Package 'ordpp'

November 14, 2026

Type Package

Title Projection Pursuit with Monotone Scoring for Ordered Categorical Data

Version 0.1.0

Date 2026-10-06

Description Exploratory projection pursuit for ordered categorical (ordinal) items. Monotone category scores and a projection direction are estimated jointly by maximizing a weighted characteristic-function discrepancy between the projected joint distribution and the projection implied by item independence with the observed marginal distributions. Fitted category dictionaries and loadings can be frozen and evaluated on held-out data, with a finite-sample-valid held-out permutation test, a refitted permutation test, a variance/shape decomposition of the index and bootstrap stability summaries.

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, graphics

URL https://github.com/dapr12/ordpp

BugReports https://github.com/dapr12/ordpp/issues

NeedsCompilation no

Author Diego Andres Perez Ruiz [aut, cre]

Maintainer Diego Andres Perez Ruiz <diego.perezruiz@manchester.ac.uk>

Repository CRAN

Date/Publication 2026-11-14 08:00:07 UTC

Contents

as_ordpp_map 2
decompose_ordpp 3

heldout_test_ordpp 3
ordpp 4
ordpp_index 6
permutation_ordpp 7
simulate_ordpp 8
stability_ordpp 9

Index **11**

`as_ordpp_map` *Frozen Map from External Scores and Loadings*

Description

Wraps category scores and a direction obtained by another method (for example optimal-scaling or polychoric PCA) as an `ordpp` object, so that it can be evaluated with `validate_ordpp`, `heldout_test_ordpp` and `decompose_ordpp`.

Usage

```
as_ordpp_map(x, scores, direction, weights = NULL,
             frequencies = seq(0.2, 3, length.out = 15),
             index = c("full", "shape"), label = "external")
```

Arguments

`x`	Training data used to standardize the scores.
`scores`	List with one numeric vector per item, one value per observed category in increasing category order.
`direction`	Loading vector.
`weights`	Optional positive training weights.
`frequencies, index`	As in `ordpp`.
`label`	Name stored in the `scoring` field.

Value

An object of class `"ordpp"` that can be evaluated but not refitted.

Examples

```
s <- simulate_ordpp(150, 4, seed = 1)
m <- as_ordpp_map(s$x, lapply(s$x, function(v) sort(unique(v))),
                  direction = c(1, 1, 1, -1), label = "sum score")
validate_ordpp(m, s$x)
```

decompose_ordpp *Variance and Shape Components of the Projection Index*

Description

Reports the ratio of the projection variance to its independence variance together with the full and shape indices.

Usage

```
decompose_ordpp(object, newdata = NULL, weights = NULL)
```

Arguments

object	A fitted ordpp object or frozen map.
newdata	Optional held-out data; if omitted the training data are used.
weights	Optional positive weights for newdata.

Details

At low frequencies the full index is dominated by $(\mathrm{Var}(Z)/\mathrm{Var}(Z^0) - 1)^2$, which is driven by pairwise correlations between transformed items and rewards both inflated and deflated projection variance. The shape index removes this component. A variance ratio well below one indicates a contrast among positively correlated items.

Value

A one-row data frame with variance_ratio, full and shape.

Examples

```
s <- simulate_ordpp(150, 5, seed = 1)
f <- suppressWarnings(ordpp(s$x, scoring = "linear", nstart = 1,
                            maxit = 30, seed = 2))
decompose_ordpp(f)
```

heldout_test_ordpp *Frozen-Map Held-out Permutation Test*

Description

Tests mutual independence of the items in held-out data along a map (scores and loadings) estimated on separate observations.

Usage

```
heldout_test_ordpp(object, newdata, B = 999L, seed = NULL, type = NULL)
## S3 method for class 'ordpp_test'
print(x, ...)
```

Arguments

object	A fitted ordpp object or frozen map.
newdata	Held-out data not used to estimate object.
B	Number of permutations.
seed	Optional seed for reproducible permutations; the caller's random number state is restored on exit.
type	Index type; defaults to the fitted type.
x	A test result.
...	Unused.

Details

Each item column of the held-out data is permuted independently and the index is recomputed with the map held fixed. Under iid rows and mutual item independence the permuted samples are exchangeable with the observed one, so the Monte Carlo p-value $(1 + \sum_b 1\{T_b \geq T\})/(B + 1)$ is valid in finite samples. Validity requires that newdata played no part in fitting or tuning. No refitting is performed, so the test is fast. It is not valid for clustered or complex survey samples.

Value

An object of class "ordpp_test" with statistic, null, p.value, B, n, type, method and hypothesis.

Examples

```
s <- simulate_ordpp(300, 5, seed = 1)
f <- suppressWarnings(ordpp(s$x[1:150, ], scoring = "linear", nstart = 1,
                            maxit = 30, seed = 2))
heldout_test_ordpp(f, s$x[151:300, ], B = 99, seed = 3)
```

---

ordpp *Fit an Exploratory Ordinal Projection*

Description

Jointly estimates monotone category scores and a unit-length projection direction by maximizing a characteristic-function discrepancy between the projected joint distribution and its counterpart under item independence with the observed marginals.

Usage

```
ordpp(x, weights = NULL, scoring = c("item", "common", "linear"),
      lambda = 0, gamma = 0.01, nstart = 3L, maxit = 500L,
      frequencies = seq(0.2, 3, length.out = 15),
      min_gap = 0.03, variance_floor = 0.05,
      index = c("full", "shape"), seed = NULL)
## S3 method for class 'ordpp'
predict(object, newdata, ...)
## S3 method for class 'ordpp'
print(x, ...)
## S3 method for class 'ordpp'
plot(x, type = c("projection", "scores"), ...)
```

Arguments

x	For ordpp, a complete data frame of numeric category codes or ordered factors (numeric labels are used as ranks only). For methods, a fitted object.
weights	Optional positive observation weights; their scale is irrelevant.
scoring	"item": item-specific monotone curves; "common": one raw curve shared by items with identical response scales; "linear": fixed equally spaced scores.
lambda	Nonnegative L1 penalty on loadings. Encourages concentration; it does not perform exact variable selection.
gamma	Nonnegative penalty on unequal adjacent score gaps.
nstart	Number of optimizer starts (the first is the leading principal component of equally scored items).
maxit	Iteration budget per start for L-BFGS-B.
frequencies	Positive frequencies, on the scale of the independence standard deviation, at which characteristic functions are compared.
min_gap	Lower bound on each raw gap as a fraction of the mean gap.
variance_floor	Soft lower bound on the projection variance (the reference variance is one).
index	"full" compares the projected law with the independence reference; "shape" first rescales the projection to the reference variance, removing the pairwise-covariance component. See decompose_ordpp.
seed	Optional seed for reproducible starts; the caller's random number state is restored on exit.
object	A fitted ordpp object.
newdata	Complete data with the same named items and no unseen categories.
type	Kernel density of the training projection, or category-score curves.
...	Further graphical arguments, or unused.

Details

All declared categories must occur in the training data; missing values and unseen categories are rejected. Scores are strictly increasing, centred and of unit variance under the (weighted) training marginals. Weights change the target distribution but do not provide design-based standard errors. Inspect convergence, starts and at_bounds before interpretation.

Value

An object of class "ordpp": a list with direction, scores, levels, variable_names, projection, transformed, weights, index (unpenalized), objective (penalized), roughness, projection_variance, n, scoring, convergence, at_bounds, starts, settings and call. predict returns the projected scores of newdata using the frozen training dictionary.

See Also

validate_ordpp, heldout_test_ordpp, permutation_ordpp, stability_ordpp

Examples

```
s <- simulate_ordpp(150, 5, seed = 1)
f <- ordpp(s$x, scoring = "linear", nstart = 2, seed = 2)
f
predict(f, s$x[1:3, ])

# Item-specific monotone scores (slower)
fi <- ordpp(s$x, scoring = "item", nstart = 2, seed = 2)
fi$scores
```

---

ordpp_index *Characteristic-Function Discrepancy and Held-out Evaluation*

Description

Compares the characteristic function of a projection with the product of the characteristic functions of its item components, i.e. with the projection's law under item independence with the same marginals.

Usage

```
ordpp_index(y, direction, weights = NULL,
            frequencies = seq(0.2, 3, length.out = 15),
            type = c("full", "shape"))
validate_ordpp(object, newdata, weights = NULL, type = NULL)
```

Arguments

y	Finite numeric matrix of transformed items.
direction	Nonzero loading vector, one entry per column of y.
weights	Optional positive observation weights.
frequencies	Positive frequencies on the independence-SD scale.
type	"full" or "shape"; for validate_ordpp the default is the type used in fitting.
object	A fitted ordpp object.
newdata	Complete held-out data with known categories and named items.

Details

validate_ordpp freezes the fitted scores and loadings; the reference marginals and the independence variance are re-estimated from the held-out rows. The index measures departure from item independence along the projection; it is not a test of polarization or of a particular factor structure. A finite frequency grid need not detect every alternative. The empirical index is positive in finite samples even under independence; use heldout_test_ordpp for calibration.

Value

A nonnegative scalar.

Examples

```
y <- as.matrix(expand.grid(a = c(-1, 1), b = c(-1, 1)))
ordpp_index(y, c(1, 1))                         # exact independence: 0
ordpp_index(cbind(c(-1, 1), c(-1, 1)), c(1, 1))  # perfect dependence
```

---

permutation_ordpp *Refitted Permutation Test of Item Independence*

Description

Independently permutes each item and repeats the entire fixed-setting optimization, so that the null distribution includes the search over scores and directions.

Usage

```
permutation_ordpp(x, B = 99L, seed = NULL, fit = NULL, trace = FALSE, ...)
```

Arguments

x	Complete ordinal data frame.
B	Number of permutations.
seed	Optional seed for the permutations and optimizer starts; the caller's random number state is restored on exit.
fit	Optional ordpp object whose settings (scoring, penalties, starts, iterations, frequencies, safeguards, index type) are reused; this guarantees that observed and null fits use the same procedure.
trace	Print progress every 10 permutations.
...	Fitting arguments passed to ordpp when fit is not supplied; weights and seed are not allowed.

Details

Valid for iid unweighted rows under mutual item independence when the settings are fixed independently of x. If tuning parameters were chosen using x, the selection must be repeated inside each permutation. The test is computationally expensive; heldout_test_ordpp is a fast alternative when a held-out sample is available.

Value

An object of class "ordpp_test"; the observed statistic is the maximized penalized objective of a refit to x, returned as observed_fit. null_convergence records optimizer codes of the null fits.

Examples

```
s <- simulate_ordpp(120, 4, seed = 1)
f <- suppressWarnings(ordpp(s$x, scoring = "linear", nstart = 1,
                            maxit = 20, seed = 2))
permutation_ordpp(s$x, B = 9, seed = 3, fit = f)
```

---

simulate_ordpp *Synthetic Ordinal Data*

Description

Five-category items obtained by thresholding latent continuous responses, with informative and noise items.

Usage

```
simulate_ordpp(n = 500, p = 8, separation = 1.5, seed = NULL, strength = 1,
               scenario = c("mixture", "independent", "rare",
                            "two_factor", "response_style"))
```

Arguments

n	Number of observations (at least 10).
p	Number of items (at least 4).
separation	Half-distance between the means of a two-component normal mixture for the latent factor; 0 gives a single normal factor.
seed	Optional seed; the caller's random number state is restored on exit.
strength	Multiplier of all factor loadings; small values give weak dependence for power studies, and 0 gives independent items.
scenario	"mixture": items 1–3 load positively and item 4 negatively on the latent factor; remaining items are noise. "independent": all items independent with skewed marginals. "rare": as "mixture" with a rarely used top category on the informative items. "two_factor": adds a second normal factor on items 5–7. "response_style": 30 percent of respondents stretch all latent responses, inducing dependence even among noise items.

Details

Zero separation still yields dependent items through a shared factor and is not an independence null; use scenario = "independent" for that.

Value

A list with x, latent, group, loading, thresholds and scenario.

Examples

```
s <- simulate_ordpp(100, 6, seed = 1, scenario = "independent")
sapply(s$x, table)
```

---

stability_ordpp *Bootstrap Stability of Loadings and Category Scores*

Description

Refits the projection on bootstrap resamples of rows (or of clusters), aligns signs with the original direction and summarizes variability.

Usage

```
stability_ordpp(fit, x, B = 50L, seed = NULL, weights = NULL, cluster = NULL,
                trace = FALSE)
## S3 method for class 'ordpp_stability'
print(x, ...)
```

Arguments

fit	A fitted ordpp object; its settings are reused.
x	For stability_ordpp, the data used for fit; for the print method, a stability result.
B	Number of bootstrap refits.
seed	Optional seed for resampling and optimizer starts; the caller's random number state is restored on exit.
weights	Optional weights, resampled with their rows.
cluster	Optional cluster identifiers; whole clusters are resampled.
trace	Print progress every 5 refits.
...	Unused.

Details

The summaries describe algorithmic stability. Percentile intervals after data-driven selection need not have nominal coverage, and resampling rows ignores stratification and other design features.

Value

An object of class "ordpp_stability" with a loading summary, the matrix of aligned directions, per-item score matrices scores, index, convergence, absolute cosine congruence with the original direction and the number of failures (for example, resamples missing a rare category).

Examples

```
s <- simulate_ordpp(150, 4, seed = 1)
f <- suppressWarnings(ordpp(s$x, scoring = "linear", nstart = 1,
                            maxit = 20, seed = 2))
stability_ordpp(f, s$x, B = 5)
```

Index

as_ordpp_map, 2

decompose_ordpp, *2*, 3, *5*

heldout_test_ordpp, *2*, 3, *6*, *7*

ordpp, *2*, 4, *7*
ordpp_index, 6

permutation_ordpp, *6*, 7
plot.ordpp *(*ordpp*)*, 4
predict.ordpp *(*ordpp*)*, 4
print.ordpp *(*ordpp*)*, 4
print.ordpp_stability
*(*stability_ordpp*)*, 9
print.ordpp_test *(*heldout_test_ordpp*)*, 3

simulate_ordpp, 8
stability_ordpp, *6*, 9

validate_ordpp, *2*, *6*
validate_ordpp *(*ordpp_index*)*, 6