualization, and computes key fit indices: RMSR, SRMR, and bias.

Value

A list containing:

- `items`: A table of fitted values and prediction intervals for each item.
- `rmsr`: The Root Mean Square Residual (RMSR).
- `srmr`: The Standardized Root Mean Square Residual (SRMR).

- `bias`: The difference between the total observed and predicted proportions.

The function also generates:

- A forest plot visualizing prediction intervals and observed success probabilities.

MLTM *The Multicomponent Latent Trait Model for Diagnosis*

Description

Estimate the parameters of the MLTM-D via Bayesian Hamiltonian Monte Carlo.

Usage

```
MLTM(
  data,
  Q,
  components,
  iters = 2000,
  chains = 2,
  iter_warmup = 1000,
  quantiles = c(0.025, 0.5, 0.975),
  cores = parallel::detectCores() - 1,
 priors = list(theta = list(mu = 0, sigma = 1), eta = list(mu = 0, sigma = 1), alpha =
    list(mu = 0, sigma = 1)),
  ...
)
```

Arguments

`data`	An $n \times p$ matrix or data.frame of binary responses (rows = subjects, columns = items).
`Q`	A $p \times K$ matrix specifying which cognitive rules each item requires (Q-matrix).
`components`	A named list grouping rules into components. Each element is a numeric vector of rule indices belonging to that component. Example: `list(global = c(1,2,3), local = c(4,5))`.
`iters`	Number of post-warmup MCMC iterations per chain. Default is 2000.
`chains`	Number of Markov chains. Default is 2.
`iter_warmup`	Number of warmup iterations per chain. Default is 1000.
`quantiles`	Numeric vector of probabilities for posterior quantiles. Default is `c(0.025, 0.50, 0.975)`.
`cores`	Number of CPU cores for parallel chains. Default is `parallel::detectCores() - 1`.
`priors`	A named list of prior hyperparameters with elements `theta`, `eta`, and `alpha`. Each is a list with `mu` and `sigma`. `alpha` uses a half-Normal prior (truncated at 0). Unspecified elements retain defaults.
`...`	Additional arguments passed to `rstan::sampling()`.

Details

`MLTM` estimates the Bayesian version of the Multicomponent Latent Trait Model for Diagnosis (MLTM-D; Embretson & Yang, 2013). This noncompensatory model specifies a hierarchical relationship between components and rules.

Prior distributions: Ability (θ) and rule difficulty (η) receive Normal priors. Discrimination (α) receives a half-Normal prior.

Value

A list of class `"MLTM"` with elements:

`EAP` Posterior mean estimates: `theta`, `alpha`, `eta`, `beta`.

`quantiles` Posterior credible intervals for each parameter.

`posterior` Full posterior samples and derived quantities.

`fit` The `stanfit` object from `rstan::sampling`.

`data` The original data matrix.

`priors` The prior hyperparameters used.

References

Embretson, S. E., & Yang, X. (2013). A multicomponent latent trait model for diagnosis. *Psychometrika*, 78, 14–36.

Ramirez, E.S.; Jimenez, M.; Franco, V.R.; Alvarado, J.M. (2024). Delving into the Complexity of Analogical Reasoning. *J. Intell.*, 12, 67. doi:10.3390/jintelligence12070067

Examples

```
if (!requireNamespace("rstan", quietly = TRUE)) return()
data(analogy)
Q <- structure(
  c(0,0,1,0,1,0,1,0,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,0,1,0,1,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,1,0,1,1,0,1,0,0,0,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,
    1,0,0,0,1,1,0,1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,0,1),
  dim = c(27L, 5L),
  dimnames = list(NULL, c("rot_fig","rot_trap","reflection",
                          "subt_seg","mov_point")))
components <- list(global = c(1, 2, 3), local = c(4, 5))
fit <- MLTM(analogy, Q, components, iters = 200, iter_warmup = 100, chains = 1)
fit$EAP$eta
reliability(fit)
```

`plot.conditional_reliability_tif`
Plot Method for conditional_reliability_tif Objects

Description

Plot Method for conditional_reliability_tif Objects

Usage

```
## S3 method for class 'conditional_reliability_tif'
plot(x, ...)
```

Arguments

`x`	An object of class `conditional_reliability_tif`.
`...`	Additional arguments passed to `plot_conditional_reliability`.

Value

Invisibly returns `NULL`. Called for its side effect of producing a reliability and/or information plot via `plot_conditional_reliability()`.

`plot.enhanced_mltm_reliability`
Simple Plot Method (Minimal Dependencies)

Description

Creates basic plots using base R (no ggplot2 dependency).

Usage

```
## S3 method for class 'enhanced_mltm_reliability'
plot(x, type = "marginal", component = NULL, ...)
```

Arguments

`x`	Object of class `enhanced_mltm_reliability`.
`type`	Character. Type of plot: "marginal", "conditional", "comparison".
`component`	Integer or character. Specific component for conditional plots.
`...`	Additional plotting parameters.

Value

Invisibly returns `NULL`. Called for its side effects.

plot_all_components *Plot All Components Separately*

Description

Creates individual plots for each component in the analysis.

Usage

```
plot_all_components(
  results,
  plot_type = "both",
  include_ci = TRUE,
  color_scheme = "blue",
  save_plots = FALSE,
  output_dir = NULL,
  ...
)
```

Arguments

results	An object of class conditional_reliability_tif.
plot_type	Character. Type of plot: "reliability", "information", or "both". Default is "both".
include_ci	Logical. Whether to include credible interval bands. Default TRUE.
color_scheme	Character. Color scheme for plots. Default "blue".
save_plots	Logical. Whether to save plots to disk. Default FALSE.
output_dir	Character. Directory for saved plots. Default NULL (no files written). Only creates directory and saves when both save_plots = TRUE and output_dir is non-NULL.
...	Additional arguments passed to plot_conditional_reliability.

Value

Invisibly returns NULL. Called for its side effect of producing reliability plots for all model components.

plot_components_comparison

Create Comparison Plot of All Components

Description

Creates a single plot comparing conditional reliability across all components.

Usage

```
plot_components_comparison(
  results,
  include_ci = TRUE,
  show_optimal_points = TRUE,
  add_reference_lines = TRUE,
  color_palette = "Set2",
  save_plot = FALSE,
  filename = NULL,
  ...
)
```

Arguments

`results`	An object of class `conditional_reliability_tif`.
`include_ci`	Logical. Whether to include credible interval bands. Default `TRUE`.
`show_optimal_points`	Logical. Whether to mark optimal theta points. Default `TRUE`.
`add_reference_lines`	Logical. Whether to add horizontal reference lines at 0.7, 0.8, 0.9. Default `TRUE`.
`color_palette`	Character. RColorBrewer palette name for component colors. Default `"Set2"`.
`save_plot`	Logical. Whether to save the plot. Default `FALSE`.
`filename`	Character. Output filename if `save_plot = TRUE`. Default `NULL` (no file written). Only writes when both `save_plot = TRUE` and `filename` is non-`NULL`.
`...`	Additional graphical arguments.

Value

Invisibly returns `NULL`. Called for its side effect of producing a comparative reliability plot across model components.

`plot_conditional_reliability` *Plot Conditional Reliability Results*

Description

Creates plots for conditional reliability analysis with multiple visualization options

Usage

```
plot_conditional_reliability(
  results,
  component = NULL,
  plot_type = "both",
  include_ci = TRUE,
  ci_level = 0.95,
  color_scheme = "blue",
  add_reference_lines = TRUE,
  save_plot = FALSE,
  filename = NULL,
  ...
)
```

Arguments

`results`	Object of class conditional_reliability_tif
`component`	Integer or character. Component to plot (NULL for first component)
`plot_type`	Character. Type of plot: "reliability", "information", "both", "comparison"
`include_ci`	Logical. Include confidence intervals
`ci_level`	Numeric. Confidence level (0.90 or 0.95)
`color_scheme`	Character. Color scheme: "blue", "viridis", "custom"
`add_reference_lines`	Logical. Add reference lines for reliability levels
`save_plot`	Logical. Save plot to file
`filename`	Character. Filename if saving plot. Default `NULL` (no file written). Only writes when both `save_plot` = `TRUE` and `filename` is non-`NULL`.
`...`	Additional plotting parameters

Value

Invisibly returns `NULL`. Called for its side effect of producing one or two plots (reliability curve, test information function, or both) depending on `plot_type`.

plot_ICC_grouped *Grouped Item Characteristic Curves (ICC) Plot*

Description

Generates Item Characteristic Curves (ICCs) for a group of items displayed in a grid layout with a shared legend.

Usage

```
plot_ICC_grouped(
  fit,
  Q,
  components,
  page = 1,
  n_items_per_page = 9,
  ncol = 3,
  nrow = 3
)
```

Arguments

`fit`	A fitted GMLTM model object containing EAP parameter estimates.
`Q`	The Q-matrix indicating the association between items and rules.
`components`	A list where each element is a vector of rule indices per component.
`page`	Integer specifying which page of items to display.
`n_items_per_page`	Number of items to include per page. Default is 9.
`ncol`	Number of columns in the layout grid. Default is 3.
`nrow`	Number of rows in the layout grid. Default is 3.

Details

Displays one legend shared across all plots and ensures consistency across theta and probability axes. Ideal for publications or appendices.

Value

A composed ICC grid with one shared legend, plotted to the active device.

References

Ramirez, E.S.; Jimenez, M.; Franco, V.R.; Alvarado, J.M. (2024). Delving into the Complexity of Analogical Reasoning. *J. Intell.*, 12, 67. doi:10.3390/jintelligence12070067

Examples

```
if (!requireNamespace("rstan", quietly = TRUE)) return()
data(analogy)
Q <- structure(
  c(0,0,1,0,1,0,1,0,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,0,1,0,1,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,1,0,1,1,0,1,0,0,0,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,
    1,0,0,0,1,1,0,1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,0,1),
  dim = c(27L, 5L),
  dimnames = list(NULL, c("rot_fig","rot_trap","reflection",
                          "subt_seg","mov_point")))
components <- list(global = c(1, 2, 3), local = c(4, 5))
fit <- GMLTM(analogy, Q, components, iters = 200, iter_warmup = 100, chains = 1)
plot_ICC_grouped(fit, Q, components, page = 1)
```

`plot_ICC_individual` *Individual Item Characteristic Curves (ICC)*

Description

Returns a list of individual ICC `ggplot2` plots (one per item).

Usage

```
plot_ICC_individual(fit, Q, components)
```

Arguments

`fit`	A fitted GMLTM model object.
`Q`	The Q-matrix for rule-item associations.
`components`	A list indicating rule groupings per component.

Value

A list of individual `ggplot` objects, one per item.

Examples

```
if (!requireNamespace("rstan", quietly = TRUE)) return()
data(analogy)
Q <- structure(
  c(0,0,1,0,1,0,1,0,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,0,1,0,1,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,1,0,1,1,0,1,0,0,0,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,
```

```
    1,0,0,0,1,1,0,1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,0,1),
  dim = c(27L, 5L),
  dimnames = list(NULL, c("rot_fig","rot_trap","reflection",
                          "subt_seg","mov_point")))
components <- list(global = c(1, 2, 3), local = c(4, 5))
fit <- GMLTM(analogy, Q, components, iters = 200, iter_warmup = 100, chains = 1)
plots <- plot_ICC_individual(fit, Q, components)
print(plots[[1]])
```

---

ppchecks *Posterior Predictive Checks (PPC) for Model Fit Evaluation*

Description

This function generates posterior predictive checks by plotting the observed and simulated total scores distribution, comparing empirical data against model predictions. It also computes fitted values and prediction intervals.

Usage

```
ppchecks(fit, nsim = 100, interval = 0.95, ...)
```

Arguments

`fit`	A fitted MLTM object containing model results.
`nsim`	Number of simulated posterior samples (default = 100).
`interval`	Probability associated with the credibility intervals (default = 0.95).
`...`	Additional graphical parameters to customize the plot.

Details

The function simulates multiple datasets from the posterior distribution and compares the empirical distribution of total scores with the predicted distribution. It overlays the observed and predicted distributions using a histogram with transparency.

The fitted values, along with their credibility intervals, are computed and returned.

Value

A list containing:

- `items`: A table of fitted values and prediction intervals for each item.
- `subjects`: Fitted and observed mean scores per subject.
- `ysim`: Simulated response matrices.

The function also generates:

- A histogram comparing observed vs. predicted total scores.

References

Ramirez, E.S.; Jimenez, M.; Franco, V.R.; Alvarado, J.M. Delving into the Complexity of Analogical Reasoning: A Detailed Exploration with the Generalized Multicomponent Latent Trait Model for Diagnosis. *J. Intell.* 2024, 12, 67. https://doi.org/10.3390/jintelligence12070067

Examples

```
if (!requireNamespace("rstan", quietly = TRUE)) return()
data(analogy)
Q <- structure(
  c(0,0,1,0,1,0,1,0,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,0,1,0,1,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,1,0,1,1,0,1,0,0,0,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,
    1,0,0,0,1,1,0,1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,0,1),
  dim = c(27L, 5L),
  dimnames = list(NULL, c("rot_fig","rot_trap","reflection",
                          "subt_seg","mov_point")))
fit <- LLTM(analogy, Q, iters = 200, iter_warmup = 100, chains = 1)
ppchecks(fit)
```

---

print.enhanced_mltm_reliability
Print Method for Enhanced MLTM Reliability

Description

Print Method for Enhanced MLTM Reliability

Usage

```
## S3 method for class 'enhanced_mltm_reliability'
print(x, digits = 3, ...)
```

Arguments

x	An object of class "enhanced_mltm_reliability".
digits	Integer. Number of decimal places to display. Default is 3.
...	Currently unused.

Value

Invisibly returns x. Called for its side effect of printing the enhanced reliability analysis results to the console.

print.reliability_data_quality

Print Method for Data Quality Diagnostics

Description

Print Method for Data Quality Diagnostics

Usage

```
## S3 method for class 'reliability_data_quality'
print(x, ...)
```

Arguments

x	An object of class "reliability_data_quality".
...	Currently unused.

Value

Invisibly returns x. Called for its side effect of printing the data quality diagnostics to the console.

print.reliability_profile

Print Method for Reliability Profile

Description

Print Method for Reliability Profile

Usage

```
## S3 method for class 'reliability_profile'
print(x, ...)
```

Arguments

x	An object of class "reliability_profile".
...	Currently unused.

Value

Invisibly returns x. Called for its side effect of printing a formatted reliability profile summary to the console.

quick_reliability_check

Quick Reliability Check

Description

Ultra-fast reliability check for initial assessment.

Usage

```
quick_reliability_check(fit, n_samples_quick = 500)
```

Arguments

fit	Fitted model object.
n_samples_quick	Integer. Number of samples for quick analysis (default 500).

Value

Named vector of reliability estimates.

Examples

```
if (!requireNamespace("rstan", quietly = TRUE)) return()
data(analogy)
Q <- structure(
  c(0,0,1,0,1,0,1,0,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,0,1,0,1,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,1,0,1,1,0,1,0,0,0,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,
    1,0,0,0,1,1,0,1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,0,1),
  dim = c(27L, 5L),
  dimnames = list(NULL, c("rot_fig","rot_trap","reflection",
                          "subt_seg","mov_point")))
fit <- LLTM(analogy, Q, iters = 200, iter_warmup = 100, chains = 1)
quick_rel <- quick_reliability_check(fit)
print(quick_rel)
```

reliability *Marginal reliability*

Description

Estimate the the marginal reliability of the GMLTM.

Usage

```
reliability(fit)
```

Arguments

fit	MLTM object.

Details

`reliability` estimates a ...

Value

A number denoting the reliability estimate.

References

Ramírez, E.S.; Jiménez, M.; Franco, V.R.; Alvarado, J.M. Delving into the Complexity of Analogical Reasoning: A Detailed Exploration with the Generalized Multicomponent Latent Trait Model for Diagnosis. *J. Intell.* 2024, 12, 67. https://doi.org/10.3390/jintelligence12070067

reliability_profile *Quick Reliability Profile Analysis*

Description

Provides a quick overview of reliability characteristics for each component.

Usage

```
reliability_profile(cond_rel_obj)
```

Arguments

cond_rel_obj	An object of class `conditional_reliability_tif` returned by `conditional_reliability_tif`.

Value

A list of class "reliability_profile" with elements:

summary Data frame with columns theta, reliability, lower, and upper.

component Integer indicating which model component was analysed.

---

reliability_usage_instructions
Usage Instructions

Description

Prints detailed usage instructions.

Usage

```
reliability_usage_instructions()
```

Value

Invisibly returns NULL. Called for its side effect of printing step-by-step usage instructions to the console.

---

summary.conditional_reliability_tif
Summary Method for Conditional Reliability Analysis

Description

Summary Method for Conditional Reliability Analysis

Usage

```
## S3 method for class 'conditional_reliability_tif'
summary(object, digits = 3, ...)
```

Arguments

object	An object of class conditional_reliability_tif.
digits	Integer. Number of decimal places to display. Default is 3.
...	Currently unused.

Value

Invisibly returns object. Called for its side effect of printing a formatted summary of conditional reliability statistics to the console.

Index