Package: GMLTM (via r-universe)

October 26, 2026

Type Package

Title Generalized Multicomponent Latent Trait Model for Diagnosis

Version 2.0.0

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>), including a variant with correlated latent components. All models are estimated via Hamiltonian Monte Carlo using 'Stan' through the 'rstan' interface. Includes tools for prior predictive checks (Gelman et al., 2020), model validation, conditional reliability estimation, examinee mastery classification following Embretson (2019) <doi:10.1007/978-3-030-05584-4_9>, and individual diagnostic reports at the rule and component level. 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

Config/pak/sysreqs cmake make

Repository https://eduar-ramirez.r-universe.dev

Date/Publication 2026-08-27 10:37:29 UTC

RemoteUrl https://github.com/eduar-ramirez/gmltm-d

RemoteRef HEAD

RemoteSha 6e05e229178ffdb47b40bd137ae880a6e04f94f3

Contents

analogy . . . 3
check_reliability_data_quality . . . 4
compare_conditional_reliability . . . 5
compute_model_validation . . . 6
conditional_reliability_tif . . . 7
demo_reliability_analysis . . . 8
enhanced_mltm_reliability . . . 9
export_reliability_results . . . 10
extract_correlation . . . 11
generate_Q_with_interactions . . . 13
GMLTM . . . 15
GMLTM_corr . . . 17
integrate_with_enhanced_reliability . . . 19
LLTM . . . 21
marginal_Pchecks . . . 23
MLTM . . . 24
plot.conditional_reliability_tif . . . 26
plot.enhanced_mltm_reliability . . . 27
plot_all_components . . . 27
plot_components_comparison . . . 29
plot_conditional_reliability . . . 30
plot_ICC_grouped . . . 32
plot_ICC_individual . . . 33
plot_prior_predictive_check . . . 34
ppchecks . . . 35
print.enhanced_mltm_reliability . . . 37
print.GMLTM . . . 37
print.GMLTM_batch_report . . . 38
print.GMLTM_corr . . . 38
print.GMLTM_correlation . . . 39
print.GMLTM_prior_predictive_check . . . 39
print.GMLTM_student_report . . . 40
print.LLTM . . . 40
print.MLTM . . . 41

print.reliability_data_quality 41
print.reliability_profile 42
prior_predictive_check 42
quick_reliability_check 44
reliability 45
reliability_profile 46
reliability_usage_instructions 47
student_report 48
student_report_batch 51
summary.conditional_reliability_tif 53

Index **54**

`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

See Also

Other reliability functions: demo_reliability_analysis(), enhanced_mltm_reliability(), export_reliability_results(), quick_reliability_check(), reliability(), reliability_usage_instructions(

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
dq <- check_reliability_data_quality(fit)
print(dq)
```

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.

See Also

Other conditional reliability functions: conditional_reliability_tif(), integrate_with_enhanced_reliability(), plot_all_components(), plot_components_comparison(), plot_conditional_reliability(), reliability_profile()

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
cond_rel <- conditional_reliability_tif(fit, theta_range = seq(-2, 2, 0.5))
compare_conditional_reliability(cond_rel, theta_points = c(-1, 0, 1))
```

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. Models fitted with `GMLTM_corr` (correlated Σ), `GMLTM` ($\Sigma = \sigma^2 I$), `MLTM`, and `LLTM` can be freely mixed in the same list; each is labeled by model class in the output.

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, labeled by model type (e.g. `"GMLTM-D (Sigma libre)"` for `GMLTM_corr` fits, `"GMLTM-D (Sigma=I)"` for `GMLTM` fits).

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 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.

See Also

Other conditional reliability functions: `compare_conditional_reliability()`, `integrate_with_enhanced_reliabilit`
`plot_all_components()`, `plot_components_comparison()`, `plot_conditional_reliability()`,
`reliability_profile()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
cond_rel <- conditional_reliability_tif(fit, theta_range = seq(-2, 2, 0.5))
cond_rel$global$optimal_theta
```

---

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.

See Also

Other reliability functions: `check_reliability_data_quality()`, `enhanced_mltm_reliability()`, `export_reliability_results()`, `quick_reliability_check()`, `reliability()`, `reliability_usage_instructions(`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 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.

See Also

Other reliability functions: check_reliability_data_quality(), demo_reliability_analysis(), export_reliability_results(), quick_reliability_check(), reliability(), reliability_usage_instructions(

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
rel <- enhanced_mltm_reliability(fit, n_samples = 100)
rel$marginal_robust
```

---

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

See Also

Other reliability functions: check_reliability_data_quality(), demo_reliability_analysis(), enhanced_mltm_reliability(), quick_reliability_check(), reliability(), reliability_usage_instructions()

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
rel <- enhanced_mltm_reliability(fit, n_samples = 100)
export_reliability_results(rel)
```

---

extract_correlation *Extract and Summarize the Latent Correlation Matrix from a GMLTM_corr Fit*

Description

Extracts the posterior correlation matrix Σ among cognitive components from a model fitted with GMLTM_corr, together with pairwise credible intervals and a heatmap for visual inspection.

Usage

```
extract_correlation(fit, credible = 0.95, plot = TRUE)
```

Arguments

`fit`	An object of class `"GMLTM_corr"`, as returned by `GMLTM_corr`.
`credible`	Width of the credible interval for each pairwise correlation, e.g. `0.95` for a 95% interval. Default is `0.95`.
`plot`	Logical. If `TRUE` (default), draws a correlation heatmap of the posterior mean Σ on the active graphics device, using the same color scheme as the correlation diagnostics in `generate_Q_with_interactions`.

Details

The pairwise credible intervals are computed from the posterior draws of `Sigma` in `fit$fit`, extracted with `rstan::extract()` (not from `fit$EAP$Sigma`, which only holds the posterior mean). A pair of components is flagged as showing evidence of a real (non-zero) correlation when its credible interval excludes 0.

Value

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

`Sigma` The `M` x `M` posterior mean correlation matrix (`fit$EAP$Sigma`).

`pairs` A data.frame with one row per pair of components: `component_1`, `component_2`, `estimate`, `lower`, `upper`, and `excludes_zero` (logical).

`credible` The credible-interval width used.

`plot` A function that redraws the heatmap when called with no arguments, or `NULL` if `plot = FALSE`.

References

Embretson, S. E., & Yang, X. (2013). A multicomponent latent trait model for diagnosis. *Psychometrika*, **78**, 14–36. doi:10.1007/s113360129296y

See Also

Other diagnostic reporting functions: `student_report()`, `student_report_batch()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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_corr(analogy, Q, components,
```

```
                    iters = 200, iter_warmup = 100, chains = 1, cores = 1)
  corr <- extract_correlation(fit)
  corr$pairs
```

`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, family = "normal"), 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). `priors$alpha` additionally accepts `family`, either `"normal"` (default; a half-Normal, i.e. Normal truncated at 0) or `"lognormal"` (a Log-Normal prior on α, using the same `mu`/`sigma` as the location/scale of the underlying Normal on the log scale). 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 either a half-Normal prior (`priors$alpha$family = "normal"`, default) or a Log-Normal prior (`priors$alpha$family = "lognormal"`). 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. *Journal of Intelligence*, **12**, 67. doi:10.3390/jintelligence12070067

See Also

Other model fitting functions: `GMLTM_corr()`, `LLTM()`, `MLTM()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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, cores = 1)
fit$EAP$eta
reliability(fit)
```

GMLTM_corr *GMLTM-D with correlated latent components (general Sigma)*

Description

Variant of GMLTM that replaces the independent prior on θ ($\Sigma = \sigma^2 I$) with a multivariate Normal prior with a general correlation matrix Σ, estimated from the data via its Cholesky factor (non-centered parameterization with an LKJ prior on the correlation). This corresponds to innovation (iii) of the GMLTM-D as formulated in Ramirez et al. (2024): allowing correlations between components to reflect the natural interdependence between cognitive abilities, instead of assuming independent latent traits.

Usage

```
GMLTM_corr(
  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, family = "normal"), c = list(shape1 = 3, shape2 = 20)),
  lkj_eta = 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, alpha, and c. For Normal parameters supply mu and sigma; for the guessing parameter supply shape1 and shape2 (Beta prior). priors$alpha additionally accepts family, either "normal" (default; a half-Normal, i.e. Normal truncated at 0) or "lognormal" (a Log-Normal prior on α, using the same mu/sigma as the location/scale of the underlying Normal on the log scale). Unspecified elements retain defaults.
lkj_eta	Concentration parameter of the LKJ prior on the correlation matrix Σ. lkj_eta = 1 (default) is uniform over the space of correlation matrices; lkj_eta > 1 concentrates the prior towards $\Sigma = I$ (low correlations); lkj_eta < 1 towards more extreme correlations.
...	Additional arguments passed to rstan::sampling().

Details

GMLTM_corr estimates the same likelihood as GMLTM (three IRT parameters per item/component: discrimination α, guessing c, and rule difficulty η/β), but replaces the prior on the subject-level ability matrix θ with $\theta_i \sim N_M(0, \Sigma)$, where Σ is a correlation matrix among the M cognitive components. Σ is parameterized through its Cholesky factor L_Omega (a cholesky_factor_corr), with theta = t(L_Omega %*% theta_raw) and theta_raw ~ N(0, 1) i.i.d. (non-centered parameterization, which improves posterior geometry relative to sampling θ directly from the multivariate Normal). priors$theta is accepted for interface compatibility with GMLTM but is not used: the scale of θ is fixed to 1 by construction (diagonal of Σ), and its prior is governed entirely by lkj_eta.

Value

A list of class c("GMLTM_corr", "GMLTM") with elements:

EAP Posterior mean estimates: theta, alpha, eta, beta, guessing, and Sigma (the estimated M x M correlation matrix among components).

quantiles Posterior credible intervals for each parameter, including Sigma.

posterior Full posterior samples and derived quantities (including loglik, used by compute_model_validation).

fit The stanfit object from rstan::sampling.

data The original data matrix.

priors The prior hyperparameters used.

lkj_eta The LKJ concentration parameter 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. *Journal of Intelligence*, **12**, 67. doi:10.3390/jintelligence12070067

Embretson, S. E., & Yang, X. (2013). A multicomponent latent trait model for diagnosis. *Psychometrika*, **78**, 14–36. doi:10.1007/s113360129296y

Lewandowski, D., Kurowicka, D., & Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, 100(9), 1989–2001. doi:10.1016/j.jmva.2009.04.008

See Also

Other model fitting functions: `GMLTM()`, `LLTM()`, `MLTM()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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_corr(analogy, Q, components,
                  iters = 200, iter_warmup = 100, chains = 1, cores = 1)
fit$EAP$eta
fit$EAP$Sigma
```

`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.

See Also

Other conditional reliability functions: `compare_conditional_reliability()`, `conditional_reliability_tif()`, `plot_all_components()`, `plot_components_comparison()`, `plot_conditional_reliability()`, `reliability_profile()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
result <- integrate_with_enhanced_reliability(fit,
             theta_range = seq(-2, 2, 0.5))
result$conditional_tif
```

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. doi:10.1016/00016918(73)900036

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

See Also

Other model fitting functions: GMLTM(), GMLTM_corr(), MLTM()

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 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.

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

See Also

Other posterior predictive check functions: `ppchecks()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
marginal_Pchecks(fit)
```

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. doi:10.1007/s113360129296y

Embretson, S. E. (2019). Diagnostic modeling of skill hierarchies and cognitive processes with MLTM-D. In M. von Davier & Y.-S. Lee (Eds.), *Handbook of Diagnostic Classification Models* (pp. 185–208). Springer. doi:10.1007/9783030055844_9

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

See Also

Other model fitting functions: GMLTM(), GMLTM_corr(), LLTM()

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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, cores = 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.

See Also

Other conditional reliability functions: compare_conditional_reliability(), conditional_reliability_tif(), integrate_with_enhanced_reliability(), plot_components_comparison(), plot_conditional_reliability(), reliability_profile()

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
cond_rel <- conditional_reliability_tif(fit, theta_range = seq(-2, 2, 0.5))
plot_all_components(cond_rel, save_plots = TRUE, output_dir = tempdir())
```

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.

See Also

Other conditional reliability functions: compare_conditional_reliability(), conditional_reliability_tif(), integrate_with_enhanced_reliability(), plot_all_components(), plot_conditional_reliability(), reliability_profile()

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
cond_rel <- conditional_reliability_tif(fit, theta_range = seq(-2, 2, 0.5))
plot_components_comparison(cond_rel)
```

---

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.

See Also

Other conditional reliability functions: compare_conditional_reliability(), conditional_reliability_tif(), integrate_with_enhanced_reliability(), plot_all_components(), plot_components_comparison(), reliability_profile()

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
cond_rel <- conditional_reliability_tif(fit, theta_range = seq(-2, 2, 0.5))
plot_conditional_reliability(cond_rel, component = 1)
```

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: A detailed exploration with the Generalized Multicomponent Latent Trait Model for Diagnosis. *Journal of Intelligence*, **12**, 67. doi:10.3390/jintelligence12070067

See Also

Other item characteristic curve plots: `plot_ICC_individual()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 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.

See Also

Other item characteristic curve plots: `plot_ICC_grouped()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
plots <- plot_ICC_individual(fit, Q, components)
print(plots[[1]])
```

`plot_prior_predictive_check` *Plot a Prior Predictive Check for the GMLTM-D*

Description

Draws a histogram of the simulated proportion-correct distribution from `prior_predictive_check`, with configurable reference lines marking a substantively plausible range, in the style of Fig. 3 of Gelman et al. (2020): a prior that concentrates simulated data outside that range is a sign the prior is too informative, too diffuse, or otherwise misspecified.

Usage

```
plot_prior_predictive_check(
  x,
  type = c("global", "by_item", "by_component"),
  plausible_range = c(0.2, 0.9),
  bins = 30
)
```

Arguments

`x`	An object of class `"GMLTM_prior_predictive_check"`, as returned by `prior_predictive_check`.
`type`	Which simulated distribution to plot: `"global"` (default, one value per replicate), `"by_item"` (item-level values pooled across items and replicates), or `"by_component"` (component-level values pooled across components and replicates; requires `x` to have been computed with `by_component = TRUE`).

`plausible_range`
: Numeric vector of length 2 giving the substantively plausible range for the proportion correct, drawn as dashed reference lines. Default is `c(0.2, 0.9)`.

`bins`
: Number of histogram bins. Default is `30`.

Value

A `ggplot2` object.

See Also

Other prior predictive check functions: `prior_predictive_check()`

Examples

```
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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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))
ppc <- prior_predictive_check(Q, components, N = 150, S = 200)
plot_prior_predictive_check(ppc)
```

---

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. (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

See Also

Other posterior predictive check functions: `marginal_Pchecks()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 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.GMLTM *Print method for GMLTM objects*

Description

Print method for GMLTM objects

Usage

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

Arguments

x	An object of class "GMLTM", as returned by GMLTM.
digits	Number of decimal digits to display. Default is 3.
...	Currently ignored.

Value

Invisibly returns x.

print.GMLTM_batch_report *Print Method for GMLTM_batch_report Objects*

Description

Print Method for GMLTM_batch_report Objects

Usage

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

Arguments

x	An object of class "GMLTM_batch_report", as returned by student_report_batch.
digits	Number of decimal digits to display. Default is 3.
...	Currently ignored.

Value

Invisibly returns x.

print.GMLTM_corr *Print method for GMLTM_corr objects*

Description

Print method for GMLTM_corr objects

Usage

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

Arguments

x	An object of class c("GMLTM_corr", "GMLTM"), as returned by GMLTM_corr.
digits	Number of decimal digits to display. Default is 3.
...	Currently ignored.

Value

Invisibly returns x.

print.GMLTM_correlation

Print method for GMLTM_correlation objects

Description

Print method for GMLTM_correlation objects

Usage

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

Arguments

x	An object of class "GMLTM_correlation", as returned by extract_correlation.
digits	Number of decimal digits to display. Default is 3.
...	Currently ignored.

Value

Invisibly returns x.

print.GMLTM_prior_predictive_check

Print method for GMLTM_prior_predictive_check objects

Description

Print method for GMLTM_prior_predictive_check objects

Usage

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

Arguments

x	An object of class "GMLTM_prior_predictive_check", as returned by prior_predictive_check.
digits	Number of decimal digits to display. Default is 3.
...	Currently ignored.

Value

Invisibly returns x.

print.GMLTM_student_report *Print Method for GMLTM_student_report Objects*

Description

Print Method for GMLTM_student_report Objects

Usage

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

Arguments

x	An object of class "GMLTM_student_report", as returned by student_report.
digits	Number of decimal digits to display. Default is 3.
...	Currently ignored.

Value

Invisibly returns x.

print.LLTM *Print method for LLTM objects*

Description

Print method for LLTM objects

Usage

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

Arguments

x	An object of class "LLTM", as returned by LLTM.
digits	Number of decimal digits to display. Default is 3.
...	Currently ignored.

Value

Invisibly returns x.

print.MLTM *Print method for MLTM objects*

Description

Print method for MLTM objects

Usage

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

Arguments

x	An object of class "MLTM", as returned by MLTM.
digits	Number of decimal digits to display. Default is 3.
...	Currently ignored.

Value

Invisibly returns x.

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.

prior_predictive_check

Prior Predictive Check for the GMLTM-D

Description

Simulates parameters directly from the priors declared for GMLTM/GMLTM_corr, simulates response data from those parameters under the GMLTM-D's conjunctive structure, and returns the resulting distribution of simulated proportion-correct – globally, per item, and (optionally) per component – across replicates. This lets the analyst check whether a candidate prior implies substantively plausible data *before* spending compute on fitting the model to real data (Gelman et al., 2020, Sect. 2.4).

Usage

```
prior_predictive_check(
  Q,
  components,
  N,
  S = 1000,
 priors = list(theta = list(mu = 0, sigma = 1), eta = list(mu = 0, sigma = 1), alpha =
   list(mu = 0, sigma = 1, family = "normal"), c = list(shape1 = 3, shape2 = 20)),
  by_component = TRUE
)
```

Arguments

Q	A $p \times K$ Q-matrix specifying which cognitive rules each item requires, identical in role to the one passed to GMLTM.
components	A named list grouping rules into components, identical in role to the one passed to GMLTM.
N	Number of simulated examinees per replicate.
S	Number of replicates. Default is 1000, following the number of prior predictive draws used in Fig. 3 of Gelman et al. (2020); values from 500 to 1000 are typical.
priors	A named list of prior hyperparameters with the same structure accepted by GMLTM's priors argument (elements theta, eta, alpha, c), including priors$alpha$family ("normal" or "lognormal"). Parameters are simulated under exactly this prior, so the check reflects what will actually be fit – not an approximation of it.
by_component	Logical. If TRUE (default), also returns the simulated proportion-correct restricted to the items of each component.

Details

In each of the S replicates: rule difficulties η are drawn as $\eta_k \sim \text{Normal}(\mu_\eta, \sigma_\eta)$ for each rule of each component, and item difficulty is $\beta = Q\eta$; abilities θ are drawn i.i.d. $\text{Normal}(\mu_\theta, \sigma_\theta)$ for N simulated examinees; guessing c is drawn $\text{Beta}(\text{shape1}, \text{shape2})$ per item; and discrimination α is drawn from priors$alpha$family – half-Normal ($\text{Normal}(\mu_\alpha, \sigma_\alpha)$ truncated to $(0, \infty)$) or Log-Normal($\mu_\alpha, \sigma_\alpha$) – with one draw per unique combination of rules that a component's items require (the same grouping GMLTM itself uses, so alpha has as many free values here as it would have as a Stan parameter). Response probabilities follow the GMLTM-D's noncompensatory structure: for item i of examinee n, $p_{ni} = c_i + (1 - c_i) \prod_m \text{plogis}(\alpha_{im}(\theta_{nm} - \beta_{im}))^{C_{im}}$, and $y_{ni} \sim \text{Bernoulli}(p_{ni})$.

Use plot_prior_predictive_check to visualize the resulting distribution against a substantively plausible range.

Value

A list of class "GMLTM_prior_predictive_check" with elements:

global Numeric vector of length S: the overall simulated proportion correct in each replicate.

by_item An S x p matrix of simulated proportion correct per item.

by_component An S x M matrix of simulated proportion correct restricted to each component's items, or NULL if by_component = FALSE.

N, S The arguments used.

Q, components, priors The arguments used (priors after applying defaults).

References

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Burkner, P.-C., & Modrak, M. (2020). Bayesian workflow. *arXiv*. doi:10.48550/arXiv.2011.01808

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

See Also

Other prior predictive check functions: `plot_prior_predictive_check()`

Examples

```
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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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))

ppc_normal <- prior_predictive_check(Q, components, N = 150, S = 200)
summary(ppc_normal$global)

ppc_lognormal <- prior_predictive_check(
  Q, components, N = 150, S = 200,
  priors = list(alpha = list(mu = 0, sigma = 1, family = "lognormal"))
)
summary(ppc_lognormal$global)
```

`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.

See Also

Other reliability functions: check_reliability_data_quality(), demo_reliability_analysis(), enhanced_mltm_reliability(), export_reliability_results(), reliability(), reliability_usage_instructio

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 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`	A fitted model object of class `"GMLTM"`, `"MLTM"`, or `"LLTM"`, as returned by `GMLTM`, `MLTM`, or `LLTM`.

Details

`reliability` computes the classical marginal reliability coefficient for each cognitive component's ability estimates, from the posterior mean and posterior variance of theta across subjects: $r_{xx} = s^2/(s^2+e)$, where s^2 is the variance of the EAP theta estimates across subjects (true-score variance) and e is the average posterior variance of theta across subjects (error variance).

Value

A named numeric vector with one marginal reliability coefficient per cognitive component (length 1 for `LLTM` fits, which have a single ability dimension).

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

See Also

Other reliability functions: `check_reliability_data_quality()`, `demo_reliability_analysis()`, `enhanced_mltm_reliability()`, `export_reliability_results()`, `quick_reliability_check()`, `reliability_usage_instructions()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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, cores = 1)
reliability(fit)
```

`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.

See Also

Other conditional reliability functions: `compare_conditional_reliability()`, `conditional_reliability_tif()`, `integrate_with_enhanced_reliability()`, `plot_all_components()`, `plot_components_comparison()`, `plot_conditional_reliability()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
cond_rel <- conditional_reliability_tif(fit, theta_range = seq(-2, 2, 0.5))
reliability_profile(cond_rel)
```

---

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.

See Also

Other reliability functions: check_reliability_data_quality(), demo_reliability_analysis(), enhanced_mltm_reliability(), export_reliability_results(), quick_reliability_check(), reliability()

Examples

```
reliability_usage_instructions()
```

---

student_report *Individual Student Diagnostic Report for GMLTM-D Models*

Description

Builds a per-component and per-rule diagnostic report for a single examinee from a fitted GMLTM/GMLTM_corr model: posterior EAP and credible interval of θ, a mastery cutline γ_m per component and τ_{km} per rule, derived from a mastery probability y, binary mastery vectors, a decision-confidence index at both levels, and the difficulty of the rules/items involved in each component. Follows the mastery-diagnosis procedure for the MLTM-D described in Embretson (2019, Chap. 9, Sects. 9.2.2 and 9.4.2).

Usage

```
student_report(
  fit,
  Q,
  components,
  student_id,
  y = 0.5,
  theta_cut = NULL,
  credible = 0.95,
  digits = 3,
  include_plots = TRUE
)
```

Arguments

fit	An object of class "GMLTM" (as returned by GMLTM or GMLTM_corr).
Q	A $p \times K$ Q-matrix, identical to the one used to fit fit, specifying which rules each item requires.
components	A named list grouping rules into components, identical to the one used to fit fit.

student_id	Identifier of the examinee: either a row name of fit$data (character/numeric matching rownames(fit$data)) or an integer row index in 1:nrow(fit$data).
y	Mastery probability used to derive the cutlines γ_m and τ_{km} (Embretson, 2019, Sect. 9.2.2). Default is 0.50. Ignored for components overridden via theta_cut (τ_{km} is always derived from y; see Details).
theta_cut	Optional fixed cutline(s) on the θ scale, used instead of deriving γ_m from y. Either a single number (applied to every component) or a numeric vector of length length(components) (optionally named with the component names). Default is NULL (derive γ_m from y). Does not affect the rule-level cutlines τ_{km}.
credible	Width of the credible interval for θ, e.g. 0.95 for a 95% interval. Default is 0.95.
digits	Number of decimal digits used in the printed summary and in the plot subtitles. Default is 3.
include_plots	Logical. If TRUE (default), builds the ggplot2 objects described under plots below. If FALSE, plots is NULL and that (comparatively expensive) step is skipped, leaving only the numeric summary; this is used by student_report_batch to compute a lightweight summary for every requested student while only rendering plots for the current page.

Details

For each component m, the mastery cutline γ_m is the value of θ_m at which the mean predicted probability of solving the items that load on component m, $\bar{P}_m(\theta_m) = \text{mean}_k\,[c_k + (1 - c_k)\,\text{plogis}(\alpha_{km}(\theta_m - \beta_{km}))]$, equals y (Embretson, 2019, Eq. in Sect. 9.2.2). It is found numerically with uniroot using the item EAP estimates fitEAPalpha, fitEAPbeta, and fitEAPguessing. If theta_cut is supplied, it is used directly as γ_m and y is not used to derive it.

In addition, a rule-level cutline τ_{km} is computed for each rule k belonging to component m, using the same root-finding procedure as γ_m but restricted to the items that both load on component m and require rule k specifically (Embretson, 2019, Sect. 9.2.2). If no such item exists – the rule never appears on its own among the component's items – τ_{km} is undefined: a warning is issued and that rule is omitted from domain_rules and confidence_rules for that component. Rule-level cutlines are always derived from y and are not affected by theta_cut.

The student is classified as mastering component m (domain = 1) if $\theta_m^{EAP} \geq \gamma_m$, and as non-mastering (domain = 0) otherwise; the same comparison of θ_m^{EAP} against τ_{km} classifies mastery of rule k (domain_rules). Decision confidence is the proportion of posterior draws of θ_m for that student that agree with the classification: the proportion $\geq \gamma_m$ (or τ_{km}) when classified as mastery, or the proportion $< \gamma_m$ (or τ_{km}) when classified as non-mastery (Embretson, 2019, Sect. 9.4.2).

The difficulty of the rules assigned to each component is read from fitEAPeta (rule-level difficulty, on the θ scale), and the difficulty of the items that load on each component is read from fitEAPbeta (item-level difficulty). Q and components are required because fit does not store the item/component structure matrix C; it is recomputed internally the same way as in plot_ICC_individual.

Value

A list of class "GMLTM_student_report" with elements:

student The resolved identifier of the examinee (as it appears in rownames(fit$data)).

`theta` A list with `EAP`, `lower`, `upper` (named by component) and `credible`.

`cutline` A list with `y`, `gamma` (named by component), `tau` (named list by component of named numeric vectors of τ_{km} by rule, excluding any rule omitted for lack of items) and `fixed` (logical, `TRUE` if `theta_cut` was supplied).

`domain` Named binary vector (`1` = mastery, `0` = non-mastery) per component.

`domain_rules` Named list by component of named binary vectors (`1` = mastery, `0` = non-mastery) per rule, excluding any rule omitted for lack of items.

`confidence` Named decision-confidence index per component.

`confidence_rules` Named list by component of named decision-confidence indices per rule, same exclusions as `domain_rules`.

`difficulty` A list with `rules` (named list of rule-level η difficulties per component) and `items` (named list of item-level β difficulties per component).

`plots` Named list of `ggplot2` objects: one per component, showing the rule difficulty continuum, the mastery cutline, and the student's θ estimate with its credible interval (colored green for mastery, red for non-mastery), inspired by Figs. 4 and 5 of Embretson & Yang (2013); plus `badges`, a single scannable summary of every component and rule with a green/red mastery badge, complementing the continuum plots. `NULL` if `include_plots = FALSE`.

References

Embretson, S. E. (2019). Diagnostic modeling of skill hierarchies and cognitive processes with MLTM-D. In M. von Davier & Y.-S. Lee (Eds.), *Handbook of Diagnostic Classification Models* (pp. 185–208). Springer. doi:10.1007/9783030055844_9

Embretson, S. E., & Yang, X. (2013). A multicomponent latent trait model for diagnosis. *Psychometrika*, **78**, 14–36. doi:10.1007/s113360129296y

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

See Also

Other diagnostic reporting functions: `extract_correlation()`, `student_report_batch()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)
report <- student_report(fit, Q, components, student_id = 11)
report$domain
report$domain_rules
report$plots$global
report$plots$badges
```

`student_report_batch` *Paginated Batch Diagnostic Reports for GMLTM-D Models*

Description

Runs `student_report` over many examinees at once. A lightweight numeric summary (posterior EAP of θ, mastery and decision confidence per component and per rule) is always computed for every requested student, since it only reads quantities already stored in `fit$EAP`/`fit$posterior$theta`. The comparatively expensive step – building the `ggplot2` objects returned by `student_report` – is only performed for the students in the current page, so large batches stay cheap to page through.

Usage

```
student_report_batch(
  fit,
  Q,
  components,
  student_ids = NULL,
  offset = 0,
  limit = 10,
  y = 0.5,
  theta_cut = NULL,
  credible = 0.95,
  digits = 3
)
```

Arguments

`fit`	An object of class `"GMLTM"` (as returned by `GMLTM` or `GMLTM_corr`).
`Q`	A $p \times K$ Q-matrix, identical to the one used to fit `fit`, specifying which rules each item requires.
`components`	A named list grouping rules into components, identical to the one used to fit `fit`.
`student_ids`	Vector of examinee identifiers to include in the batch, in either form accepted by `student_report`'s `student_id` (row names of `fit$data`, or integer row indices). Default is `NULL`, meaning every student in `fit$data` (`seq_len(nrow(fit$data))`).
`offset`	Integer. Number of students in `student_ids` to skip before the current page. Default is `0`.

`limit` Integer. Maximum number of students in the current page (i.e. with full reports/plots generated). Default is `10`.

`y`, `theta_cut`, `credible`, `digits`
Passed through to `student_report`; see its documentation.

Details

The current page covers `student_ids[(offset + 1):(offset + limit)]`. If `offset` is at or beyond `length(student_ids)`, the page is empty (`reports` has length 0) rather than an error; `summary` is unaffected, since it always covers every requested student regardless of paging.

Value

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

`summary` A `data.frame` with one row per requested student (`student`, `theta_<component>`, `domain_<component>`, `confidence_<component>` for each component, and `domain_rule_<rule>`, `confidence_rule_<rule>` for each rule), built without generating any plots.

`reports` Named list (by resolved student label) of full `student_report` objects, including their `plots`, for the students in the current page only.

`page` A list with `offset`, `limit`, and `total` (the number of requested students in `student_ids`).

See Also

Other diagnostic reporting functions: `extract_correlation()`, `student_report()`

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,
    1,0,0,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,
    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,
    0,0,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,
    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, cores = 1)

# First page: students 1-10 (summary covers all 149 examinees either way)
page1 <- student_report_batch(fit, Q, components, limit = 10)
page1
page1$reports[["1"]]$plots$badges

# Next page: students 11-20, individual reports for students 1-10 are
# still fully available in page1$reports
page2 <- student_report_batch(fit, Q, components, offset = 10, limit = 10)
page2$reports[["11"]]
```

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

∗ **Q-matrix functions**
generate_Q_with_interactions, 13
∗ **conditional reliability functions**
compare_conditional_reliability, 5
conditional_reliability_tif, 7
integrate_with_enhanced_reliability, 19
plot_all_components, 27
plot_components_comparison, 29
plot_conditional_reliability, 30
reliability_profile, 46
∗ **datasets**
analogy, 3
∗ **diagnostic reporting functions**
extract_correlation, 11
student_report, 48
student_report_batch, 51
∗ **item characteristic curve plots**
plot_ICC_grouped, 32
plot_ICC_individual, 33
∗ **model fitting functions**
GMLTM, 15
GMLTM_corr, 17
LLTM, 21
MLTM, 24
∗ **model validation functions**
compute_model_validation, 6
∗ **posterior predictive check functions**
marginal_Pchecks, 23
ppchecks, 35
∗ **prior predictive check functions**
plot_prior_predictive_check, 34
prior_predictive_check, 42
∗ **reliability functions**
check_reliability_data_quality, 4
demo_reliability_analysis, 8
enhanced_mltm_reliability, 9
export_reliability_results, 10
quick_reliability_check, 44
reliability, 45
reliability_usage_instructions, 47

analogy, 3

check_reliability_data_quality, 4, *8*, *10, 11*, *45, 46*, *48*
compare_conditional_reliability, 5, *7*, *20*, *28, 29*, *31*, *47*
compute_model_validation, 6, *18*
conditional_reliability_tif, *5*, 7, *20*, *28, 29*, *31*, *47*

demo_reliability_analysis, *4*, 8, *10, 11*, *45, 46*, *48*

enhanced_mltm_reliability, *4*, *8*, 9, *11*, *20*, *45, 46*, *48*
export_reliability_results, *4*, *8*, 10, *10*, *45, 46*, *48*
extract_correlation, 11, *39*, *50*, *52*

generate_Q_with_interactions, *12*, 13
GMLTM, *6*, 15, *17–19*, *22*, *25*, *37*, *42, 43*, *45*, *48*, *51*
GMLTM_corr, *6*, *11, 12*, *16*, 17, *22*, *25*, *38*, *42*, *48*, *51*

integrate_with_enhanced_reliability, *5*, *7*, 19, *28, 29*, *31*, *47*

LLTM, *6*, *16*, *19*, 21, *25*, *40*, *45*

marginal_Pchecks, 23, *36*
MLTM, *6*, *16*, *19*, *22*, 24, *41*, *45*

plot.conditional_reliability_tif, 26
plot.enhanced_mltm_reliability, 27
plot_all_components, *5*, *7*, *20*, 27, *29*, *31*, *47*
plot_components_comparison, *5*, *7*, *20*, *28*, 29, *31*, *47*

plot_conditional_reliability, *5*, *7*, *20*, *28, 29*, 30, *47*
plot_ICC_grouped, 32, *33*
plot_ICC_individual, 33, *33*, *49*
plot_prior_predictive_check, 34, *43, 44*
ppchecks, *23*, 35
print.enhanced_mltm_reliability, 37
print.GMLTM, 37
print.GMLTM_batch_report, 38
print.GMLTM_corr, 38
print.GMLTM_correlation, 39
print.GMLTM_prior_predictive_check, 39
print.GMLTM_student_report, 40
print.LLTM, 40
print.MLTM, 41
print.reliability_data_quality, 41
print.reliability_profile, 42
prior_predictive_check, *34, 35*, *39*, 42

quick_reliability_check, *4*, *8*, *10, 11*, 44, *46*, *48*

reliability, *4*, *8*, *10, 11*, 45, *45*, *48*
reliability_profile, *5*, *7*, *20*, *28, 29*, *31*, 46
reliability_usage_instructions, *4*, *8*, *10, 11*, *45, 46*, 47

student_report, *12*, *40*, 48, *51, 52*
student_report_batch, *12*, *38*, *49, 50*, 51
summary.conditional_reliability_tif, 53

uniroot, *49*