Package: mglpa (via r-universe)

November 11, 2026

Type Package

Title Multi-Group Latent Profile Analysis and Tests of Profile Similarity

Version 0.0.0.9000

Description Estimates latent profile analysis (LPA) models with continuous indicators across known groups and implements the sequence of multi-group tests of profile similarity proposed by Morin, Meyer, Creusier and Bietry (2016) <doi:10.1177/1094428115621148>: class enumeration, configural, structural, dispersion, distributional, predictive and explanatory similarity. Estimation is carried out in R with an expectation-maximisation algorithm using random starts, with maximum-likelihood and robust (sandwich) standard errors, so that no external mixture-modelling software is required.

License MIT + file LICENSE

URL https://github.com/cahitgs/mglpa

BugReports https://github.com/cahitgs/mglpa/issues

Encoding UTF-8

LazyData true

Depends R (>= 4.1)

Imports stats, utils, parallel, ggplot2 (>= 3.5.0), rlang, Rcpp

LinkingTo Rcpp

Suggests testthat (>= 3.0.0), shiny (>= 1.7.0), bslib, DT, nnet, knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.1

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

Date/Publication 2026-11-11 22:01:40 UTC

RemoteUrl https://github.com/cahitgs/mglpa

RemoteRef HEAD

RemoteSha ea0c3c4f8df07c832e53e5d8aeb45e12871601cc

Contents

apa_table.mglpa_classification . . . 3
blrt . . . 3
classification_quality . . . 5
compare_fit . . . 6
contrast_estimates . . . 6
elbow_plot . . . 7
enumerate_profiles . . . 7
enumeration_blrt . . . 9
enumeration_table . . . 10
enumeration_vlmr . . . 10
explanatory_similarity . . . 11
fit_lpa . . . 12
fixed_from_fit . . . 14
mglpa_app . . . 15
mglpa_model . . . 15
mglpa_palettes . . . 17
mglpa_wm . . . 17
outcome_contrasts . . . 18
parameters . . . 19
posterior_probabilities . . . 19
predictive_similarity . . . 20
predictor_effects . . . 21
profile_means . . . 22
profile_plot . . . 22
profile_proportions . . . 23
reorder_profiles . . . 24
similarity_table . . . 24
similarity_tests . . . 25
simulate_lpa . . . 26
vlmr_test . . . 27
write_table_doc . . . 28

Index **30**

apa_table.mglpa_classification *APA-style tables of the results*

Description

Formatting helpers that turn the result objects of the package into publication-ready tables (data frames of character columns).

Usage

```
## S3 method for class 'mglpa_classification'
apa_table(x, digits = 3)

apa_table(x, digits = 3)

## S3 method for class 'mglpa_enumeration'
apa_table(x, digits = 3)

## S3 method for class 'mglpa_similarity'
apa_table(x, digits = 3)

## S3 method for class 'mglpa_predictive'
apa_table(x, digits = 3)

## S3 method for class 'mglpa_explanatory'
apa_table(x, digits = 3)
```

Arguments

x	A result object (`mglpa_enumeration`, `mglpa_similarity`, `mglpa_predictive`, `mglpa_explanatory` or `mglpa_classification`).
digits	Number of decimals.

Value

A data frame.

blrt *Bootstrap likelihood ratio test for k versus k - 1 profiles*

Description

Parametric bootstrap analogue of the Mplus TECH14 output. Data sets are generated from the (k - 1)-profile solution, the (k - 1)- and k-profile models are estimated on each of them, and the bootstrap distribution of the likelihood-ratio statistic 2(LL_k - LL_(k-1)) gives the p-value of the observed statistic: the proportion of bootstrap statistics at least as large as the observed one, as in Mplus. As in Mplus, the (k - 1)-profile model is started from the parent estimates and the k-profile model uses random starts.

Usage

```
blrt(
  fit_k,
  fit_km1,
  B = 100,
  starts = c(20, 4),
  seed = NULL,
  verbose = TRUE,
  cores = 1L,
  maxiter = NULL,
  tol = NULL
)
```

Arguments

fit_k, fit_km1	mglpa_fit objects for k and k - 1 profiles estimated on the same single-group data with the same variance specification.
B	Number of bootstrap draws.
starts	Random starts for the k-profile model in each draw (initial stage, final stage); at least two final-stage optimisations are needed to judge the replication of the best log-likelihood.
seed	Optional seed (for cores > 1 the seed initialises the parallel random number streams of the worker processes).
verbose	Logical; report progress.
cores	Number of CPU cores over which the bootstrap draws are distributed (needs the installed package, see fit_lpa()).
maxiter, tol	Iteration limit and convergence criterion of the EM algorithm in the bootstrap draws; by default those of fit_k.

Details

For every draw the convergence of both models and the number of final-stage optimisations of the k-profile model that replicated its best log-likelihood are recorded (element draws). Two kinds of draws can distort the p-value:

- the statistic of a draw is understated when the k-profile model stopped at a local maximum (best log-likelihood not replicated, estimation not converged, or a negative statistic, which is impossible at the global maximum of nested models). Such draws below the observed statistic

could have exceeded it; like Mplus, `blrt()` reports them in a warning and gives the largest p-value they could produce (`p_upper`);

- the statistic of a draw is overstated when the (k - 1)-profile model did not converge. Such draws above the observed statistic might not exceed it; `p_lower` is the smallest p-value they could produce.

Non-converged draws are included in the p-value (as in Mplus) and listed in the warnings; draws in which no finite log-likelihood was obtained are excluded and counted in `B - successful`. When `p`, `p_lower` and `p_upper` lead to the same conclusion the result is robust to these problems; otherwise increase the random starts of the k-profile model (`starts`) or the iteration limit (`maxiter`).

Value

A list of class `mglpa_blrt` with the observed `statistic`, the `bootstrap` statistics, a data frame `draws` with the per-draw statistic, convergence and replication record, the p-value `p` and its bounds `p_lower` and `p_upper`, the number of draws requested `B`, the number of `successful` draws, the numbers of draws flagged as understated (`n_flagged`), overstated (`n_flagged_high`) and non-converged (`n_not_converged`), and the `warnings`.

`classification_quality` *Classification quality of a profile solution*

Description

Entropy and the classification tables that Mplus prints under "Classification quality": the average posterior probabilities of profile membership of the observations assigned to each profile (most likely profile in rows, profiles in columns) and the probabilities of the most likely profile given the latent profile (profiles in rows, most likely profile in columns). For multi-group models the tables are computed within each group.

Usage

```
classification_quality(fit)
```

Arguments

`fit`	An `mglpa_fit` object.

Value

An object of class `mglpa_classification`: a list with `entropy` (as reported in `fit`), `entropy_profiles`, `entropy_by_group`, `counts` (most likely profile counts by group), `average_posterior` and `classification_probabiliti` (lists of matrices, one per group).

Examples

```
data(mglpa_wm)
ind <- c("verbal_wm", "visual_wm", "updating", "inhibition", "proc_speed")
fit <- fit_lpa(mglpa_wm, mglpa_model(3, ind, group = "group", equal_means = TRUE),
               starts = c(20, 4), seed = 1)
classification_quality(fit)
```

---

compare_fit *Compare two nested models with the two-out-of-three rule*

Description

Compare two nested models with the two-out-of-three rule

Usage

```
compare_fit(new, old)
```

Arguments

new, old	mglpa_fit objects; new is the more constrained model.

Value

A list with the differences in CAIC, BIC and ABIC (new - old), the number of criteria that decreased and the decision ("supported", "mixed" or "rejected").

---

contrast_estimates *Linear combinations of parameters with delta-method standard errors*

Description

Linear combinations of parameters with delta-method standard errors

Usage

```
contrast_estimates(fit, L, exponentiate = FALSE)
```

Arguments

fit	An mglpa_fit object.
L	A matrix with one row per contrast and one column per free parameter (named by label), or a named list of named numeric vectors giving the weights of the labels involved in each contrast.
exponentiate	Logical; also report exp() of the estimates and confidence limits (odds ratios).

Value

A data frame with estimates, standard errors, z statistics, p-values and 95% confidence limits.

`elbow_plot` *Elbow plot of the information criteria of a class enumeration*

Description

Elbow plot of the information criteria of a class enumeration

Usage

```
elbow_plot(x, indices = c("AIC", "BIC", "CAIC", "ABIC"))
```

Arguments

`x`	An `mglpa_enumeration` object.
`indices`	Which criteria to plot.

Value

A `ggplot` object.

`enumerate_profiles` *Class enumeration: profile solutions with 1 to K profiles in each group*

Description

Fits latent profile models with an increasing number of profiles, separately within each known group (step 0 of the multi-group similarity workflow), and collects the fit indices used to decide on the number of profiles. Consecutive solutions are compared with the Vuong-Lo-Mendell-Rubin likelihood ratio tests (`vlmr_test()`, Mplus `TECH11`); the bootstrap likelihood ratio test can be added with `enumeration_blrt()`.

Usage

```
enumerate_profiles(
  data,
  indicators,
  group = NULL,
  k = 1:8,
  variances = c("class_invariant", "class_varying"),
  starts = c(100, 10),
  stiterations = 10,
  maxiter = 500,
```

```
  tol = 1e-06,
  se = c("robust", "ml", "none"),
  order_profiles = c("level", "size", "none"),
  lrt = TRUE,
  group_levels = NULL,
  seed = NULL,
  verbose = TRUE,
  cores = 1L
)
```

Arguments

data	A data frame.
indicators	Character vector naming the profile indicators.
group	Optional name of the known-group variable. When NULL the model is estimated in a single group.
k	Integer vector of the numbers of profiles to estimate.
variances	"class_invariant" keeps the within-profile variances equal across profiles (the Mplus default); "class_varying" estimates a separate variance in each profile.
starts	Integer vector of length two: number of initial-stage random starts and number of final-stage optimisations (like STARTS in Mplus).
stiterations	Number of EM iterations in the initial stage.
maxiter	Maximum number of EM iterations in the final stage.
tol	Convergence criterion on the absolute change in log-likelihood.
se	Type of standard errors: "robust" (sandwich estimator with the observed information matrix, equivalent to ESTIMATOR = MLR), "ml" (inverse observed information) or "none".
order_profiles	How profiles are labelled after estimation: by average standardized level of the indicator means ("level", lowest first), by size ("size", largest first) or left as estimated ("none"). See reorder_profiles().
lrt	Logical; compute the VLMR and adjusted LMR tests for each solution against the solution with one profile less.
group_levels	Optional vector fixing the order of the group codes. Defaults to the sorted unique values found in the data.
seed	Optional random seed.
verbose	Logical; report progress.
cores	Number of CPU cores used to distribute the random starts (a PSOCK cluster from the parallel package). Requires the installed package; falls back to sequential estimation with a warning otherwise. Results depend on the random starts, not on the number of cores, but the assignment of starts to workers makes exact reproducibility require the same seed and cores.

Value

An object of class `mglpa_enumeration` with elements `fits` (a list by group of lists by `k` of `mglpa_fit` objects), `table` (one row per model with log-likelihood, free parameters, AIC, BIC, CAIC, ABIC, entropy, the likelihood ratio statistic and the VLMR and aLMR p-values, the smallest profile proportion, and the number of final-stage optimisations that replicated the best log-likelihood) and `vlmr` (the test objects).

Examples

```
data(mglpa_wm)
enum <- enumerate_profiles(mglpa_wm,
  indicators = c("verbal_wm", "visual_wm", "updating", "inhibition", "proc_speed"),
  group = "group", k = 1:4, starts = c(20, 4), seed = 1)
enum
```

`enumeration_blrt` *Add bootstrap likelihood ratio tests to a class enumeration*

Description

Runs `blrt()` for every consecutive pair of solutions in each group and adds a `BLRT_p` column to the enumeration table; the warnings of the tests are appended to the `notes` column.

Usage

```
enumeration_blrt(
  enum,
  B = 100,
  starts = c(20, 4),
  seed = NULL,
  verbose = TRUE,
  cores = 1L,
  maxiter = NULL,
  tol = NULL
)
```

Arguments

`enum`	An `mglpa_enumeration` object.
`B`	Number of bootstrap draws.
`starts`	Random starts for the k-profile model in each draw (initial stage, final stage); at least two final-stage optimisations are needed to judge the replication of the best log-likelihood.
`seed`	Optional seed (for `cores` > 1 the seed initialises the parallel random number streams of the worker processes).

verbose	Logical; report progress.
cores	Number of CPU cores over which the bootstrap draws are distributed (needs the installed package, see fit_lpa()).
maxiter, tol	Iteration limit and convergence criterion of the EM algorithm in the bootstrap draws; by default those of fit_k.

Value

The updated mglpa_enumeration object, with the test objects in the element blrt.

---

enumeration_table *Fit-index table of a class enumeration*

Description

Fit-index table of a class enumeration

Usage

```
enumeration_table(x)
```

Arguments

x	An mglpa_enumeration object.

Value

A data frame.

---

enumeration_vlmr *Add Vuong-Lo-Mendell-Rubin tests to a class enumeration*

Description

Runs vlmr_test() for every consecutive pair of solutions in each group and adds the columns LRT (twice the log-likelihood difference), VLMR_p and aLMR_p to the enumeration table. enumerate_profiles() calls this function by default (lrt = TRUE).

Usage

```
enumeration_vlmr(enum, variant = c("mplus", "vuong"))
```

Arguments

enum	An mglpa_enumeration object.
variant	"mplus" (sandwich covariance matrices, as in the Mplus TECH11 output) or "vuong" (inverse information matrices, Vuong's original matrix).

Value

The updated `mglpa_enumeration` object, with the test objects in the element `vlmr`.

`explanatory_similarity` *Explanatory similarity: profile-specific levels of distal outcomes across groups*

Description

Tests whether the profile-specific means of distal outcomes are the same across groups (step 6 of Morin et al., 2016). The profile solution, including the profile-membership logits, is held fixed at the estimates of a previously retained similarity model. For each residual-variance specification of the outcomes (one variance per outcome, or profile- and group-specific variances) a model with group-specific outcome means is compared with a model whose outcome means are equal across groups. Explanatory similarity is supported when at least two of CAIC, BIC and ABIC are lower for the constrained model. Pairwise differences between profiles (within groups) and between groups (within profiles) are obtained with the delta method.

Usage

```
explanatory_similarity(
  fit,
  outcomes,
  data = NULL,
  source = NULL,
  outcome_variances = c("both", "equal", "free"),
  starts = c(20, 5),
  stiterations = 10,
  maxiter = 500,
  tol = 1e-06,
  se = c("robust", "ml", "none"),
  digits = 5,
  seed = NULL,
  verbose = TRUE
)
```

Arguments

`fit`	An `mglpa_fit` whose profile solution is held fixed, or an `mglpa_similarity` object (its most constrained supported model is used unless `source` names another one).
`outcomes`	Character vector of outcome names.
`data`	Data frame; defaults to the data stored in `fit`.
`source`	For an `mglpa_similarity` object, the name of the model whose solution is held fixed.

outcome_variances
: "both", "equal" (one residual variance per outcome) or "free" (profile- and group-specific variances).

starts
: Integer vector of length two: number of initial-stage random starts and number of final-stage optimisations (like STARTS in Mplus).

stiterations
: Number of EM iterations in the initial stage.

maxiter
: Maximum number of EM iterations in the final stage.

tol
: Convergence criterion on the absolute change in log-likelihood.

se
: Type of standard errors: "robust" (sandwich estimator with the observed information matrix, equivalent to ESTIMATOR = MLR), "ml" (inverse observed information) or "none".

digits
: Rounding applied to the fixed parameter values (Mplus writes start values with five decimals); NULL keeps full precision.

seed
: Optional random seed.

verbose
: Logical; print progress.

Value

An object of class mglpa_explanatory with elements fits (named free_means_equal_var, equal_means_equal_var, free_means_free_var, equal_means_free_var), table, decisions, means (outcome means by profile and group, with confidence limits) and contrasts (pairwise differences), the last two by model.

---

fit_lpa *Fit a latent profile model by maximum likelihood*

Description

Estimates the model described by mglpa_model() with an expectation-maximisation (EM) algorithm. Profile indicators and distal outcomes are conditionally independent normal variables given the profile (a diagonal Gaussian mixture), profile membership follows a multinomial logistic model with optional predictors and known groups, and cross-group equality constraints are handled by pooling the sufficient statistics of the parameters that share a label. Missing indicator values are handled by full-information maximum likelihood (each observation contributes the density of its observed indicators). The procedure mirrors the random-start strategy of Mplus: starts[1] random starts are run for stiterations iterations, the best starts[2] are iterated to convergence and the highest log-likelihood is retained.

Usage

```
fit_lpa(
  data,
  model,
  starts = c(100, 10),
  stiterations = 10,
```

```
  maxiter = 500,
  tol = 1e-06,
  fixed = NULL,
  start_values = NULL,
  se = c("robust", "ml", "none"),
  mplus_convention = TRUE,
  seed = NULL,
  verbose = FALSE,
  engine = c("auto", "r"),
  cores = 1L
)
```

Arguments

`data`	A data frame.
`model`	An `mglpa_model` object.
`starts`	Integer vector of length two: number of initial-stage random starts and number of final-stage optimisations (like `STARTS` in Mplus).
`stiterations`	Number of EM iterations in the initial stage.
`maxiter`	Maximum number of EM iterations in the final stage.
`tol`	Convergence criterion on the absolute change in log-likelihood.
`fixed`	Optional named numeric vector of parameter labels and the values at which they are fixed; see `fixed_from_fit()`.
`start_values`	Optional named numeric vector (parameter labels) or `mglpa_fit` whose estimates are used as an additional, unperturbed start.
`se`	Type of standard errors: `"robust"` (sandwich estimator with the observed information matrix, equivalent to `ESTIMATOR = MLR`), `"ml"` (inverse observed information) or `"none"`.
`mplus_convention`	Logical. When `TRUE` (the default) the log-likelihood includes the contribution of the known-group proportions and these proportions are counted as free parameters, which reproduces the log-likelihood and information criteria reported by Mplus for `KNOWNCLASS` models.
`seed`	Optional random seed.
`verbose`	Logical; print progress.
`engine`	`"auto"` runs the EM iterations in compiled C++ code whenever the model allows it (no predictors of profile membership and free membership logits) and in R otherwise; `"r"` forces the pure-R implementation, which gives identical results and is kept as a reference.
`cores`	Number of CPU cores used to distribute the random starts (a PSOCK cluster from the `parallel` package). Requires the installed package; falls back to sequential estimation with a warning otherwise. Results depend on the random starts, not on the number of cores, but the assignment of starts to workers makes exact reproducibility require the same `seed` and `cores`.

Value

An object of class `mglpa_fit`.

Examples

```
data(mglpa_wm)
m <- mglpa_model(3, c("verbal_wm", "visual_wm", "updating", "inhibition",
                      "proc_speed"))
fit <- fit_lpa(mglpa_wm[mglpa_wm$group == 1, ], m, starts = c(20, 4), seed = 1)
fit
```

`fixed_from_fit` *Extract parameter values from a fit to fix them in a later model*

Description

Returns a named vector of estimates (named by parameter label) that can be passed to the `fixed` argument of `fit_lpa()`. This is how the predictive and explanatory similarity models hold the profile solution constant at the values obtained in an earlier step, in the spirit of the start-value approach used with Mplus.

Usage

```
fixed_from_fit(fit, types = c("mean", "variance"), digits = NULL)
```

Arguments

`fit`	An `mglpa_fit` object.
`types`	Parameter types to extract: any of `"mean"`, `"variance"`, `"intercept"` (profile-membership logits) and `"slope"`.
`digits`	Optional rounding of the extracted values (Mplus writes start values with five decimals). `NULL` keeps full precision.

Value

A named numeric vector.

mglpa_app *Launch the MG-LPA Shiny application*

Description

A graphical interface to the full workflow of the package: data import, class enumeration, the four similarity tests, profile plots, predictive and explanatory similarity, APA-style tables and project files. All estimation is done in R by `fit_lpa()`; no external software is needed.

Usage

```
mglpa_app(data = NULL, ...)
```

Arguments

`data`	Optional data frame loaded when the application starts.
`...`	Passed to `shiny::runApp()`, e.g. `launch.browser`.

Value

Called for its side effect (runs the application).

Examples

```
if (interactive()) mglpa_app()
```

mglpa_model *Specify a (multi-group) latent profile model*

Description

Describes a latent profile analysis (LPA) model with continuous indicators that is estimated in one sample or across known groups. The equality constraints across groups correspond to the similarity tests of Morin, Meyer, Creusier and Bietry (2016): the configural model imposes none, the structural model adds equal profile means, the dispersion model adds equal within-profile variances and the distributional model adds equal profile proportions. The model can be extended with predictors of profile membership (multinomial logistic regression; predictive similarity) and with distal outcomes modelled as additional indicators (explanatory similarity).

Usage

```
mglpa_model(
  k,
  indicators,
  group = NULL,
  variances = c("class_invariant", "class_varying"),
  equal_means = FALSE,
  equal_variances = FALSE,
  equal_proportions = FALSE,
  predictors = NULL,
  predictor_effects = c("free", "equal"),
  outcomes = NULL,
  outcome_means = c("free", "equal"),
  outcome_variances = c("equal", "free"),
  group_levels = NULL
)
```

Arguments

k	Number of latent profiles.
indicators	Character vector naming the profile indicators.
group	Optional name of the known-group variable. When NULL the model is estimated in a single group.
variances	"class_invariant" keeps the within-profile variances equal across profiles (the Mplus default); "class_varying" estimates a separate variance in each profile.
equal_means, equal_variances, equal_proportions	Logical. Constrain the indicator means, the indicator variances or the profile proportions to be equal across groups.
predictors	Optional character vector of predictors of profile membership.
predictor_effects	"free" estimates group-specific multinomial regression coefficients; "equal" constrains them to equality across groups.
outcomes	Optional character vector of distal outcomes.
outcome_means	"free" estimates group- and profile-specific outcome means; "equal" constrains them to equality across groups.
outcome_variances	"equal" estimates one residual variance per outcome; "free" estimates a profile- and group-specific variance.
group_levels	Optional vector fixing the order of the group codes. Defaults to the sorted unique values found in the data.

Value

An object of class mglpa_model.

References

Morin, A. J. S., Meyer, J. P., Creusier, J., & Bietry, F. (2016). Multiple-group analysis of similarity in latent profile solutions. *Organizational Research Methods, 19*(2), 231-254.

Examples

```
mglpa_model(3, c("verbal_wm", "visual_wm", "updating"), group = "group",
            equal_means = TRUE)
```

`mglpa_palettes` *Colour palettes for profile plots*

Description

The nine palettes of the MG-LPA Analyzer web application.

Usage

```
mglpa_palettes
```

Format

A named list of character vectors of hexadecimal colours.

`mglpa_wm` *Simulated working-memory data set (two age groups)*

Description

The simulated data set distributed with the MG-LPA Analyzer: 1,000 participants (500 younger and 500 older adults) with five standardized working-memory indicators, two predictors of profile membership and two distal outcomes. The Mplus results reported for this data set in the accompanying article are reproduced by the estimation functions of this package.

Usage

```
mglpa_wm
```

Format

A data frame with 1,000 rows and 11 variables:

id Participant identifier.

group Age group: 1 = younger adults, 2 = older adults.

verbal_wm, visual_wm, updating, inhibition, proc_speed Profile indicators (z scores).

education Years of education (predictor).

cog_reserve Cognitive reserve (z score; predictor).

task_perf Task performance, percent correct (outcome).

everyday_func Everyday functioning, 0-100 scale (outcome).

Source

Marangoz, M. C., Cokluk-Bokeoglu, O., & Morin, A. J. S. (2026). MG-LPA Analyzer (Version 1.0.0). doi:10.5281/zenodo.22698717

`outcome_contrasts` *Pairwise differences in outcome means between profiles and between groups*

Description

Pairwise differences in outcome means between profiles and between groups

Usage

```
outcome_contrasts(fit)
```

Arguments

`fit`	An `mglpa_fit` with distal outcomes.

Value

A data frame of delta-method contrasts: differences between pairs of profiles within each group (or `"all"` when outcome means are equal across groups) and, when outcome means are group-specific, differences between pairs of groups within each profile.

parameters *Parameter estimates of a fitted profile model*

Description

Parameter estimates of a fitted profile model

Usage

```
parameters(object, type = NULL, ...)
```

Arguments

object	An mglpa_fit object.
type	Optional parameter types to keep.
...	Unused.

Value

A data frame with one row per model parameter (free or fixed).

posterior_probabilities *Posterior profile probabilities and most likely profile*

Description

Posterior profile probabilities and most likely profile

Usage

```
posterior_probabilities(fit)
```

Arguments

fit	An mglpa_fit object.

Value

A data frame with the posterior probabilities of each profile and the most likely profile for each observation used in the analysis.

predictive_similarity *Predictive similarity: predictors of profile membership across groups*

Description

Tests whether the multinomial logistic relations between predictors and profile membership are the same across groups (step 5 of Morin et al., 2016). The profile solution (indicator means and variances) is held fixed at the estimates of a previously retained similarity model, as in the start-value approach used with Mplus, and two models are compared: one with group-specific predictor effects and one with effects constrained to be equal across groups. Predictive similarity is supported when at least two of CAIC, BIC and ABIC are lower for the constrained model.

Usage

```
predictive_similarity(
  fit,
  predictors,
  data = NULL,
  source = NULL,
  effects = c("both", "free", "equal"),
  starts = c(20, 5),
  stiterations = 10,
  maxiter = 500,
  tol = 1e-06,
  se = c("robust", "ml", "none"),
  digits = 5,
  seed = NULL,
  verbose = TRUE
)
```

Arguments

fit	An mglpa_fit whose profile solution is held fixed, or an mglpa_similarity object (its most constrained supported model is used unless source names another one).
predictors	Character vector of predictor names.
data	Data frame; defaults to the data stored in fit.
source	For an mglpa_similarity object, the name of the model whose solution is held fixed.
effects	Which models to estimate: "both", "free" or "equal".
starts	Integer vector of length two: number of initial-stage random starts and number of final-stage optimisations (like STARTS in Mplus).
stiterations	Number of EM iterations in the initial stage.
maxiter	Maximum number of EM iterations in the final stage.
tol	Convergence criterion on the absolute change in log-likelihood.

se	Type of standard errors: "robust" (sandwich estimator with the observed information matrix, equivalent to ESTIMATOR = MLR), "ml" (inverse observed information) or "none".
digits	Rounding applied to the fixed parameter values (Mplus writes start values with five decimals); NULL keeps full precision.
seed	Optional random seed.
verbose	Logical; print progress.

Value

An object of class mglpa_predictive with elements fits (free and/or equal), table (fit indices and decision), effects (regression coefficients and odds ratios for all pairwise profile comparisons, by model) and source (the fixed profile solution).

---

predictor_effects *Multinomial regression effects of the predictors for all profile comparisons*

Description

Multinomial regression effects of the predictors for all profile comparisons

Usage

```
predictor_effects(fit)
```

Arguments

fit	An mglpa_fit with predictors of profile membership.

Value

A data frame with, for every group (or "all" when the effects are equal across groups), predictor and pair of profiles, the logit coefficient (profile vs. comparison profile), its standard error, z statistic, p-value, and the odds ratio with 95% confidence limits.

profile_means *Profile means in raw and standardized metric*

Description

Returns the estimated indicator (and outcome) means of each profile, with standard errors, and their standardized counterparts `z = (mean - sample mean) / sample SD`, where the sample statistics are computed on the data used in the analysis (pooled across groups by default, as in the profile plots of the MG-LPA Analyzer).

Usage

```
profile_means(fit, standardized = TRUE, standardize_by = c("pooled", "group"))
```

Arguments

`fit`	An `mglpa_fit` object.
`standardized`	Logical; add standardized means.
`standardize_by`	`"pooled"` uses the sample statistics of the whole analysed sample; `"group"` standardizes within each known group.

Value

A data frame with columns `group`, `group_level`, `class`, `variable`, `mean`, `se`, `ci_lower`, `ci_upper`, `variance`, `z` and `proportion` (model-based profile proportion in the group).

profile_plot *Profile plot*

Description

Plots the profile-specific means of the indicators, standardized with the sample mean and standard deviation of each indicator (z scores) or in their original metric. In multi-group models one panel per group is drawn when the plotted means differ between groups (the configural model); when they are constrained to equality (structural, dispersion and distributional models) a single panel is drawn and the group-specific profile proportions are reported in the labels.

Usage

```
profile_plot(
  fit,
  type = c("bar", "line", "radar"),
  standardized = TRUE,
  variables = NULL,
  palette = "vibrant",
```

```
  ylim = NULL,
  title = NULL,
  show_proportions = TRUE,
  standardize_by = "pooled",
  facet = c("auto", "group", "none")
)
```

Arguments

fit	An mglpa_fit object.
type	"bar" (profiles on the x axis, one bar per indicator), "line" (indicators on the x axis, one line per profile) or "radar".
standardized	Logical; plot z scores (default) or raw means.
variables	Variables to plot; defaults to the profile indicators.
palette	Name of a palette in mglpa_palettes or a vector of colours.
ylim	Optional limits of the value axis (radial axis for "radar").
title	Optional plot title.
show_proportions	Logical; report the model-based profile proportions in the labels.
standardize_by	Passed to profile_means().
facet	"auto" (one panel per group only when the plotted means differ between groups), "group" (always one panel per group) or "none" (a single panel showing the means of the first group).

Value

A ggplot object.

---

profile_proportions *Profile proportions by group*

Description

Profile proportions by group

Usage

```
profile_proportions(fit, type = c("model", "posterior", "most_likely"))
```

Arguments

fit	An mglpa_fit object.
type	"model" (model-implied), "posterior" (average posterior probabilities) or "most_likely" (modal assignment).

Value

A matrix with one row per group and one column per profile.

---

reorder_profiles *Reorder (relabel) the profiles of a fitted model*

Description

Latent profiles are only identified up to a permutation of their labels. This function relabels the profiles of a fitted model, either with an explicit permutation or automatically by the average standardized level of the indicator means (`"level"`, lowest profile first) or by profile size (`"size"`, largest first), and refits the model from the relabelled solution so that all estimates, standard errors and posterior probabilities are consistent with the new labels. The log-likelihood is unchanged.

Usage

```
reorder_profiles(fit, order = "level")
```

Arguments

`fit`	An `mglpa_fit` object.
`order`	Either `"level"`, `"size"`, or an integer permutation of `1:k` giving, for each new profile label, the old label it takes over. For models whose profile means are group-specific (the configural model), a list with one permutation per group may be given; the automatic rules are then applied within each group.

Value

An `mglpa_fit` object.

---

similarity_table *Fit-index table of the similarity tests*

Description

Fit-index table of the similarity tests

Usage

```
similarity_table(x)
```

Arguments

`x`	An `mglpa_similarity` object.

Value

A data frame.

`similarity_tests` *Multi-group tests of profile similarity*

Description

Fits the sequence of configural, structural, dispersion and distributional similarity models of Morin et al. (2016) for a given number of profiles and applies the decision rule used in the MG-LPA Analyzer: similarity at a given level is supported when at least two of the three information criteria (CAIC, BIC and ABIC) decrease relative to the previous, less constrained model; it is rejected when none decreases; otherwise the evidence is mixed.

Usage

```
similarity_tests(
  data,
  indicators,
  group,
  k,
  variances = c("class_invariant", "class_varying"),
  models = c("configural", "structural", "dispersion", "distributional"),
  starts = c(100, 10),
  stiterations = 10,
  maxiter = 500,
  tol = 1e-06,
  se = c("robust", "ml", "none"),
  order_profiles = c("level", "size", "none"),
  group_levels = NULL,
  seed = NULL,
  verbose = TRUE,
  cores = 1L
)
```

Arguments

`data`	A data frame.
`indicators`	Character vector naming the profile indicators.
`group`	Name of the known-group variable.
`k`	Number of profiles retained in every group at the enumeration stage.
`variances`	`"class_invariant"` keeps the within-profile variances equal across profiles (the Mplus default); `"class_varying"` estimates a separate variance in each profile.
`models`	Which models to estimate, in order.
`starts`	Integer vector of length two: number of initial-stage random starts and number of final-stage optimisations (like `STARTS` in Mplus).
`stiterations`	Number of EM iterations in the initial stage.

maxiter	Maximum number of EM iterations in the final stage.
tol	Convergence criterion on the absolute change in log-likelihood.
se	Type of standard errors: "robust" (sandwich estimator with the observed information matrix, equivalent to ESTIMATOR = MLR), "ml" (inverse observed information) or "none".
order_profiles	How profiles are labelled after estimation: by average standardized level of the indicator means ("level", lowest first), by size ("size", largest first) or left as estimated ("none"). See reorder_profiles().
group_levels	Optional vector fixing the order of the group codes. Defaults to the sorted unique values found in the data.
seed	Optional random seed.
verbose	Logical; report progress.
cores	Number of CPU cores used to distribute the random starts (a PSOCK cluster from the parallel package). Requires the installed package; falls back to sequential estimation with a warning otherwise. Results depend on the random starts, not on the number of cores, but the assignment of starts to workers makes exact reproducibility require the same seed and cores.

Value

An object of class mglpa_similarity with elements fits (named list of mglpa_fit objects), table (fit indices, differences and decisions), retained (name of the most constrained model supported by the sequence) and data.

Examples

```
data(mglpa_wm)
sim <- similarity_tests(mglpa_wm,
  indicators = c("verbal_wm", "visual_wm", "updating", "inhibition", "proc_speed"),
  group = "group", k = 3, starts = c(20, 4), seed = 1)
sim
```

---

simulate_lpa *Simulate data from a fitted single-group profile model*

Description

Draws profile membership from the estimated profile proportions and the indicators (and outcomes) from the profile-specific normal distributions. Predictors of profile membership are not supported.

Usage

```
simulate_lpa(fit, n = fit$data$N)
```

Arguments

`fit`	An `mglpa_fit` estimated in a single group without predictors.
`n`	Number of observations; defaults to the sample size of the fit.

Value

A data frame with the indicator (and outcome) columns and a `profile` column with the generating profile.

`vlmr_test` *Vuong-Lo-Mendell-Rubin likelihood ratio tests for the number of profiles*

Description

Compares a k-profile solution with the (k - 1)-profile solution estimated on the same data with the Vuong-Lo-Mendell-Rubin likelihood ratio test (VLMR; Lo, Mendell and Rubin, 2001) and its adjusted version (aLMR), the two tests reported in the Mplus `TECH11` output.

Usage

```
vlmr_test(fit_k, fit_km1, variant = c("mplus", "vuong"))
```

Arguments

`fit_k`, `fit_km1`	`mglpa_fit` objects with k and k - 1 profiles estimated on the same single-group data with the same variance specification.
`variant`	`"mplus"` (sandwich covariance matrices, as in the Mplus `TECH11` output) or `"vuong"` (inverse information matrices, Vuong's original matrix).

Details

The test statistic is $2(LL_k - LL_{k-1})$. Following Vuong (1989) and Lo et al. (2001), its null distribution is approximated by a weighted sum of independent chi-square variables with one degree of freedom whose weights are the eigenvalues of the matrix

$$W = \begin{pmatrix} C_k V_k & C_{k,k-1} V_{k-1} \\ -C_{k-1,k} V_k & -C_{k-1} V_{k-1} \end{pmatrix},$$

where the C matrices are cross-products of the per-observation score vectors of the two models at their maximum likelihood estimates and the V matrices are covariance matrices of the parameter estimates. Vuong's derivation uses the inverse observed information matrices (`variant = "vuong"`, the implementation of LatentGOLD). Mplus uses the robust sandwich covariance matrices instead (Vermunt, 2024); `variant = "mplus"`, the default, reproduces the `TECH11` output of Mplus (statistic, mean, standard deviation and p-values of the VLMR and aLMR tests). The mean of the reference distribution is the sum of the eigenvalues, its standard deviation the square root of twice the sum of their squares, and p-values are computed with the method of Imhof (1961). The adjusted statistic

of Lo et al. (2001) divides the statistic by $1 + 1/(d \log N)$, where d is the difference in the number of free parameters and N the sample size, and uses the same reference distribution.

As in Mplus, the tests are available for models estimated in a single group without predictors; the two fits must differ only in the number of profiles. Jeffries (2003) showed that the derivation of Lo et al. is not rigorous, so these p-values are best used together with the bootstrap likelihood ratio test (`blrt()`) and the information criteria.

Value

An object of class `mglpa_vlmr`: a list with the number of profiles `k`, the `statistic`, the difference in the number of parameters `df`, the sample size `N`, the two log-likelihoods, the `mean`, `sd` and `p` of the VLMR test, the `adjusted` statistic and `adjusted_p` of the aLMR test, and the `eigenvalues` of the weight matrix.

References

Imhof, J. P. (1961). Computing the distribution of quadratic forms in normal variables. *Biometrika, 48*, 419-426.

Jeffries, N. O. (2003). A note on 'Testing the number of components in a normal mixture'. *Biometrika, 90*, 991-994.

Lo, Y., Mendell, N. R., & Rubin, D. B. (2001). Testing the number of components in a normal mixture. *Biometrika, 88*, 767-778.

Vermunt, J. K. (2024). The Vuong-Lo-Mendell-Rubin test for latent class and latent profile analysis: A note on the different implementations in Mplus and LatentGOLD. *Methodology, 20*(1), Article e12467.

Vuong, Q. H. (1989). Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica, 57*, 307-333.

Examples

```
data(mglpa_wm)
ind <- c("verbal_wm", "visual_wm", "updating", "inhibition", "proc_speed")
d1 <- mglpa_wm[mglpa_wm$group == 1, ]
f2 <- fit_lpa(d1, mglpa_model(2, ind), starts = c(20, 4), seed = 1)
f3 <- fit_lpa(d1, mglpa_model(3, ind), starts = c(20, 4), seed = 1)
vlmr_test(f3, f2)
```

---

write_table_doc *Write a table as a Word-compatible document*

Description

Writes the table as an HTML file with a `.doc` extension that Microsoft Word and LibreOffice open directly, with APA-style formatting (Times New Roman, horizontal rules only). No additional software is required.

Usage

```
write_table_doc(
  table,
  file,
  title = NULL,
  note = attr(table, "note"),
  landscape = TRUE
)
```

Arguments

table	A data frame, e.g. from apa_table().
file	Output file name (.doc).
title	Table title (printed in italics above the table).
note	Table note; defaults to the note attribute of table.
landscape	Logical; landscape page orientation.

Value

The file name, invisibly.

Index

∗ **datasets**
    mglpa_wm, 17

apa_table
        *(*apa_table.mglpa_classification*)*, 3
apa_table(), *29*
apa_table.mglpa_classification, 3
apa_tables
        *(*apa_table.mglpa_classification*)*, 3

blrt, 3
blrt(), *9*, *28*

classification_quality, 5
compare_fit, 6
contrast_estimates, 6

elbow_plot, 7
enumerate_profiles, 7
enumerate_profiles(), *10*
enumeration_blrt, 9
enumeration_blrt(), *7*
enumeration_table, 10
enumeration_vlmr, 10
explanatory_similarity, 11

fit_lpa, 12
fit_lpa(), *4*, *10*, *14*, *15*
fixed_from_fit, 14
fixed_from_fit(), *13*

mglpa_app, 15
mglpa_model, 15
mglpa_model(), *12*
mglpa_palettes, 17, *23*
mglpa_wm, 17

outcome_contrasts, 18

parameters, 19
posterior_probabilities, 19
predictive_similarity, 20
predictor_effects, 21
profile_means, 22
profile_means(), *23*
profile_plot, 22
profile_proportions, 23

reorder_profiles, 24
reorder_profiles(), *8*, *26*

shiny::runApp(), *15*
similarity_table, 24
similarity_tests, 25
simulate_lpa, 26

vlmr_test, 27
vlmr_test(), *7*, *10*

write_table_doc, 28