Package: HDElliptical (via r-universe)

October 14, 2026

Type Package

Title High-Dimensional Methods for Elliptically Symmetric Distributions

Version 0.1.3

Description Fast, documented implementations of robust estimation, testing, dimension reduction, classification, and clustering methods for high-dimensional elliptically symmetric data. Computational kernels use 'Rcpp' and 'RcppArmadillo'. The package follows methods reviewed in Feng (2026), ``High-Dimensional Data Analysis for Elliptically Symmetric Distributions'' <https://github.com/flnankai/HDElliptical/releases>.

License MIT + file LICENSE

URL https://github.com/flnankai/HDElliptical

BugReports https://github.com/flnankai/HDElliptical/issues

Depends R (>= 4.2.0)

Imports mvtnorm, Rcpp, stats

LinkingTo Rcpp, RcppArmadillo

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

NeedsCompilation yes

SystemRequirements C++17

RoxygenNote 7.3.3

Author Long Feng [aut, cre, cph] (Copyright holder for the HDElliptical implementation), Dan Zhuang [ctb] (SEMC methodology and the external GEMcluster reference implementation used for validation; no GEMcluster source code is included)

Maintainer Long Feng <flnankai@nankai.edu.cn>

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

Date/Publication 2026-09-14 14:15:07 UTC

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

RemoteRef HEAD

RemoteSha 7fa5c2878b58d9ed5a63fe371366756306d0ffa7

Contents

acg_loglik . . . 6
bai_saranadasa_two_sample_test . . . 6
basic_shape . . . 7
basics_shape . . . 9
bickel_levina_covariance_threshold . . . 10
book_gaussian_alpha_cauchy_test . . . 11
cai_liu_adaptive_covariance_threshold . . . 13
cai_liu_xia_two_sample_test . . . 14
cca_bartlett_test . . . 16
chen_qin_two_sample_test . . . 17
chen_song_feng_rank_white_noise_test . . . 18
chen_zhang_zhong_covariance_test . . . 19
cheng_sscm_equality_test . . . 20
chime_clustering . . . 21
classical_cca . . . 23
classical_cusum_test . . . 24
classical_lda_classifier . . . 26
classical_pca . . . 27
classical_qda_classifier . . . 28
classical_spatial_tests . . . 29
clime_precision . . . 31
composite_t2_two_sample_test . . . 32
conditional_alpha_sieve_design . . . 34
conditional_alpha_sieve_fit . . . 35
conditional_factor_wald_test . . . 36
dsda_classifier . . . 37
ec2_covariance . . . 38
elliptical_factor_number . . . 40
elliptical_factor_precision . . . 41
elliptical_oracle_classifier . . . 42
elliptical_regularized_hotelling_cauchy_test . . . 43
elliptical_regularized_hotelling_test . . . 45
erht_change_point_test . . . 46
erht_wbs . . . 48
fair_classifier . . . 50
fantope_pca . . . 52
feng_jiang_liu_xiong_panel_independence_test . . . 53

feng_lan_liu_ma_alpha_max_test . . . 54
feng_liu_ma_white_noise_test . . . 55
feng_liu_rank_sphericity_test . . . 56
feng_spatial_rank_proportionality_test . . . 57
feng_sun_one_sample_test . . . 59
feng_wang_pdq_two_sample_test . . . 60
feng_zhang_liu_spatial_rank_test . . . 63
feng_zou_wang_two_sample_sign_test . . . 65
feng_zou_wang_zhu_two_sample_test . . . 67
fisher_sun_gallagher_sphericity_test . . . 68
gaussian_alpha_combination_test . . . 69
gaussian_covariance_lrt . . . 70
gaussian_graphical_lasso . . . 71
gaussian_lda_oracle . . . 72
gaussian_mixture_em . . . 73
gaussian_qda_oracle . . . 75
gaussian_wilks_independence_test . . . 76
generalized_sign_pca . . . 77
generic_weighted_hr_location . . . 79
gqda_classifier . . . 81
grs_alpha_test . . . 82
hallin_paindaveine_shape_test . . . 83
hd_classifier_fit . . . 84
high_dimensional_hr . . . 85
hotelling_tests . . . 88
hr_estimator . . . 89
hr_gqda . . . 90
if_pca . . . 92
independence_classifier . . . 94
inst_one_sample_test . . . 95
jiang_da_qda . . . 97
john_sphericity_test . . . 98
k_spatial_median . . . 99
kendall_factor_number . . . 100
kendall_pca . . . 101
li_chen_covariance_test . . . 102
li_shao_sparse_qda . . . 103
li_wang_zou_two_sample_sign_test . . . 105
linear_pool_covariance . . . 107
liu_feng_ma_spatial_sign_alpha_test . . . 109
lloyd_kmeans . . . 111
lpd_classifier . . . 112
ma_feng_wang_bao_conditional_alpha_test . . . 114
ma_lan_su_tsai_conditional_alpha_sum_test . . . 115
mauchly_sphericity_test . . . 116
nagao_identity_test . . . 117
normalize_shape . . . 118
ollila_raninen_shrinkage_covariance . . . 118

oracle_weighted_sign_sum_test . . . 120
park_ayyala_one_sample_test . . . 121
pesaran_cd_test . . . 123
pesaran_yamagata_alpha_test . . . 124
pmd_sparse_cca . . . 125
pmd_sparse_pca . . . 126
poet_covariance . . . 128
poet_tme . . . 129
predict.hd_classifier_fit . . . 130
predict.semc_fit . . . 131
regularized_spatial_sign_covariance . . . 132
relliptical . . . 133
robust_factor_subspace . . . 134
robust_gqda . . . 135
rothman_levina_zhu_covariance_threshold . . . 136
rspherical . . . 138
rts_factor . . . 138
scaled_spatial_median . . . 140
sdar_qda . . . 141
semc_fit . . . 142
semc_select_k_gap . . . 147
shao_threshold_lda . . . 149
sign_whitened_sparse_cca . . . 150
sm_sscm . . . 152
sparse_k_spatial_median . . . 153
sparse_kmeans . . . 155
sparse_kmeans_select_s . . . 156
sparse_kmedian . . . 158
sparse_plugin_lda . . . 160
sparse_plugin_qda . . . 161
sparse_sm_select_k . . . 162
sparse_sm_select_tau . . . 164
sparse_spatial_sign_pca . . . 166
spatial_kendall . . . 168
spatial_median . . . 169
spatial_rank . . . 170
spatial_rank_covariance . . . 170
spatial_sign . . . 171
spatial_sign_change_point_test . . . 172
spatial_sign_max_test . . . 174
spatial_sign_maxsum_test . . . 175
spatial_sign_pca . . . 177
spatial_sign_poet . . . 178
spatial_sign_precision . . . 179
spatial_sign_precision_lda . . . 181
spatial_sign_sphericity_test . . . 183
srivastava_du_one_sample_test . . . 184
srivastava_katayama_kano_two_sample_test . . . 185

sscca . . . 186
sscm . . . 188
sslda . . . 189
ssqda . . . 190
tensor_spatial_sign_precision . . . 192
threshold_spatial_sign_precision . . . 194
threshold_tensor_spatial_sign_precision . . . 195
tinst_two_sample_test . . . 196
truncated_power_pca . . . 197
tyler_shape . . . 199
wang_feng_dms_test . . . 200
wang_liu_feng_ma_serial_panel_test . . . 201
wang_liu_feng_vector_independence_test . . . 202
wang_liu_feng_vector_u_independence_test . . . 204
wang_peng_li_one_sample_test . . . 205
wang_xu_approx_randomization_test . . . 207
wang_yao_corrected_john_test . . . 209
wang_yao_corrected_lrt . . . 210
wang_zhao_feng_wang_mutual_fund_fdr . . . 211
weighted_scaled_spatial_median . . . 213
weighted_spatial_sign_alpha_oracle_test . . . 214
white_noise_portmanteau_test . . . 215
xu_lin_wei_pan_aspu_test . . . 216
yan_zhao_feng_weighted_max_test . . . 219
yan_zhao_feng_weighted_maxsum_test . . . 220
zhang_feng_radial_directional_test . . . 222
zhang_feng_rank_tests . . . 224
zhang_zhou_guo_tests . . . 227
zhang_zhu_zhang_two_sample_test . . . 229
zhao_chen_wang_spatial_sign_white_noise_test . . . 231
zhao_chen_zi_inst_alpha_test . . . 232
zhao_conditional_spatial_sign_sum_test . . . 233
zhao_feng_strongcorr_sign_test . . . 234
zhao_feng_wang_wang_robust_alpha_test . . . 236
zhao_wang_conditional_spatial_sign_test . . . 238
zhao_yang_zhang_feng_wang_adaptive_sphericity_test . . . 239
zhao_yang_zhang_feng_wang_sign_max_test . . . 241
zou_peng_feng_wang_sphericity_test . . . 242

Index **244**

acg_loglik *Angular central Gaussian log-likelihood*

Description

Evaluates the angular central Gaussian log-likelihood for nonzero row vectors. Row lengths need not equal one because they cancel from the quadratic contribution.

Usage

```
acg_loglik(x, shape, include_constant = FALSE)
```

Arguments

x	Nonzero directions in rows.
shape	A positive definite shape matrix.
include_constant	Include the density normalizing constant.

Value

A scalar log-likelihood.

References

Tyler, D. E. (1987). A distribution-free M-estimator of multivariate scatter. *Annals of Statistics*, 15, 234–251. doi:10.1214/aos/1176350263.

Examples

```
directions <- rbind(c(1, 0), c(0, 1), c(1, 1))
acg_loglik(directions, diag(2))
```

bai_saranadasa_two_sample_test *Original Bai–Saranadasa two-sample mean test*

Description

Tests equality of two multivariate means under a common covariance matrix using the original Bai–Saranadasa statistic. This is deliberately a separate API from covariance-unrestricted Chen–Qin tests: its numerator is

$$\xi = \|\bar{X}_1 - \bar{X}_2\|^2 - \frac{N}{n_1 n_2}\,\mathrm{tr}(S_p),$$

where S_p is the unbiased pooled covariance matrix and $N = n_1 + n_2$.

Usage

```
bai_saranadasa_two_sample_test(x, y)
```

Arguments

x, y	Numeric matrices or data frames, with observations in rows and the same variables in columns. Each group must have at least two observations.

Details

The traces of S_p and S_p^2 are computed using the smaller of a primal $p \times p$ matrix and a dual $N \times N$ Gram matrix. Both groups are divided by the same global numerical scale before any moments are formed. The standardised statistic is computed on that safe scale, while reported means, traces, numerator, and variance are converted back according to their physical dimensions. Consequently a reported fourth-order component can be infinite when its value exceeds double range, or zero when it is below the representable range, without making the standardised statistic or p-value non-finite. The finite internal normalising variance is retained in `diagnostics`. The estimated null variance must be strictly positive; no ridge, absolute-value repair, or variance flooring is applied.

Value

An object of class `c("hd_location_test", "htest")`. Its `components` include the original numerator `xi`, pooled covariance traces, the bias-corrected estimate of $\mathrm{tr}(\Sigma^2)$, and the estimated null variance.

References

Bai, Z. and Saranadasa, H. (1996). Effect of high dimension: by an example of a two sample problem. *Statistica Sinica*, 6, 311–329.

Examples

```
set.seed(22)
x <- matrix(rnorm(24), nrow = 8, ncol = 3)
y <- matrix(rnorm(27, mean = 0.2), nrow = 9, ncol = 3)
bai_saranadasa_two_sample_test(x, y)
```

---

basic_shape *BASIC bias-adjusted spatial-sign shape estimator*

Description

Applies the real-valued BASIC approximate inverse map independently to the eigenvalues of the fitted SSCM. For a shape eigenvalue λ, the map used by Raninen and Ollila is

$$\widetilde{\delta}(\lambda) = \frac{1}{2}\int_0^1 \frac{\lambda}{1-t+t\lambda}(1-t)^{p/2-1}\,dt.$$

The implementation evaluates the equivalent smooth integral after $1 - t = v^2$ with paired Gauss–Legendre rules, brackets each inverse root, and uses bisection. It is independent of the official lookup table: no interpolation, spline extrapolation, eigenvalue floor, or out-of-range repair is used. Raw inverted eigenvalues are finally normalized to sum p.

Usage

```
basic_shape(
  x,
  center = c("spatial", "mean", "none"),
  tol = 1e-08,
  max_iter = 1000L,
  strict = TRUE,
  keep_signs = FALSE,
  quadrature_order = 64L,
  inversion_tol = 1e-10,
  integration_tol = 1e-10,
  inversion_max_iter = 100L,
  max_bracket_iter = 100L
)
```

Arguments

x	A finite numeric matrix or data frame, observations in rows.
center	One of "spatial", "mean", "none", or a supplied finite numeric center. "none" treats the data as already centered.
tol, max_iter, strict	Spatial-median convergence controls.
keep_signs	Whether to retain the fitted sign matrix.
quadrature_order	Order of the coarse Gauss–Legendre rule; a rule of twice this order supplies the returned value and error estimate.
inversion_tol	Positive absolute/relative bisection tolerance.
integration_tol	Positive maximum coarse/fine quadrature discrepancy.
inversion_max_iter	Maximum bisection iterations per eigenvalue.
max_bracket_iter	Maximum upper-bracket doublings per eigenvalue.

Details

Boundary SSCM eigenvalues equal to zero map exactly to zero and are reported. For p = 1, shape is identically one and no numerical inversion is needed.

Value

An hd_shape_estimator with the BASIC shape, SSCM eigensystem, raw and normalized inverse eigenvalues, mapped values, integration and inversion errors, brackets, iterations, and boundary diagnostics.

References

Raninen, E. and Ollila, E. (2022). Bias adjusted sign covariance matrix. *IEEE Signal Processing Letters*, 29, 339–343. doi:10.1109/LSP.2021.3134940.

Examples

```
x <- matrix(c(-2, 0, -1, 2, 0, -2, 1, 1, 2, -1, 3, 2),
            ncol = 2, byrow = TRUE)
basic_shape(x, quadrature_order = 32)
```

---

basics_shape *BASICS shrinkage plus bias-adjusted shape estimator*

Description

Computes the RSSCM data weight, forms $\widehat{\Lambda}_{RSSCM} = \widehat{\alpha} p S_{sgn} + (1 - \widehat{\alpha}) I_p$, and applies the same self-contained BASIC inverse map as basic_shape() to the eigenvalues of $\widehat{\Lambda}_{RSSCM}/p$. This follows the BASICS construction while avoiding the official code's spline extrapolation. The RSSCM and BASIC numerical diagnostics are both retained.

Usage

```
basics_shape(
  x,
  center = c("spatial", "mean", "none"),
  tol = 1e-08,
  max_iter = 1000L,
  strict = TRUE,
  keep_signs = FALSE,
  quadrature_order = 64L,
  inversion_tol = 1e-10,
  integration_tol = 1e-10,
  inversion_max_iter = 100L,
  max_bracket_iter = 100L
)
```

Arguments

x	A finite numeric matrix or data frame, observations in rows.
center	One of "spatial", "mean", "none", or a supplied finite numeric center. "none" treats the data as already centered.

`tol`, `max_iter`, `strict`
: Spatial-median convergence controls.

`keep_signs`
: Whether to retain the fitted sign matrix.

`quadrature_order`
: Order of the coarse Gauss–Legendre rule; a rule of twice this order supplies the returned value and error estimate.

`inversion_tol`
: Positive absolute/relative bisection tolerance.

`integration_tol`
: Positive maximum coarse/fine quadrature discrepancy.

`inversion_max_iter`
: Maximum bisection iterations per eigenvalue.

`max_bracket_iter`
: Maximum upper-bracket doublings per eigenvalue.

Value

An `hd_shape_estimator` containing the BASICS shape, RSSCM, its raw/projected shrinkage weight, inverse-map details, and full diagnostics.

References

Raninen, E. and Ollila, E. (2022). Bias adjusted sign covariance matrix. *IEEE Signal Processing Letters*, 29, 339–343. doi:10.1109/LSP.2021.3134940.

Examples

```
x <- matrix(c(-2, 0, -1, 2, 0, -2, 1, 1, 2, -1, 3, 2),
            ncol = 2, byrow = TRUE)
basics_shape(x, quadrature_order = 32)
```

---

bickel_levina_covariance_threshold

Bickel–Levina hard-thresholded covariance estimator

Description

Forms the centered sample covariance with divisor n, as in Bickel and Levina (2008), and applies

$$s_{ij}\, 1\{|s_{ij}| > \lambda\}.$$

When `threshold = NULL`, $\lambda = C\sqrt{\log(p)/n}$. The book draft first defines a divisor-$n-1$ covariance and then cites the primary divisor-n formula. `divisor` makes this finite-sample distinction explicit; the default is the primary-paper convention. Thresholding need not preserve positive definiteness and no eigenvalue repair is performed.

Usage

```
bickel_levina_covariance_threshold(
  x,
  threshold = NULL,
  constant = 1,
  center = TRUE,
  divisor = c("n", "n-1"),
  threshold_diagonal = TRUE
)
```

Arguments

x	Numeric matrix with observations in rows.
threshold	A finite non-negative threshold in covariance units. NULL uses the rate threshold controlled by constant.
constant	A finite non-negative multiplier for the rate threshold.
center	Whether to remove column means.
divisor	Either "n" (primary-paper default) or "n-1".
threshold_diagonal	Whether to apply the rule to diagonal entries.

Value

An hd_covariance_estimator list whose estimate field is the thresholded covariance matrix.

References

Bickel, P. J. and Levina, E. (2008). *Annals of Statistics*, 36, 2577–2604.

Examples

```
set.seed(31)
x <- matrix(rnorm(40), nrow = 10, ncol = 4)
bickel_levina_covariance_threshold(x, threshold = 0.2)
```

---

book_gaussian_alpha_cauchy_test

Book Gaussian alpha Cauchy benchmark

Description

Implements the formula-complete Gaussian benchmark printed in Chapter 4, kept distinct from gaussian_alpha_combination_test(), which implements the primary Feng–Lan–Liu–Ma Bonferroni rule. With unrestricted OLS residual variances $\hat{\sigma}_i^2$, residual correlation $\hat{R}$, and intercept estimates $\hat{\alpha}$, this function uses

$$T_{sum} = \frac{T \sum_i \hat{\alpha}_i^2 / \hat{\sigma}_i^2 - N}{\{2\mathrm{tr}(\hat{R}^2)\}^{1/2}}$$

and the maximum squared OLS intercept t statistic. Their upper-tail normal and extreme-value p-values are combined by the equal-weight Cauchy rule.

Usage

```
book_gaussian_alpha_cauchy_test(returns, trace_R2, factors = NULL)
```

Arguments

returns	Finite observation-by-asset numeric matrix or data frame.
trace_R2	Strictly positive supplied value of $\widehat{\mathrm{tr}(R^2)}$.
factors	NULL, a finite length-T numeric vector, or a finite observation-by-factor numeric matrix or data frame.

Details

The manuscript denotes the trace term abstractly and does not prescribe a unique finite-sample estimator for this Gaussian benchmark. Consequently, trace_R2 is required rather than silently replacing it by a residual plug-in or a bias-corrected alternative.

Value

An upper-tail alpha-test object explicitly labelled as a book benchmark rather than a primary named procedure.

References

High-Dimensional Data Analysis for Elliptically Symmetric Distributions, Chapter 4, equations for the Gaussian alpha benchmark.

Examples

```
f <- cbind(seq(-1, 1, length.out = 12))
y <- outer(seq_len(12), 1:3, function(i, j) cos(i / 2 + j))
book_gaussian_alpha_cauchy_test(y, trace_R2 = 3, factors = f)
```

`cai_liu_adaptive_covariance_threshold` *Cai–Liu adaptive covariance thresholding*

Description

The primary-paper pilot covariance and variability estimate are

$$\hat{\sigma}_{ij} = n^{-1} \sum_k Z_{ki} Z_{kj}, \qquad \hat{\theta}_{ij} = n^{-1} \sum_k (Z_{ki} Z_{kj} - \hat{\sigma}_{ij})^2,$$

with entrywise threshold

$$\lambda_{ij} = \delta \sqrt{\hat{\theta}_{ij} \log(p)/n}.$$

Both quantities therefore use the same divisor-n covariance by default. Selecting `divisor = "n-1"` is explicit and is reported in the diagnostics; it is not the primary finite-sample formula.

Usage

```
cai_liu_adaptive_covariance_threshold(
  x,
  delta = 2,
  rule = c("hard", "soft", "scad", "adaptive_lasso"),
  scad_a = 3.7,
  adaptive_eta = 1,
  center = TRUE,
  divisor = c("n", "n-1"),
  threshold_diagonal = TRUE
)
```

Arguments

`x`	Numeric matrix with observations in rows.
`delta`	Finite non-negative adaptive-threshold multiplier.
`rule`	One of `"hard"`, `"soft"`, `"scad"`, or `"adaptive_lasso"`.
`scad_a`	SCAD shape parameter, greater than two.
`adaptive_eta`	Non-negative adaptive-lasso exponent.
`center`	Whether to remove column means.
`divisor`	Either `"n"` (primary-paper default) or `"n-1"`.
`threshold_diagonal`	Whether to apply the rule to diagonal entries.

Value

An `hd_covariance_estimator` object retaining $\hat{\theta}$ and every entrywise threshold.

References

Cai, T. and Liu, W. (2011). *Journal of the American Statistical Association*, 106, 672–684. arXiv:1102.2237.

Examples

```
set.seed(33)
x <- matrix(rnorm(40), nrow = 10, ncol = 4)
cai_liu_adaptive_covariance_threshold(x, delta = 2)
```

`cai_liu_xia_two_sample_test`

Cai–Liu–Xia precision-adjusted two-sample maximum test

Description

Tests equality of two high-dimensional mean vectors using the precision-adjusted maximum statistic of Cai, Liu, and Xia (2014). The calibration is the type-I extreme-value limit

$$F(g) = \exp\{-\pi^{-1/2}\exp(-g/2)\},$$

for

$$G = M - 2\log(p) + \log\{\log(p)\}.$$

Usage

```
cai_liu_xia_two_sample_test(
  x,
  y,
  precision = NULL,
  precision_source = c("adaptive", "oracle", "feasible"),
  threshold_delta = 2,
  eigen_floor = sqrt(.Machine$double.eps)
)
```

Arguments

`x`, `y`	Numeric matrices or data frames with observations in rows and the same variables in columns. Each group needs at least two rows and the dimension must satisfy $p \geq 2$.
`precision`	A finite symmetric $p \times p$ matrix. It is required for `precision_source = "oracle"` or `"feasible"` and must be `NULL` for the adaptive path. An oracle matrix must be positive definite. A feasible estimate may be indefinite, as allowed by the original paper, provided all transformed empirical variances are strictly positive.
`precision_source`	One of `"adaptive"`, `"oracle"`, or `"feasible"`. This argument must identify whether a supplied matrix is known or estimated because the two cases have different denominators.

`threshold_delta`
: Positive multiplier δ for adaptive thresholding. The paper recommends 2 as a fixed choice.

`eigen_floor`
: A non-negative relative eigenvalue floor below 1 for the adaptive thresholded covariance. Set it to zero to prohibit adjustment; a non-positive thresholded eigenvalue then produces an error.

Details

`precision_source` deliberately separates three statistically different paths. With `"oracle"`, `precision` is a known population precision matrix and the denominator is its diagonal. With `"feasible"`, `precision` is a user-supplied estimate and the denominator is formed from empirical within-group variances of the transformed observations, each with divisor n_k, as in equations (6)–(7) of the paper. With `"adaptive"`, the function estimates a precision matrix by hard-thresholding the pooled covariance and then uses that same feasible denominator. It never uses `diag(precision)` to standardise an estimated-precision statistic.

For the adaptive backend, the pooled covariance and $\widehat{\theta}_{ij}$ both use divisor $N = n_1 + n_2$, and

$$\lambda_{ij} = \delta\sqrt{\widehat{\theta}_{ij}\log(p)/N}.$$

If the thresholded covariance is not numerically positive definite, its eigenvalues are raised to `eigen_floor` times a data-scale reference before inversion. The floor, the unadjusted and adjusted extreme eigenvalues, and the number of modified eigenvalues are all returned in `diagnostics`. Both samples are centered relative to observed anchors before precision transformation, avoiding cancellation under a large common translation. The adaptive covariance and fourth-order threshold moments are also formed on an internal common scale and then mapped back to the original units.

Value

An object of class `c("hd_location_test", "htest")`. The reported statistic `G` has the extreme-value calibration, while `raw.statistic` contains `M`. Coordinatewise transformed scores, their denominators, the maximizing coordinate index and name, and the precision matrix used by the test are retained in `components`. `diagnostics` identifies the precision path, denominator construction, and every adaptive-threshold eigenvalue adjustment.

References

Cai, T. T., Liu, W., and Xia, Y. (2014). Two-sample test of high dimensional means under dependence. *Journal of the Royal Statistical Society: Series B*, **76**, 349–372.

Cai, T. T. and Liu, W. (2011). Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association*, **106**, 672–684.

Examples

```
set.seed(29)
x <- matrix(rnorm(80), 20, 4)
y <- matrix(rnorm(96, 0.2), 24, 4)
cai_liu_xia_two_sample_test(x, y)
```

```
omega <- diag(4)
cai_liu_xia_two_sample_test(
  x, y, precision = omega, precision_source = "oracle"
)
```

`cca_bartlett_test` *Bartlett likelihood-ratio approximation for canonical correlations*

Description

Tests the tail null that canonical correlations after `null_rank` are zero. For `null_rank = k`, the statistic uses

$$\Lambda_k = \prod_{j=k+1}^{m} (1 - \widehat{\rho}_j^2), \qquad -\{n - 1 - (p_x + p_y + 1)/2\} \log \Lambda_k,$$

with chi-squared degrees of freedom `(p_x-k)(p_y-k)`. The special case `k = 0` is the full independence test printed in the book.

Usage

```
cca_bartlett_test(object, null_rank = 0L)
```

Arguments

`object`	A valid `classical_cca_fit` produced with mean centering.
`null_rank`	Non-negative integer rank under the tail null.

Value

An object of class `htest` with Wilks' Lambda and the Bartlett approximation diagnostics.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Wiley.

Examples

```
basis <- stats::poly(seq_len(12), degree = 4)
x <- basis[, 1:2, drop = FALSE]
y <- cbind(0.8 * basis[, 1] + 0.6 * basis[, 3],
           0.3 * basis[, 2] + sqrt(0.91) * basis[, 4])
colnames(x) <- c("x1", "x2")
colnames(y) <- c("y1", "y2")
fit <- classical_cca(x, y)
cca_bartlett_test(fit, null_rank = 0)
```

`chen_qin_two_sample_test` *Chen–Qin two-sample high-dimensional mean test*

Description

Tests equality of two mean vectors with the covariance-unrestricted U-statistic of Chen and Qin (2010). Its numerator is

$$T_{CQ} = \|\bar{X} - \bar{Y}\|^2 - \operatorname{tr}(S_X)/n_1 - \operatorname{tr}(S_Y)/n_2.$$

The variance estimator is the original paper's leave-out estimator of the two within-sample covariance traces and their cross trace. This is distinct from the common-covariance Bai–Saranadasa test.

Usage

```
chen_qin_two_sample_test(x, y)
```

Arguments

`x`, `y`	Numeric matrices or data frames with observations in rows and the same variables in columns. Each group must contain at least three observations.

Details

The numerator is exactly invariant to a common translation. The original finite-sample leave-out variance estimator, however, is not translation invariant; this known property is retained rather than silently replacing the published estimator. Both samples are internally divided by one common global scale. The returned `components` include both the original-unit quantities and their finite internally scaled counterparts. An original-unit fourth-order quantity can legitimately overflow to `Inf` or underflow to zero when the input itself spans the limits of double precision; the reported statistic is reconstructed from the scaled quantities and remains well defined. No ridge, absolute-value repair, or variance flooring is applied.

Value

An object of class `c("hd_location_test", "htest")`. The `components` field includes the U-statistic numerator, leave-out trace estimates `A1`, `A2`, and `A12`, the estimated null variance, and their `.scaled` counterparts used for stable calibration.

References

Chen, S. X. and Qin, Y.-L. (2010). A two-sample test for high-dimensional data with applications to gene-set testing. *Annals of Statistics*, 38, 808–835.

Examples

```
set.seed(24)
x <- matrix(rnorm(21), nrow = 7, ncol = 3)
y <- matrix(rnorm(24, mean = 0.2), nrow = 8, ncol = 3)
chen_qin_two_sample_test(x, y)
```

`chen_song_feng_rank_white_noise_test` *Chen–Song–Feng rank-based max white-noise test*

Description

Implements the two rank procedures for which the primary paper supplies direct, scalable statistics and complete extreme-value calibrations: Spearman's rho and Kendall's tau. The paper does not establish rank-based sum or adaptive Fisher tests; those constructions in the book draft are not used here.

Usage

```
chen_song_feng_rank_white_noise_test(
  x,
  lag = 1L,
  measure = c("spearman", "kendall"),
  keep_lag = FALSE
)
```

Arguments

`x`	Numeric matrix with time points in rows.
`lag`	Positive lag truncation level, no larger than `n - 2`.
`measure`	Either `"spearman"` or `"kendall"`.
`keep_lag`	Whether to retain lag-specific maxima and their locations.

Value

An `htest` object. Exact ties are rejected because the published null variance and distribution-free calibration assume continuous margins.

References

Chen, D., Song, F. and Feng, L. Rank Based Tests for High Dimensional White Noise. Statistica Sinica 35, 1323–1347. doi:10.5705/ss.202022.0382.

Examples

```
t <- seq_len(18)
x <- cbind(sin(t), cos(t / 2), sin(t / 3 + 0.2))
chen_song_feng_rank_white_noise_test(x, lag = 2, measure = "spearman")
```

---

chen_zhang_zhong_covariance_test

Chen–Zhang–Zhong high-dimensional covariance test

Description

Constructs the location-invariant unbiased estimators

$$T_{1n} = Y_{1n} - Y_{3n}, \qquad T_{2n} = Y_{2n} - 2Y_{4n} + Y_{5n}$$

of $\mathrm{tr}(\Sigma)$ and $\mathrm{tr}(\Sigma^2)$. For sphericity it uses $U_n = pT_{2n}/T_{1n}^2 - 1$; for identity it uses $V_n = T_{2n}/p - 2T_{1n}/p + 1$. In either case the primary null calibration is $nU_n/2$ or $nV_n/2$, respectively, against an upper standard-normal tail.

Usage

```
chen_zhang_zhong_covariance_test(x, null = c("sphericity", "identity"))
```

Arguments

x	Numeric matrix with observations in rows; at least four rows.
null	Either "sphericity" or "identity".

Value

An hd_covariance_test object retaining all five U-statistic building blocks.

References

Chen, S. X., Zhang, L.-X. and Zhong, P.-S. (2010). *Journal of the American Statistical Association*, 105, 810–819. doi:10.1198/jasa.2010.tm09560.

Examples

```
set.seed(42)
x <- matrix(rnorm(48), nrow = 12, ncol = 4)
chen_zhang_zhong_covariance_test(x, null = "sphericity")
```

`cheng_sscm_equality_test` *Cheng–Liu–Peng–Zhang–Zheng two-sample SSCM test*

Description

Tests equality of two population spatial-sign covariance matrices (SSCMs), equivalently proportionality of the two scatter matrices under elliptical symmetry. With separately estimated spatial medians and centered signs, the feasible statistic is

$$T = p(A + B - 2C),$$

where A and B average squared inner products over ordered within-group pairs and C averages them over all cross pairs.

Usage

```
cheng_sscm_equality_test(
  x,
  y,
  alpha = 0.05,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

`x`, `y`	Numeric matrices or data frames with observations in rows and the same variables in columns. At least two rows are needed per group.
`alpha`	Nominal level for the upper-tail calibration.
`tol`, `max_iter`	Spatial-median convergence controls.
`zero_tol`	Non-negative tolerance for a zero centered radius. The default detects exact zeros only.
`strict`	If `TRUE`, fail when the feasible calibration is undefined; otherwise return explicitly uncalibrated raw diagnostics.

Details

The implementation uses the complete finite-sample bias expression in Cheng et al. (2019), Equation (3.3). For each group it estimates the inverse radial moments through order three, evaluates the n^{-2} and n^{-3} terms separately, and subtracts $p\widehat{\delta}$. Under the null, Remark 3.1 estimates

$$q = p^{-1}\mathrm{tr}(\Lambda^2) = \frac{p}{n_1 + n_2}\{n_1(A - \delta_1) + n_2(B - \delta_2)\}.$$

Consequently the *standard-error denominator* and its square are

$$\widehat{v}_0 = \frac{2(n_1^{-1} + n_2^{-1})}{p+2}(p\widehat{q}), \qquad \widehat{v}_0^2 = \{\widehat{v}_0\}^2.$$

The reported statistic is $Z = (T - p\widehat{\delta})/\widehat{v}_0$ and the paper's rejection rule is the upper standard-normal tail.

A non-positive trace estimate, a zero fitted radius, or an unconverged spatial median invalidates calibration. No absolute value, variance floor, radius perturbation, or replacement center is used. With `strict = FALSE` the raw components are returned, but `statistic`, `p.value`, and the rejection decision are `NA` and `diagnostics$calibrated` is `FALSE`. The plug-in bias calibration has the additional regime stated in Remark 3.2 of the primary paper: under bounded eigenvalues it requires $p = o(n_l^{3/2})$ for both groups. This is stronger than merely using the theorem's $p = O(n_l^2)$ expansion condition.

Value

An object of class `c("hd_proportionality_test", "htest")`. `components` contains all ordered-pair components, bias terms, scaled inverse radial moments, the trace estimate, the standard-error denominator and its squared variance, fitted centers, signs, and SSCMs.

References

Cheng, G., Liu, B., Peng, L., Zhang, B., and Zheng, S. (2019). Testing the equality of two high-dimensional spatial sign covariance matrices. *Scandinavian Journal of Statistics*, **46**, 257–271. doi:10.1111/sjos.12350.

Examples

```
x <- matrix(c(-2, 0, 1, 2, 1, -1, 0, 2, -1, -2, 2, 0), 6, 2)
y <- matrix(c(-1, 1, 2, -2, 0, 3, 1, -1, 2, 0, -2, 1), 6, 2)
cheng_sscm_equality_test(x, y)
```

---

chime_clustering *CHIME clustering for a two-component Gaussian mixture*

Description

Fits the CHIME iteration of Cai, Ma, and Zhang for a two-component Gaussian mixture with a common covariance. At stage t, the sparse discriminant vector is the certified solution of

$$\min_{\beta} \ \tfrac{1}{2}\beta^T\widehat{\Sigma}_t\beta - \beta^T(\widehat{\mu}_{1t} - \widehat{\mu}_{2t}) + \lambda_t\|\beta\|_1.$$

The covariance is never inverted. Coordinate descent uses the actual diagonal of the quadratic metric and must satisfy the exact lasso KKT equations before an iterate is accepted.

Usage

```
chime_clustering(
  x,
  lambda,
  initial,
  K = 2L,
  max_iter = length(lambda) - 1L,
  lasso_tol = 1e-08,
  lasso_max_iter = 10000L,
  support_tol = 0,
  psd_tol = sqrt(.Machine$double.eps)
)
```

Arguments

x	Finite numeric matrix with observations in rows.
lambda	Explicit non-negative vector of length max_iter + 1. Its first entry fits the initial discriminant vector and each remaining entry is used by one EM update. No path is generated or reordered.
initial	Explicit list with omega, mu1, mu2, and covariance. Whenever omega exceeds one half, component labels are exchanged so the identifiable representation has omega at most one half.
K	Number of components. Only 2 is implemented.
max_iter	Number of EM updates. All are run; there is no unreported early stopping or reuse of a shorter lambda path.
lasso_tol	Positive relative-update and scaled-KKT tolerance.
lasso_max_iter	Maximum coordinate-descent sweeps at every stage.
support_tol	Non-negative reporting threshold for coefficient support. It does not alter coefficients or KKT equations.
psd_tol	Non-negative relative tolerance used only to diagnose the initial covariance as positive semidefinite. Eigenvalues are not changed.

Details

This function deliberately differs from the authors' numerical script in three ways required for a reusable unsupervised method: it applies no hidden covariance ridge, it never uses true labels to choose a penalty, and it requires the complete penalty path (including the initial beta fit) from the caller. The method is only defined here for K = 2, matching the stated model and the available primary implementation.

Value

A chime_fit object containing labels, responsibilities, all parameter stages, the explicit lambda path, covariance matrices, and per-stage objective, convergence, and KKT certificates. Score ties use the printed weak inequality and are assigned to component 1.

References

Cai, T. T., Ma, J., and Zhang, L. (2019). CHIME: Clustering of high-dimensional Gaussian mixtures with EM algorithm and its optimality. *Annals of Statistics*, **47**, 1234–1267. doi:10.1214/18AOS1711.

Examples

```
x <- rbind(
  c(-2.2, -1.0), c(-1.8, -1.2), c(-2.0, -0.8), c(-1.7, -1.1),
  c( 1.8,  1.0), c( 2.2,  1.1), c( 2.0,  0.9), c( 1.7,  1.2)
)
initial <- list(
  omega = 0.5, mu1 = c(-1.5, -0.8), mu2 = c(1.5, 0.8),
  covariance = diag(2)
)
chime_clustering(x, lambda = c(0.2, 0.15, 0.1), initial = initial,
                 max_iter = 2)
```

---

classical_cca *Classical canonical correlation analysis*

Description

Forms all three covariance blocks with one explicit divisor, constructs the symmetric inverse square roots of the two within-block covariance matrices, and applies an SVD to the whitened cross-covariance matrix. Both within-block matrices must be strictly positive definite and full rank at `rank_tol`. No ridge, pseudoinverse, or eigenvalue floor is used.

Usage

```
classical_cca(
  x,
  y,
  components = NULL,
  center = c("mean", "none"),
  covariance_divisor = c("n-1", "n"),
  rank_tol = sqrt(.Machine$double.eps)
)
```

Arguments

`x, y`	Paired numeric data matrices with equal row counts.
`components`	Number of leading canonical pairs. `NULL` returns all `min(ncol(x), ncol(y))` pairs after the within-block rank certificates.
`center`	Either `"mean"` or `"none"`, applied to both blocks.

`covariance_divisor`
: Either `"n-1"` or `"n"` for every covariance block; canonical correlations are invariant to this common choice.

`rank_tol`
: Positive relative tolerance for strict within-block rank and repeated-canonical-correlation diagnostics.

Value

A `classical_cca_fit` object with canonical coefficients, scores, structure loadings, correlations, covariance blocks, and whitened singular-vector projectors.

References

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28, 321–377.

Examples

```
basis <- stats::poly(seq_len(8), degree = 4)
x <- basis[, 1:2, drop = FALSE]
y <- cbind(0.8 * basis[, 1] + 0.6 * basis[, 3],
           0.3 * basis[, 2] + sqrt(0.91) * basis[, 4])
colnames(x) <- c("x1", "x2")
colnames(y) <- c("y1", "y2")
fit <- classical_cca(x, y, components = 1)
fit$canonical.correlations
```

---

classical_cusum_test *Classical CUSUM change-point test*

Description

Computes the univariate CUSUM

$$C_n(k) = n^{-1/2}\{S_k - kS_n/n\}$$

and rejects for a large trimmed maximum of $\{(k/n)(1-k/n)\}^{-\gamma}|C_n(k)|/\widehat{\sigma}$. For `gamma = 0`, `calibration = "asymptotic"` uses the classical two-sided Brownian-bridge (Kolmogorov) limit. `calibration = "grid-gaussian"` simulates the method's finite scan-grid Gaussian bridge; this is null calibration, not replication of a paper simulation study.

Usage

```
classical_cusum_test(
  x,
  gamma = 0,
  trim = 1L,
  sigma = NULL,
  variance = c("difference", "sample"),
  calibration = c("asymptotic", "grid-gaussian"),
```

```
  calibration_draws = 4999L,
  alpha = 0.05,
  seed = NULL,
  strict = TRUE
)
```

Arguments

x	Numeric vector or one-column matrix.
gamma	Boundary weight in [0, 1/2].
trim	Number of candidate observations removed from each boundary.
sigma	Optional positive known/null standard deviation.
variance	Scale estimator used when sigma = NULL.
calibration	Brownian-bridge calibration method.
calibration_draws	Number of intrinsic Gaussian calibration draws.
alpha	Test level.
seed	Optional calibration seed; the previous RNG state is restored.
strict	If TRUE, method failures error; otherwise they return an invalid object with estimate = NULL.

Details

If sigma is not supplied, the scale is either the sample standard deviation or Rice's adjacent-difference estimate. This software choice is returned explicitly because the book's classical display assumes a population σ and does not prescribe its estimator.

Value

An htest-compatible object with the complete CUSUM path.

References

Page (1954); Csorgo and Horvath (1997).

Examples

```
x <- c(-1, 0, 1, -1, 0, 3, 4, 3)
classical_cusum_test(x)
```

`classical_lda_classifier` *Classical plug-in linear discriminant classifier*

Description

Estimates the common covariance with the pooled unbiased estimator and inserts it into the Gaussian LDA log-likelihood ratio. The covariance must be strictly positive definite. No generalized inverse or ridge is used.

Usage

```
classical_lda_classifier(x, y, prior = "equal", strict = TRUE)
```

Arguments

`x`	Numeric matrix or all-numeric data frame, observations in rows.
`y`	Two-class label vector or factor.
`prior`	`"equal"`, `"empirical"`, or a positive numeric vector of length two. Named entries must match the class levels.
`strict`	If `TRUE`, a method failure is an error. If `FALSE`, it is a warning followed by an invalid `hd_classifier_fit`.

Value

An `hd_classifier_fit`.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Wiley.

Examples

```
x <- rbind(c(2, 0), c(1, 1), c(1, -1),
           c(-2, 0), c(-1, 1), c(-1, -1))
classical_lda_classifier(x, rep(c("A", "B"), each = 3))
```

`classical_pca` *Classical principal component analysis with explicit covariance convention*

Description

Computes PCA from the centered second-moment matrix with a user-visible divisor. The book's displayed estimator uses `covariance_divisor = "n"`; `"n-1"` gives the usual unbiased sample covariance. No variance floor or rank repair is applied. `components = NULL` returns all numerically usable positive-variance components.

Usage

```
classical_pca(
  x,
  components = NULL,
  center = c("mean", "none"),
  covariance_divisor = c("n-1", "n"),
  eigen_tol = sqrt(.Machine$double.eps)
)
```

Arguments

`x`	Numeric matrix or data frame with observations in rows.
`components`	Number of leading components. `NULL` uses the numerical rank certified at `eigen_tol`.
`center`	Either `"mean"`, `"none"`, or a supplied finite vector with one entry per variable. Named supplied centers are reordered to the data.
`covariance_divisor`	Either `"n-1"` or `"n"`.
`eigen_tol`	Positive relative tolerance for PSD, numerical-rank, and repeated-eigenvalue diagnostics.

Details

Eigenvector signs are anchored by making the first coordinate attaining the largest absolute loading positive. Individual eigenvectors inside a repeated-eigenvalue block are not identified; the returned projector is the invariant object for such a block.

Value

A `classical_pca_fit` and `hd_pca_fit` object containing loadings, scores, covariance operator, eigenvalues, projector, reconstruction, and complete centering/divisor diagnostics.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Wiley.

Examples

```
x <- rbind(c(2, 0), c(1, 1), c(-1, -1), c(-2, 0))
fit <- classical_pca(x, components = 1, covariance_divisor = "n")
fit$loadings
```

`classical_qda_classifier` *Classical plug-in quadratic discriminant classifier*

Description

Uses classwise unbiased covariance matrices with divisors `n1 - 1` and `n2 - 1`, then forms the canonical Gaussian QDA log-likelihood ratio. Both covariance matrices must be strictly positive definite. Ordinary positive determinants are used; no pseudo-determinant, absolute value, ridge, or generalized inverse is substituted.

Usage

```
classical_qda_classifier(x, y, prior = "equal", strict = TRUE)
```

Arguments

`x`	Numeric matrix or all-numeric data frame, observations in rows.
`y`	Two-class label vector or factor.
`prior`	`"equal"`, `"empirical"`, or a positive numeric vector of length two. Named entries must match the class levels.
`strict`	If `TRUE`, a method failure is an error. If `FALSE`, it is a warning followed by an invalid `hd_classifier_fit`.

Value

An `hd_classifier_fit`.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Wiley.

Examples

```
x <- rbind(c(2, 0), c(1, 1), c(1, -1), c(2, 2),
           c(-2, 0), c(-1, 2), c(-1, -2), c(-2, -1))
classical_qda_classifier(x, rep(c("A", "B"), each = 4))
```

`classical_spatial_tests` *Classical spatial sign and rank location tests*

Description

Implements the fixed-dimensional spatial-sign test, the one-sample spatial signed-rank test, and the pooled two-sample spatial-rank test. All three use the zero-direction convention $U(0) = 0$ and a Cholesky solve; a singular studentizing matrix is reported rather than replaced by a generalized inverse.

Usage

```
spatial_sign_test(x, mu = NULL, tol = sqrt(.Machine$double.eps), zero_tol = 0)

spatial_signed_rank_test(
  x,
  mu = NULL,
  tol = sqrt(.Machine$double.eps),
  zero_tol = 0
)

spatial_rank_test(x, y, tol = sqrt(.Machine$double.eps), zero_tol = 0)
```

Arguments

`x`	A numeric matrix or data frame with observations in rows.
`mu`	For a one-sample test, a finite null location with one value per column of `x`. `NULL` uses the zero vector.
`tol`	Reciprocal-condition-number tolerance for the Cholesky solve. It must be strictly between zero and one.
`zero_tol`	Non-negative tolerance, in the original data units, below which a residual, pair sum, or pair difference is mapped to zero.
`y`	For `spatial_rank_test()`, a second numeric matrix or data frame with the same variables as `x`.

Details

For centered observations $Z_i = X_i - \mu_0$, the sign statistic uses

$$\bar{U} = n^{-1} \sum_i U(Z_i), \quad B_U = n^{-1} \sum_i U(Z_i) U(Z_i)^T, \quad Q_{sign} = n \bar{U}^T B_U^{-1} \bar{U}.$$

The signed ranks are

$$R_i = n^{-1} \sum_j U(Z_i + Z_j).$$

Since the first-order projection of their average has covariance $4B_R$, where $B_R = n^{-1}\sum_i R_i R_i^T$, its statistic is

$$Q_{SR} = \frac{n}{4}\bar{R}^T B_R^{-1}\bar{R}.$$

The factor `1/4` is essential.

For two samples, every observation is ranked against the pooled sample:

$$R(Y_i) = N^{-1}\sum_j U(Y_i - Y_j).$$

If $C = (N-1)^{-1}\sum_i R(Y_i)R(Y_i)^T$, the sample covariance of the pooled ranks (whose average is exactly zero), the statistic is

$$Q_{2SR} = \frac{n_1 n_2}{N}(\bar{R}_1 - \bar{R}_2)^T C^{-1}(\bar{R}_1 - \bar{R}_2).$$

These calibrations are asymptotic chi-squared laws for fixed dimension, not high-dimensional approximations. Euclidean spatial signs make the tests invariant to translations, common positive rescaling, and orthogonal transformations, but not to arbitrary nonspherical affine transformations.

Value

An object of classes `hd_location_test` and `htest`. `statistic` is the chi-squared statistic, `parameter` is its asymptotic degrees of freedom, `variance` is the studentizing second-moment matrix (or `4 B_R` for the signed-rank test), and `components` contains the directional mean and unscaled moment matrices. Numerical rank, reciprocal condition number, solver, zero counts, and calibration type are in `diagnostics`.

References

Mottonen, J. and Oja, H. (1995). Multivariate spatial sign and rank methods. *Journal of Nonparametric Statistics*, **5**, 201–213.

Oja, H. (2010). *Multivariate Nonparametric Methods with R*. Springer.

Examples

```
set.seed(12)
x <- matrix(rnorm(80), 20, 4)
spatial_sign_test(x)
spatial_signed_rank_test(x)

y <- matrix(rnorm(96, 0.25), 24, 4)
spatial_rank_test(x, y)
```

clime_precision *Gaussian CLIME sparse precision estimator*

Description

Solves the Cai–Liu–Luo CLIME program column by column,

$$\min_b \|b\|_1 \quad \text{subject to} \quad \|S_n b - e_j\|_\infty \le \lambda,$$

using a primal–dual algorithm. Every raw column must pass primal feasibility, dual feasibility, lasso stationarity, and relative duality-gap certificates. The final symmetric estimator retains the entry of smaller absolute value from each transposed pair; exact ties retain the row-column entry before mirroring, making the convention deterministic.

Usage

```
clime_precision(
  x,
  lambda,
  center = TRUE,
  divisor = c("n", "n-1"),
  solver_tol = 1e-07,
  solver_max_iter = 100000L,
  strict = TRUE
)
```

Arguments

x	Numeric observation-by-variable matrix or data frame.
lambda	Finite non-negative off-diagonal lasso penalty on the correlation scale.
center	Whether to subtract column means.
divisor	Either "n" (formal default) or "n-1".
solver_tol	Positive tolerance for every feasibility and KKT certificate.
solver_max_iter	Positive ISP/ADMM iteration limit.
strict	If TRUE, a failed solver certificate is an error. If FALSE, the function warns and returns estimate = NULL with valid = FALSE and the uncertified last iterate in diagnostics.

Details

Primary CLIME does not guarantee that this post-LP symmetrization remains feasible, nor that the final matrix is positive definite. Both facts are reported and neither is repaired. Consequently valid certifies the raw column programs, not an invented SPD condition. SCIO and scaled-lasso precision estimation remain review-only because their additional programs are not specified by the short book display.

Value

A `gaussian_precision_fit` list. `estimate` is the primary smaller-absolute-value symmetrization of the certified raw columns.

References

Cai, T. T., Liu, W. and Luo, X. (2011). A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106, 594–607. doi:10.1198/jasa.2011.tm10155.

Examples

```
x <- rbind(c(-2, 0), c(-1, -1), c(1, 1), c(2, 0))
clime_precision(x, lambda = 0)
```

`composite_t2_two_sample_test`
Feng–Zou–Wang–Zhu Composite T-squared two-sample test

Description

Implements the two-sample Composite T^2 test of Feng, Zou, Wang, and Zhu (2017). Observations are rows and variables are columns. Despite the wording at `ch2_location.tex:778--784` in the current book draft, this is **not** a Behrens–Fisher test: the paper assumes that the two groups have the same unknown covariance matrix. Nor does the method combine existing test p-values or estimate correlations among component tests. Instead, it builds a block-diagonal approximation to the common pooled covariance and combines the resulting block Hotelling quadratic forms.

Usage

```
composite_t2_two_sample_test(x, y, block_size = 2L, selection = "paper_greedy")
```

Arguments

`x, y`	Numeric matrices or data frames with observations in rows and the same variables in columns. The first sample supplies the feasible trace calibration.
`block_size`	Positive integer block size K, no larger than the number of variables. The paper recommends small fixed blocks and uses 2 in its main implementation.
`selection`	Block construction rule. The only supported value is `"paper_greedy"`, the practical algorithm in Remark 1 of the paper.

Details

For a fixed leave-out covariance, the paper's practical block rule starts with the pair having the largest absolute sample correlation and repeatedly adds the variable with the largest sum of absolute correlations to the current block. Variables already assigned to a block are removed and the procedure is repeated. `selection = "paper_greedy"` implements that rule, with lexicographic tie-breaking. The final block contains all remaining variables when fewer than `block_size` remain. The paper's notation and simulations use equal-size blocks; this deterministic remainder convention extends the displayed rule to dimensions not divisible by the block size.

Let $\widehat{\Sigma}^{-1}_{O_K,i_1,i_2,j_1,j_2}$ be the block-diagonal inverse obtained after deleting observations i_1, i_2 from group 1 and j_1, j_2 from group 2, recomputing the unbiased pooled covariance and its correlation matrix, and rebuilding the blocks. The published statistic is

$$Q_n = \frac{1}{n_1 n_2 (n_1-1)(n_2-1)} \sum_{i_1 \neq i_2} \sum_{j_1 \neq j_2} (X_{1i_1} - X_{2j_1})^T \widehat{\Sigma}^{-1}_{O_K,i_1,i_2,j_1,j_2} (X_{1i_2} - X_{2j_2}).$$

The upper-tail normal calibration is

$$Z = \frac{Q_n}{2(n_1^{-1} + n_2^{-1})^2 \widehat{\mathrm{tr}(\Lambda_K^2)}^{1/2}}.$$

Following Theorem 2 and the paragraph immediately after it, the feasible trace estimate is the paper's one-sample leave-four-out estimator computed from **group 1 only**:

$$\widehat{\mathrm{tr}(\Lambda_K^2)} = \frac{1}{2P^4_{n_1}} \sum^{*} (X_{i_1} - X_{i_2})^T \widehat{\Sigma}^{-1}_{O_K,\setminus 4} (X_{i_3} - X_{i_4})(X_{i_1} - X_{i_4})^T \widehat{\Sigma}^{-1}_{O_K,\setminus 4} (X_{i_3} - X_{i_2}).$$

Thus Q_n and its population scaling are symmetric in the two samples, but the paper's feasible finite-sample standardisation is label-asymmetric. Exchanging the samples can change the reported Z and p-value.

The first group must satisfy `nrow(x) >= block_size + 5`: after leaving out four rows, a block of size `block_size` then has enough residual degrees of freedom to be nonsingular. The second group must contain at least three rows for the pooled 2+2 leave-out covariance. These count conditions are necessary, not sufficient. Every marginal variance used for selection and every selected full or leave-out covariance block must be finite and strictly positive definite. Failure produces an error. No ridge, generalized inverse, absolute-value repair, or variance floor is applied.

The C++ kernel uses sufficient statistics and sums unordered pairs and quadruples as exact symmetry reductions of the displayed ordered formulas. Both samples are internally translated by a common anchor and divided by a common positive scale in each column. This is algebraically neutral and protects the scale-invariant method under very large or small units.

Value

An object of class `c("hd_location_test", "htest")`. The statistic is the standardised Z and the p-value is its upper standard-normal tail probability. `components` retains Q_n, the group-1 leave-four-out trace estimate, every calibration coefficient and ordered denominator, combination counts, and a full-sample diagnostic partition. `diagnostics` records the common-covariance assumption, dynamic block selection, first-sample calibration, conditioning summaries, and the absence of numerical repair.

References

Feng, L., Zou, C., Wang, Z., and Zhu, L. (2017). Composite T-squared test for high-dimensional data. *Statistica Sinica*, **27**, 1419–1436. doi:10.5705/ss.202015.0199.

Examples

```
set.seed(2717)
x <- matrix(rnorm(21), 7, 3)
y <- matrix(rnorm(15, 0.2), 5, 3)
composite_t2_two_sample_test(x, y)
```

`conditional_alpha_sieve_design` *Construct a conditional-alpha sieve design from a supplied basis*

Description

This helper performs only the algebra specified by the conditional-alpha papers. Rows are time observations. If `center_alpha = TRUE`, its first block is the column-centered supplied basis; otherwise it is the original basis. The remaining blocks are, in factor order, the rowwise products $f_{jt}B(t/T)$. It does not choose spline order, knots, basis dimension, or a BIC rule.

Usage

```
conditional_alpha_sieve_design(
  basis,
  factors = NULL,
  center_alpha = TRUE,
  alpha_contrast = NULL
)
```

Arguments

`basis`	Finite observation-by-basis numeric matrix.
`factors`	`NULL`, a finite numeric vector, or an observation-by-factor numeric matrix.
`center_alpha`	Whether to center the alpha-basis block. Use `TRUE` for the restricted null fit and `FALSE` for the unrestricted residual fit required by the Zhao CSS trace estimator.
`alpha_contrast`	Optional finite matrix with `ncol(basis)` rows. The centered or uncentered alpha block is post-multiplied by this matrix.

Details

A normalized B-spline basis usually contains the constant function, so all of its centered columns are linearly dependent. The papers write an ordinary inverse despite this identity. This function never silently drops a column: pass an explicit full-column-rank `alpha_contrast`, or supply a reduced basis. The contrast changes only the coordinates, not the spanned centered-alpha space, when it has the intended range.

Value

A full-column-rank design matrix carrying its basis/factor metadata.

References

Ma, S., Lan, W., Su, L. and Tsai, C.-L. (2020). Testing alphas in conditional time-varying factor models with high-dimensional assets. *Journal of Business & Economic Statistics*, 38, 214–227. doi:10.1080/07350015.2018.1482758.

Examples

```
tt <- seq(0, 1, length.out = 12)
basis <- cbind(tt, tt^2)
factors <- cbind(market = sin(seq_len(12)))
conditional_alpha_sieve_design(basis, factors)
```

---

conditional_alpha_sieve_fit

Fit the restricted conditional-alpha sieve regression

Description

Fits, independently for every asset, the no-intercept null regression on a supplied full-rank sieve design. The residualized intercept $h = M_Z 1_T$ is retained because it enters every feasible calibration. The function uses one common response scale and independent design-column scales for numerical stability; both leave the fitted residual subspace and every reported test unchanged. It never uses a generalized inverse or ridge.

Usage

```
conditional_alpha_sieve_fit(returns, design)
```

Arguments

`returns`	Finite observation-by-asset numeric matrix or data frame, with at least two assets.
`design`	A finite full-column-rank observation-by-regressor matrix, normally produced by `conditional_alpha_sieve_design()`.

Value

A reusable `conditional_alpha_sieve_fit` object containing the restricted residuals, `h`, design metadata, and projection diagnostics.

References

Ma, S., Lan, W., Su, L. and Tsai, C.-L. (2020). Testing alphas in conditional time-varying factor models with high-dimensional assets. *Journal of Business & Economic Statistics*, 38, 214–227. doi:10.1080/07350015.2018.1482758.

Examples

```
tt <- seq(0, 1, length.out = 14)
z <- conditional_alpha_sieve_design(cbind(tt, tt^2))
y <- cbind(sin(1:14), cos(1:14), sin(1:14 / 2))
conditional_alpha_sieve_fit(y, z)
```

`conditional_factor_wald_test` *Fixed-dimensional conditional-factor Wald benchmark*

Description

Evaluates the Chapter 4 supplied-estimate benchmark

$$W = T\hat{\delta}^T\hat{\Omega}_\delta^{-1}\hat{\delta},$$

with a chi-squared reference distribution having `length(delta)` degrees of freedom. This function does not estimate spline or kernel nuisances.

Usage

```
conditional_factor_wald_test(delta, covariance, sample_size)
```

Arguments

`delta`	Finite estimated conditional-alpha vector.
`covariance`	Finite, exactly symmetric, strictly positive-definite covariance matrix for `delta`.
`sample_size`	Positive integer T.

Value

A strict Cholesky-based upper-tail Wald `htest` object.

References

Li, D. and Yang, L. (2011). Nonparametric tests of conditional factor models. Ang, A. and Kristensen, D. (2012). Testing conditional factor models.

Examples

```
conditional_factor_wald_test(c(0.1, -0.2), diag(c(1, 2)), 40)
```

`dsda_classifier` *Direct sparse discriminant analysis*

Description

Fits the Mai–Zou–Yuan DSDA lasso with the exact response coding $-n/n_1$ for class 1 and $+n/n_2$ for class 2 and objective

$$n^{-1}\sum_i (y_i - a - x_i^\top \beta)^2 + \lambda\|\beta\|_1.$$

The regression coefficient points toward class 2. The package reverses it only when constructing its documented score, so positive always means class 1. Following the primary equal-prior rule, classification uses the midpoint intercept rather than the penalized regression intercept. The coordinate-descent solution must pass both KKT and Fenchel dual-gap checks.

Usage

```
dsda_classifier(
  x,
  y,
  lambda,
  prior = c(0.5, 0.5),
  solver_tol = 1e-07,
  solver_max_iter = 100000L,
  strict = TRUE,
  tie = c("class1", "class2")
)
```

Arguments

`x`	Numeric training matrix with observations in rows.
`y`	Binary response. Its first factor level is class 1.
`lambda`	Positive Dantzig constraint radius; it is never selected implicitly.
`prior`	Equal class probabilities. Non-equal probabilities are rejected in this first implementation because the primary DSDA classification intercept has a scale-dependent unequal-prior correction.
`solver_tol`	Positive tolerance required by every solver certificate.
`solver_max_iter`	Positive maximum number of primal–dual iterations.
`strict`	If TRUE, a failed certificate is an error. If FALSE, an explicitly invalid, non-predictable fit is returned with a warning.
`tie`	Which class receives a score exactly equal to zero.

Value

An object of class hd_classifier_fit.

References

Mai, Q., Zou, H. and Yuan, M. (2012). A direct approach to sparse discriminant analysis in ultra-high dimensions. Biometrika, 99, 29–42. doi:10.1093/biomet/asr066.

Examples

```
x <- rbind(
  c(2, 1), c(1, 2), c(2, 2), c(3, 1),
  c(-2, -1), c(-1, -2), c(-2, -2), c(-3, -1)
)
y <- factor(rep(c("first", "second"), each = 4))
dsda_classifier(x, y, lambda = 0.2)
```

`ec2_covariance` *EC2 sparse covariance estimation with an eigenvalue constraint*

Description

Implements the convex off-diagonal-lasso branch of Liu, Wang and Zhao's EC2 estimator. The primary method first forms the empirical correlation matrix R, solves

$$\min_{C:\,\mathrm{diag}(C)=1,\,\lambda_{\min}(C)\geq\tau} \frac{1}{2}\|R-C\|_F^2+\lambda\sum_{i\neq j}|c_{ij}|,$$

and restores marginal empirical standard deviations. This differs from the book's shortened direct-covariance display. The default covariance divisor is n; selecting `"n-1"` is explicit and changes only the marginal covariance scale, not the sample correlation.

Usage

```
ec2_covariance(
  x,
  lambda,
  tau,
  penalty = "l1",
  center = TRUE,
  divisor = c("n", "n-1"),
  rho = 1,
  solver_tol = 1e-07,
  solver_max_iter = 50000L,
  strict = TRUE
)
```

Arguments

x	Numeric observation-by-variable matrix or data frame.
lambda	Finite non-negative off-diagonal lasso penalty on the correlation scale.
tau	Finite minimum correlation-eigenvalue bound in $(0, 1]$.
penalty	Currently only "l1". Adaptive and MC+ variants are deliberately review-only.
center	Whether to subtract column means.
divisor	Either "n" (formal default) or "n-1".
rho	Positive ISP/ADMM penalty multiplier.
solver_tol	Positive tolerance for every feasibility and KKT certificate.
solver_max_iter	Positive ISP/ADMM iteration limit.
strict	If TRUE, a failed solver certificate is an error. If FALSE, the function warns and returns estimate = NULL with valid = FALSE and the uncertified last iterate in diagnostics.

Details

The ISP/ADMM solver uses the paper's soft-threshold and spectral-projection updates. A fit is valid only when the equality, fixed-point, off-diagonal subgradient, spectral-dual, complementarity, diagonal, and eigenvalue certificates all pass. The method-defining tau projection is not a numerical repair. Adaptive EC2 and MC+ EC2 require additional weight or shape choices and remain review-only; this function never guesses them.

Value

An ec2_covariance_fit list. estimate is the covariance matrix only for a certified fit; correlation is its EC2 correlation estimate.

References

Liu, H., Wang, L. and Zhao, T. (2014). Sparse covariance matrix estimation with eigenvalue constraints. *Journal of Computational and Graphical Statistics*, 23, 439–459. doi:10.1080/10618600.2013.782818.

Examples

```
x <- rbind(c(-2, -1), c(-1, 0), c(1, 0), c(2, 1))
ec2_covariance(x, lambda = 0.2, tau = 0.1)
```

`elliptical_factor_number` *Select the number of spikes in an elliptical factor model*

Description

Computes the trace-p spatial-sign pilot centered at the sample spatial median and applies the eigenvalue-ratio (ER) or growth-ratio (GR) selector of Xu, Ma, Wang and Feng. For ordered eigenvalues λ_j,

$$\widehat{m}_{ER} = \arg\max_{1 \le j \le M} \lambda_j / \lambda_{j+1}.$$

The GR criterion uses $V_j = \sum_{l=j+1}^{\min(n,p)-1} \lambda_l$ exactly as in the primary paper. `max_factors` is deliberately supplied: the paper treats M as a predetermined upper bound and its numerical choice is not a universal tuning rule.

Usage

```
elliptical_factor_number(
  x,
  max_factors,
  method = c("er", "gr"),
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

`x`	Numeric matrix or data frame with observations in rows.
`max_factors`	Predetermined positive integer upper bound M.
`method`	Either eigenvalue ratio (`"er"`) or growth ratio (`"gr"`).
`tol`, `max_iter`, `zero_tol`	Spatial-median controls.
`strict`	If `TRUE`, fail when the spatial median is not certified; otherwise return an explicitly invalid diagnostic object.

Value

An `elliptical_factor_number` object containing the selected count, all pilot eigenvalues, criterion values, and spatial-median diagnostics.

References

Xu, X., Ma, H., Wang, H. and Feng, L. (2025). High dimensional matrix estimation through elliptical factor models. arXiv:2512.19325. https://arxiv.org/abs/2512.19325.

Examples

```
x <- rbind(c(-3, -2, 0), c(-2, -1, 1), c(-1, 0, -1),
           c(1, 0, 1), c(2, 1, -1), c(3, 2, 0))
elliptical_factor_number(x, max_factors = 1)
```

`elliptical_factor_precision` *Sparse precision estimation for an elliptical factor fit*

Description

Applies the primary CLIME or GLASSO problem to the *raw* idiosyncratic complement of a `spatial_sign_poet()` or `poet_tme()` fit, and reconstructs the full inverse scatter through the Woodbury identity,

$$V = V_u - V_u\Gamma(\Lambda^{-1} + \Gamma^T V_u \Gamma)^{-1}\Gamma^T V_u.$$

CLIME must pass per-column primal, dual, stationarity, gap, and final symmetrized-feasibility certificates. GLASSO must remain positive definite and satisfy its full elementwise-ℓ_1 KKT equation. The tuning parameter is explicit because the theoretical constant multiplying the rate is not an observable default.

Usage

```
elliptical_factor_precision(
  fit,
  lambda,
  method = c("sclime", "sglasso"),
  solver_tol = 1e-07,
  solver_max_iter = 100000L,
  initial_step = 1,
  max_backtracking = 100L,
  strict = TRUE
)
```

Arguments

`fit`	A valid `elliptical_factor_fit`.
`lambda`	Positive CLIME constraint or GLASSO penalty.
`method`	Either `"sclime"` or `"sglasso"`.
`solver_tol`, `solver_max_iter`, `initial_step`, `max_backtracking`	Solver controls; see `spatial_sign_precision()`.
`strict`	If `TRUE`, fail on any solver or Woodbury certificate failure. If `FALSE`, return `estimate = NULL` and explicit diagnostics.

Value

An `elliptical_factor_precision_fit` containing the idiosyncratic and full precision estimates and complete solver certificates.

References

Xu, X., Ma, H., Wang, H. and Feng, L. (2025). arXiv:2512.19325.

Examples

```
x <- rbind(c(-3, -2), c(-2, -1), c(-1, 1),
           c(1, -1), c(2, 1), c(3, 2))
f <- spatial_sign_poet(x, factors = 0, threshold = 0.2)
elliptical_factor_precision(f, lambda = 0.5, method = "sglasso")
```

`elliptical_oracle_classifier` *Exact oracle classifier for two elliptical populations*

Description

Computes the exact generator-aware score

$$\log \pi_1 - \tfrac{1}{2} \log |\Lambda_1| + \log g_1(d_1) - \log \pi_2 + \tfrac{1}{2} \log |\Lambda_2| - \log g_2(d_2),$$

where $d_k = (z - \mu_k)' \Lambda_k^{-1} (z - \mu_k)$. The log-generator functions must be vectorized. They may return `-Inf` for a zero density, but not `NA`, `NaN`, or `Inf`.

Usage

```
elliptical_oracle_classifier(
  location1,
  location2,
  shape1,
  shape2,
  log_generator1,
  log_generator2 = log_generator1,
  prior = "equal",
  levels = c("class1", "class2"),
  feature_names = NULL
)
```

Arguments

`location1`, `location2`	Finite class locations.
`shape1`, `shape2`	Positive-definite class shape matrices, on the scales used by the corresponding generators.
`log_generator1`, `log_generator2`	Vectorized functions evaluating $\log g_k(d)$. By default the second generator equals the first.
`prior`	`"equal"` or a positive numeric vector of length two.
`levels`	Two class labels.
`feature_names`	Optional unique feature names.

Details

A common generator and common shape do not generally reduce this rule to a midpoint linear rule under unequal priors. That shortcut is exact for equal priors, and for the exponential radial generator underlying Gaussian LDA.

Value

A generator-aware `hd_classifier_fit`.

References

Fang, K.-T. and Anderson, T. W. (1990). *Statistical Inference in Elliptically Contoured and Related Distributions*. Allerton Press. Wakaki, H. (1994). Discriminant analysis under elliptical populations. *Hiroshima Mathematical Journal*, 24, 257–298.

Examples

```
normal_log_generator <- function(d) -d / 2
fit <- elliptical_oracle_classifier(
  c(1, 0), c(-1, 0), diag(2), diag(2), normal_log_generator
)
predict(fit, rbind(c(1, 0), c(-1, 0)))
```

---

elliptical_regularized_hotelling_cauchy_test

Elliptical regularized Hotelling test with Cauchy aggregation

---

Description

Computes the feasible ERHT statistic over a fixed deterministic ridge grid and combines its upper-tail normal p-values with the analytic Cauchy rule of Feng, Zhou, and Wang (2026). Equal weights are used by default. The default grid $\{0.1, 0.2, \ldots, 1\}$ is the grid used in the paper's numerical implementation; this function does not reproduce any simulation design from that paper.

Usage

```
elliptical_regularized_hotelling_cauchy_test(
  x,
  mu = NULL,
  rho = seq(0.1, 1, by = 0.1),
  weights = NULL,
  alpha = 0.05,
  tol = 1e-08,
  max_iter = 1000L,
  strict = TRUE,
  keep_companion = FALSE
)
```

Arguments

x	Numeric matrix or data frame with observations in rows. At least three observations and two variables are required.
mu	Null location vector. NULL means the zero vector.
rho	A strictly increasing vector of at least two positive ridge values.
weights	Optional prespecified positive Cauchy weights, one per ridge. They must be deterministic and chosen independently of the observed test statistics, as required by the fixed-grid result. They are normalized to sum to one. NULL gives equal weights.
alpha	Significance level for the upper-tail rejection rule.
tol, max_iter	Convergence controls for the sample spatial median.
strict	If TRUE, fail when the spatial median does not converge; otherwise warn and continue only when all ERHT quantities remain defined.
keep_companion	If TRUE, retain the full $n \times n$ companion matrix. The default retains only its eigenvalues and scalar functionals.

Details

For positive weights ϖ_k summing to one,

$$T_{CC} = \sum_k \varpi_k \tan[\pi\{1/2 - p_k\}], \qquad p_{CC} = 1/2 - \pi^{-1} \arctan(T_{CC}).$$

The computation uses signed-log cotangents and an angle representation to avoid overflow and cancellation in extreme tails. The article proves dependence-robust small-tail validity for this analytic p-value; except in special dependence structures it does not claim exact fixed-level calibration at an ordinary level such as 0.05.

Value

An object of class c("hd_location_test", "htest"). Its p-value is the analytic Cauchy combination. The marginal table contains every raw ERHT statistic, feasible center and variance, standardized statistic, p-value, log-p-value, and fixed-ridge decision.

References

Feng, L., Zhou, L., and Wang, X. (2026). Elliptical regularized Hotelling testing for high dimensional data. arXiv:2606.25942. doi:10.48550/arXiv.2606.25942.

Examples

```
set.seed(2)
x <- matrix(rnorm(160), 40, 4)
elliptical_regularized_hotelling_cauchy_test(
  x, rho = c(0.2, 0.5, 1)
)
```

`elliptical_regularized_hotelling_test` *Elliptical regularized Hotelling fixed-ridge test*

Description

Tests a high-dimensional location vector under elliptical symmetry and pervasive dependence using the fixed-ridge ERHT statistic of Feng, Zhou, and Wang (2026). Observations are rows and variables are columns.

Usage

```
elliptical_regularized_hotelling_test(
  x,
  mu = NULL,
  rho = 0.5,
  alpha = 0.05,
  tol = 1e-08,
  max_iter = 1000L,
  strict = TRUE,
  keep_companion = FALSE
)
```

Arguments

`x`	Numeric matrix or data frame with observations in rows. At least three observations and two variables are required.
`mu`	Null location vector. `NULL` means the zero vector.
`rho`	One finite, strictly positive ridge value.
`alpha`	Significance level for the upper-tail rejection rule.
`tol`, `max_iter`	Convergence controls for the sample spatial median.
`strict`	If `TRUE`, fail when the spatial median does not converge; otherwise warn and continue only when all ERHT quantities remain defined.
`keep_companion`	If `TRUE`, retain the full $n \times n$ companion matrix. The default retains only its eigenvalues and scalar functionals.

Details

Let $\widehat{\theta}$ be the sample spatial median, $\widehat{Y}_i = \sqrt{p}\, U(X_i - \widehat{\theta})$, and

$$\widehat{R}_n = n^{-1} \sum_i \widehat{Y}_i \widehat{Y}_i^{\mathsf{T}}.$$

For $\rho > 0$, the raw statistic is

$$T_n(\rho) = n(\widehat{\theta} - \theta_0)^{\mathsf{T}} (\widehat{R}_n + \rho I)^{-1} (\widehat{\theta} - \theta_0).$$

The function implements the paper's direct feasible companion-matrix centering $n\widehat{\mu}_n(\rho)$ and variance $n\widehat{\sigma}^2_{D,n}(\rho)$, returning the upper-tail normal calibration of

$$Z_n(\rho) = \{T_n(\rho) - n\widehat{\mu}_n(\rho)\}/\{n\widehat{\sigma}^2_{D,n}(\rho)\}^{1/2}.$$

The C++ kernel uses one economy SVD of the sign matrix. It evaluates the ridge quadratic in row and null spaces and switches between the companion matrix A and its complement $I-A$ for the feasible calibration. Thus it never forms a persistent $p \times p$ inverse or subtracts nearly equal Woodbury, diagonal-weight, or companion terms. A zero fitted residual makes the required inverse distance undefined and is reported as an error; no observation is omitted or perturbed and no additional ridge, pseudoinverse, absolute-value repair, or variance floor is used.

The paper's numerical section mentions a separate Bartlett center correction with $a = 8n/p$, but neither the article nor its public arXiv source specifies the formula that maps a into the feasible center. This function therefore implements the fully stated equations only and deliberately does not guess that simulation-program modification.

Value

An object of class `c("hd_location_test", "htest")`. The reported statistic is $Z_n(\rho)$; `raw.statistic` is $T_n(\rho)$. Components include all feasible functionals, raw and internally scaled values, inverse distances, spatial-median diagnostics, and companion eigenvalues. Raw dimensional fields can underflow to zero or overflow to infinity when their physical units exceed the double-precision range; the separately normalized fields in `components$scaled` remain the numerical-audit representation used to compute the reported statistic.

References

Feng, L., Zhou, L., and Wang, X. (2026). Elliptical regularized Hotelling testing for high dimensional data. arXiv:2606.25942. doi:10.48550/arXiv.2606.25942.

Examples

```
set.seed(1)
x <- matrix(rnorm(120), 30, 4)
elliptical_regularized_hotelling_test(x, rho = 0.5)
```

---

erht_change_point_test

Elliptical regularized Hotelling change-point scan

Description

Implements the single-change path or the discretized adjacent-triple scan of Song, Wen, and Feng (2026). For adjacent segments with sizes `n1`, `n2`, the raw statistic is

$$V_\rho^{raw} = N\widehat{\Delta}^T(\widehat{R} + \rho I)^{-1}\widehat{\Delta}, \qquad N = n_1 n_2/(n_1 + n_2).$$

The common pool is the full sample. Its centered spatial signs are columns of Y; with $A = Y^T QY/n$, the primary studentization is

$$\widehat{\kappa} = \sum_i \beta_i^2 A_{ii}, \qquad \widehat{\sigma}^2 = 2n \sum_{i \neq j} \beta_i^2 \beta_j^2 A_{ij}^2,$$

$$Z_\rho = (V_\rho^{raw} - n\widehat{\kappa})/\{n\widehat{\sigma}^2\}^{1/2}.$$

The ordered off-diagonal variance is accumulated directly. It is never obtained by flooring a cancellation-prone difference.

Usage

```
erht_change_point_test(
  x,
  ridge,
  scan = c("single", "multiple"),
  epsilon = 0.1,
  calibration = c("gaussian-supremum", "time-permutation", "none"),
  calibration_draws = 4999L,
  alpha = 0.05,
  seed = NULL,
  tol = 1e-08,
  max_iter = 1000L,
  zero_tol = 0,
  cauchy_weights = NULL,
  keep_calibration = FALSE,
  strict = TRUE
)
```

Arguments

`x`	Numeric `n` by `p` data matrix.
`ridge`	Positive actual ridge value or finite grid.
`scan`	`"single"` or the primary discretized `"multiple"` adjacent-triple scan.
`epsilon`	Trimming/minimum-segment fraction in `(0, 1/2)`; for `scan = "multiple"` it is also the primary grid spacing.
`calibration`	Marginal Gaussian-supremum, time-permutation, or no p-value calibration.
`calibration_draws`	Number of intrinsic null draws/permutations.
`alpha`	Test level.
`seed`	Optional calibration seed; previous RNG state is restored.
`tol`, `max_iter`, `zero_tol`	Spatial-median controls.
`cauchy_weights`	Optional positive ridge-combination weights.
`keep_calibration`	Whether to retain simulated marginal null maxima.
`strict`	Failure contract. No invalid candidate is silently removed.

Details

`ridge` contains the actual positive ρ values in the primary formula. The public ERHTCP reproduction code instead accepts empirical ratios `rho / (p/n)`; this function does not silently make that conversion. Users who want that convention can supply `ridge_ratio * p / n` explicitly, and all resolved ridge values are returned.

`calibration = "gaussian-supremum"` simulates the primary marginal Gaussian-process limit on the exact candidate grid. `"time-permutation"` is the paper's practical exchangeability calibration and preserves each multivariate row. It is invalid under unaddressed serial dependence. With multiple ridge values, the returned Cauchy transform combines the marginal p-values, but `calibration.exact` is `FALSE`: the paper explicitly distinguishes this analytic rule from exact calibration by a joint-limit quantile involving the unknown cross-ridge correlation `r_E`.

The primary inverse-distance average is exactly `mean(sqrt(p) / distance)`, with a coincident observation assigned weight zero. The optional `(p - 1) / p` factor used by the public reproduction repository is not part of the displayed primary statistic and is not used.

Value

An `htest` object with every local raw/center/variance/Z component.

References

Song, Wen and Feng (2026), arXiv:2607.22162.

Examples

```
x <- rbind(
  c(-2, 0), c(2, 0), c(0, -2), c(0, 2),
  c(-1, -1), c(1, 1), c(-1, 1), c(1, -1),
  c(-2, 1), c(2, -1), c(-1, 2), c(1, -2)
)
erht_change_point_test(
  x, ridge = 0.5, epsilon = 0.25,
  calibration_draws = 99, seed = 1
)
```

---

erht_wbs *Wild binary segmentation with the ERHT local score*

Description

Implements the primary WBS–ERHT algorithm of Song, Wen, and Feng (2026). On an interval `I`, every (or, when `triple_step > 1`, an explicitly thinned) adjacent triple compares two segments of length at least `ceiling(epsilon * length(I))`; the common scatter pool is the whole interval. Scores are maximized over the supplied actual ridge grid.

Usage

```
erht_wbs(
  x,
  ridge,
  threshold,
  min_interval,
  refinement_radius,
  deletion_radius,
  intervals = NULL,
  M = NULL,
  epsilon = 0.1,
  triple_step = 1L,
  max_changes = Inf,
  seed = NULL,
  tol = 1e-08,
  max_iter = 1000L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

`x`	Numeric `n` by `p` data matrix.
`ridge`	Positive actual ERHT ridge value or grid.
`threshold`	Required strict WBS score threshold.
`min_interval`	Required minimum recursive/WBS interval length.
`refinement_radius`	Required non-negative local half-window radius.
`deletion_radius`	Required non-negative recursion deletion radius.
`intervals`	Optional two-column integer matrix of WBS intervals.
`M`	Required number of uniformly sampled intervals when `intervals` is `NULL`.
`epsilon`	Primary minimum adjacent-segment fraction.
`triple_step`	Candidate endpoint thinning step; one is exact.
`max_changes`	Optional explicit software stopping cap; `Inf` follows the primary recursion until no interval crosses the threshold.
`seed`	Optional random-interval seed; previous RNG state is restored.
`tol`, `max_iter`, `zero_tol`	Spatial-median controls.
`strict`	Failure contract. With `FALSE`, a method failure returns `estimate = NULL`, never a partially certified segmentation.

Details

The recursion uses the paper's narrowest-over-threshold rule: among sampled intervals contained in the current segment, plus the current segment itself, retain those with score strictly greater than `threshold`; choose the narrowest, break a length tie by the larger score, refine within the requested radius, delete the requested neighborhood, and recurse.

The primary theory does not choose the ridge grid, WBS threshold, number of intervals, minimum length, refinement radius, or deletion radius. Hence this API requires those quantities (or explicit intervals) rather than inventing automatic tuning constants. Random WBS intervals are generated independently of the observations from uniformly sampled unordered endpoint pairs. Scanning/WBS is intrinsic to the estimator and is not replication of a paper simulation study.

`triple_step = 1` is the literal exhaustive adjacent-triple collection. Larger values are an explicit computational approximation and are recorded. Any invalid spatial median or local variance invalidates the fit; failed triples are never silently removed.

Value

A change-point fit with sorted estimates and selection history.

References

Song, Wen and Feng (2026), arXiv:2607.22162, Algorithm 1.

Examples

```
x <- rbind(
  c(-2, 0), c(2, 0), c(0, -2), c(0, 2),
  c(-1, -1), c(1, 1), c(-1, 1), c(1, -1),
  c(-2, 1), c(2, -1), c(-1, 2), c(1, -2)
)
erht_wbs(
  x, ridge = 0.5, threshold = 10,
  min_interval = 8, refinement_radius = 3, deletion_radius = 1,
  intervals = matrix(c(1, 12), ncol = 2), epsilon = 0.25
)
```

`fair_classifier` *Features annealed independence rule (FAIR)*

Description

Implements Fan and Fan's FAIR classifier. Features are ranked by the absolute Welch two-sample statistic

$$T_j = (\bar{X}_j - \bar{Y}_j)/\sqrt{S_{1j}^2/n_1 + S_{2j}^2/n_2},$$

while classification uses the marginal variance $(S_{1j}^2 + S_{2j}^2)/2$. Feature-count selection by `"paper"` maximizes equation (4.3), including the largest eigenvalue of every truncated pooled within-class correlation matrix. Ties in absolute t-statistics are broken by the original feature index, and a tie in the criterion selects the smallest feature count.

Usage

```
fair_classifier(
  x,
  y,
  selection = c("paper", "m", "threshold"),
  m = NULL,
  t_threshold = NULL,
  prior = "equal",
  zero_variance = c("error", "drop"),
  strict = TRUE
)
```

Arguments

x	Numeric matrix or all-numeric data frame, observations in rows.
y	Two-class label vector or factor.
selection	Select an explicit feature count, an explicit absolute t-statistic threshold, or the primary paper criterion.
m	Positive feature count used only by `selection = "m"`.
t_threshold	Non-negative threshold used only by `selection = "threshold"`; the comparison is strict.
prior	`"equal"`, `"empirical"`, or a positive numeric vector of length two. Named entries must match the class levels.
zero_variance	Whether non-positive marginal variances are errors or are explicitly dropped.
strict	If `TRUE`, a method failure is an error. If `FALSE`, it is a warning followed by an invalid `hd_classifier_fit`.

Details

FAIR is implemented under its equal-prior common-covariance contract. An exactly zero marginal variance is never floored: it is either an error or is dropped explicitly.

Value

An `hd_classifier_fit`.

References

Fan, J. and Fan, Y. (2008). High-dimensional classification using features annealed independence rules. *Annals of Statistics*, 36, 2605–2637. doi:10.1214/07AOS504.

Examples

```
x <- rbind(c(3, 1, 0), c(2, 2, 1), c(4, 0, -1),
           c(-3, -1, 0), c(-2, -2, -1), c(-4, 0, 1))
fair_classifier(x, rep(c("A", "B"), each = 3), selection = "m", m = 1)
```

fantope_pca *Fantope projection and selection sparse PCA*

Description

Solves

$$\max_H \langle S, H\rangle - \tau\|H\|_{1,1}, \quad 0 \preceq H \preceq I, \quad \mathrm{tr}(H) = r$$

by a certified two-block ADMM. The `H` update is the Euclidean Fantope projection and the split update is entrywise soft thresholding. The returned `relaxed.projector` is the primary convex estimate; `loadings` are its leading `rank` eigenvectors and are not claimed to be individually identified across repeated eigenvalues.

Usage

```
fantope_pca(
  x,
  rank,
  tau,
  center = c("mean", "none"),
  scale = FALSE,
  covariance_divisor = c("n", "n-1"),
  rho = 1,
  solver_tol = 1e-07,
  solver_max_iter = 5000L,
  symmetry_tol = sqrt(.Machine$double.eps),
  strict = TRUE,
  keep_operator = TRUE
)
```

Arguments

`x`	Numeric data with observations in rows.
`rank`	Required Fantope trace/rank.
`tau`	Required non-negative entrywise lasso penalty.
`center`	Numeric center or `"mean"`/`"none"`.
`scale`	Logical or supplied positive scale vector.
`covariance_divisor`	`"n"` (book convention) or `"n-1"`.
`rho`	Positive ADMM penalty.
`solver_tol`	Positive scaled primal/dual residual tolerance.
`solver_max_iter`	Positive ADMM iteration limit.
`symmetry_tol`	Positive operator certification tolerance.
`strict`	If `TRUE`, failure is an error; otherwise return an invalid fit.
`keep_operator`	Retain the covariance operator.

Value

A `fantope_pca_fit` inheriting from `hd_pca_fit`.

References

Vu, V. Q., Cho, J., Lei, J., and Rohe, K. (2013). Fantope projection and selection: a near-optimal convex relaxation of sparse PCA. *NeurIPS 26*.

Examples

```
x <- rbind(c(3, 0), c(-3, 0), c(0, 1), c(0, -1))
fantope_pca(x, rank = 1, tau = 0.1)
```

---

feng_jiang_liu_xiong_panel_independence_test

Feng-Jiang-Liu-Xiong max-sum panel-independence test

Description

Implements the three procedures in Feng, Jiang, Liu and Xiong (2022) for serially uncorrelated panel errors. For OLS residual correlations $\hat{\rho}_{ij}$,

$$S_N = \sum_{i<j} T\hat{\rho}_{ij}^2, \qquad L_N = \max_{i<j} |\hat{\rho}_{ij}|.$$

The sum component is $(S_N - \mu_N)/N$, where

$$\mu_N = \frac{T}{(T-p)^2} \sum_{i<j} \operatorname{tr}(P_i P_j),$$

and the max component is $TL_N^2 - 4\log N + \log\log N$. Their primary max-sum statistic is $C_N = \min(p_L, p_S)$ with calibrated p-value $2C_N - C_N^2$.

Usage

```
feng_jiang_liu_xiong_panel_independence_test(
  panel,
  regressors = NULL,
  component = c("max-sum", "max", "sum"),
  keep_correlations = FALSE
)
```

Arguments

`panel`	Numeric T by N matrix. With `regressors = NULL`, columns are treated as already-computed residual vectors. Otherwise they are unit outcomes and OLS residuals are computed internally.
`regressors`	`NULL`, one common T by p design matrix, or a list of N unit-specific T by p full-rank designs. The designs must have the same p; no intercept is added implicitly.
`component`	One of `"max-sum"`, `"max"`, or `"sum"`.
`keep_correlations`	Whether to retain the residual-correlation matrix.

Value

An object inheriting from `htest`, with all three component statistics and p-values retained in `components`.

References

Feng, L., Jiang, T., Liu, B. and Xiong, W. (2022). Max-Sum Tests for Cross-Sectional Independence of High-Dimensional Panel Data. *Annals of Statistics*, 50, 1124-1143. doi:10.1214/21AOS2142

Examples

```
panel <- matrix(c(-2, 1, 0, 2, -1, 3, 1, -2,
                  2, 1, -3, 1, 3, -1, 2, -2), 4, 4)
feng_jiang_liu_xiong_panel_independence_test(panel)
```

`feng_lan_liu_ma_alpha_max_test` *Feng–Lan–Liu–Ma high-dimensional maximum alpha test*

Description

Computes the maximum squared unrestricted OLS intercept t statistic, $M = \max_i t_i^2$, and its centered value $M - 2\log N + \log\log N$. The limiting cdf is $\exp\{-\pi^{-1/2}\exp(-x/2)\}$. The upper tail is evaluated without subtracting a cdf numerically.

Usage

```
feng_lan_liu_ma_alpha_max_test(returns, factors = NULL)
```

Arguments

`returns`	Finite observation-by-asset numeric matrix or data frame.
`factors`	NULL, a finite length-T numeric vector, or a finite observation-by-factor numeric matrix or data frame.

Value

An asymptotic Gumbel-calibrated upper-tail alpha-test object.

References

Feng, L., Lan, W., Liu, B., and Ma, Y. (2022). High-dimensional test for alpha in linear factor pricing models. Journal of Econometrics. doi:10.1016/j.jeconom.2021.07.011.

Examples

```
f <- cbind(seq(-1, 1, length.out = 12))
y <- outer(seq_len(12), 1:3, function(i, j) cos(i + 2 * j))
feng_lan_liu_ma_alpha_max_test(y, f)
```

---

feng_liu_ma_white_noise_test

Feng–Liu–Ma high-dimensional white-noise test

Description

Implements the feasible max statistic, the diagonal-deleted U-statistic sum test, and their Fisher combination from Feng, Liu and Ma. The primary paper uses `n` (not `n - h`) in every lagged sample covariance and uses the ordered-pair denominator `n * (n - 1)` in both the sum statistic and the feasible trace estimate.

Usage

```
feng_liu_ma_white_noise_test(
  x,
  lag = 1L,
  component = c("fisher", "sum", "max"),
  center = c("none", "mean"),
  keep_lag = FALSE
)
```

Arguments

`x`	Numeric matrix with time points in rows and coordinates in columns.
`lag`	Positive lag truncation level, no larger than `n - 2`.
`component`	Which calibrated result supplies the top-level statistic and p-value: `"fisher"`, `"sum"`, or `"max"`. All three are returned in `components` regardless of this choice.
`center`	`"none"` reproduces the mean-zero primary definition. `"mean"` subtracts column sample means as an explicit preprocessing step; the paper's null theorem does not account for estimated means.
`keep_lag`	Whether to retain lag-specific maxima and sum numerators.

Value

An `htest` object with raw feasible components and diagnostics.

References

Feng, L., Liu, B. and Ma, Y. Testing for High-Dimensional White Noise. Statistica Sinica. doi:10.5705/ss.202023.0300.

Examples

```
t <- seq_len(18)
x <- cbind(sin(t), cos(t / 2), sin(t / 3 + 0.2))
feng_liu_ma_white_noise_test(x, lag = 2)
```

`feng_liu_rank_sphericity_test`

Feng–Liu spatial-rank sphericity tests

Description

Implements the Spearman- and Kendall-type high-dimensional sphericity tests of Feng and Liu (2017). `method = "spearman"` uses

$$\widehat{\mathrm{tr}(\Omega^2)} = \{2n(n-1)(n-2)(n-3)\}^{-1} \sum\nolimits^{*} (U_{ij}^{\mathsf{T}} U_{kl})(U_{kj}^{\mathsf{T}} U_{il})$$

and $\widetilde{Q} = 4p\widehat{\mathrm{tr}(\Omega^2)} - 1$. `method = "kendall"` removes the factor two, replaces the summand by $(U_{ij}^{\mathsf{T}} U_{kl})^2$, and uses $\widetilde{Q} = p\widehat{\mathrm{tr}(\Omega^2)} - 1$. The star means that all four ordered indices are distinct. Both divide by the same σ_0 used by the spatial-sign test.

Usage

```
feng_liu_rank_sphericity_test(
  x,
  method = c("spearman", "kendall"),
  alpha = 0.05,
  keep_pair_signs = FALSE
)
```

Arguments

`x`	A finite numeric matrix or data frame with observations in rows; at least two rows and two columns are required.
`method`	Either `"spearman"` or `"kendall"`.
`alpha`	A finite test level strictly between zero and one.
`keep_pair_signs`	Whether to retain the $\binom{n}{2} \times p$ pair-sign matrix and its endpoint table.

Details

Pair directions are computed once in C++. For each unordered four-set, its 24 ordered terms are reduced exactly to the three disjoint pairings. A tied pair has the literal direction `U(0) = 0`, is counted in diagnostics, and is never dropped from a denominator.

Value

An object of class `c("hd_sphericity_test", "htest")` containing the exact ordered sum, denominator, trace estimate, pair-tie diagnostics, and optional pair signs.

References

Feng, L. and Liu, B. (2017). High-dimensional rank tests for sphericity. *Journal of Multivariate Analysis*, 155, 217–233. doi:10.1016/j.jmva.2017.01.005.

Examples

```
x <- rbind(c(-2, 0), c(-1, 1), c(0, -2), c(1, 2), c(3, -1), c(2, 1))
feng_liu_rank_sphericity_test(x, method = "kendall")
```

`feng_spatial_rank_proportionality_test`
Feng–Zhang–Liu spatial-rank proportionality test

Description

Implements the high-dimensional spatial-rank test of Feng, Zhang, and Liu (2022). For each group, the within component averages $\{U(X_i - X_j)^T U(X_k - X_l)\}^2$ over four *ordered, mutually distinct* indices. The cross component averages over an ordered unequal pair in each group. If the corresponding unscaled averages are A_1, A_2, C_{12}, then

$$T_{HT} = p(A_1 + A_2 - 2C_{12}).$$

Usage

```
feng_spatial_rank_proportionality_test(
  x,
  y,
  alpha = 0.05,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

x, y	Numeric matrices or data frames with at least four observations per group and the same variables in columns.
alpha	Nominal level for the paper's upper-tail rejection rule.
zero_tol	Non-negative tolerance below which a pairwise difference is assigned the exact zero spatial sign.
strict	If TRUE, fail when the feasible variance is invalid; otherwise return raw components with no calibrated statistic or p-value.

Details

The feasible null variance in the primary paper is implemented literally:

$$\widehat{\sigma}^2_{0,n} = \frac{4(n_1^{-1} + n_2^{-1})^2}{(n_1 + n_2)(p+2)^2} p^2 (n_1 A_1 + n_2 A_2).$$

The statistic is calibrated by the upper normal tail. Pairwise ties use the paper's convention $U(0) = 0$. A non-positive feasible variance is not floored or replaced.

The primary paper normalizes its shape matrix Λ to trace one, and the estimand is $p\,tr\{(S_1 - S_2)^2\}$ for the SSCM/Kendall functionals. This is not the trace-p shape-scale formula printed in the current book draft; multiplying Λ by p requires the corresponding powers of p in every variance trace.

Value

An hd_proportionality_test object. Both raw U-statistic averages and their p-scaled contributions are returned.

References

Feng, L., Zhang, X., and Liu, B. (2022). High-dimensional proportionality test of two covariance matrices and its application to gene expression data. *Statistical Theory and Related Fields*, **6**, 161–174. doi:10.1080/24754269.2021.1984373.

Examples

```
x <- matrix(seq_len(40), 8, 5) + matrix(c(-1, 0, 1, 0), 8, 5)
y <- matrix(seq_len(45), 9, 5) + matrix(c(0, 1, -1), 9, 5)
feng_spatial_rank_proportionality_test(x, y)
```

`feng_sun_one_sample_test` *Feng–Sun scalar-invariant one-sample spatial-sign test*

Description

Tests $H_0 : \theta = \mu_0$ against an unrestricted location alternative with the scalar-invariant high-dimensional spatial-sign procedure of Feng and Sun (2016). For every unordered pair $i < j$, the function jointly fits a leave-two-out location $\widehat{\theta}_{ij}$ and positive diagonal scale $\widehat{D}_{ij}$ from all observations except i, j. The raw statistic is

$$T_{SS} = \frac{2}{n(n-1)} \sum_{i<j} U\{\widehat{D}_{ij}^{-1/2}(X_i - \mu_0)\}^{\mathsf{T}} U\{\widehat{D}_{ij}^{-1/2}(X_j - \mu_0)\}.$$

In particular, $\widehat{\theta}_{ij}$ does **not** enter this numerator.

Usage

```
feng_sun_one_sample_test(x, mu = NULL, tol = 1e-07, max_iter = 500L)
```

Arguments

`x`	A numeric matrix or data frame with observations in rows and variables in columns. At least four observations are required, and every leave-two-out fit must have positive variation in every variable.
`mu`	A finite numeric vector giving the null location. The default is the zero vector.
`tol`	A finite positive tolerance for both leave-out estimating equations. The default is `1e-7`.
`max_iter`	A positive integer giving the maximum number of diagonal HR updates for each pair. Non-convergence is an error.

Details

The implementation also supplies the feasible null calibration from the original paper, which is needed for an operational test but is omitted from the short presentation in the book chapter:

$$\widehat{\mathrm{tr}(R^2)} = \frac{p^2}{n(n-1)} \sum_{i \neq j} \left[U\{\widehat{D}_{ij}^{-1/2}(X_i - \widehat{\theta}_{ij})\}^{\mathsf{T}} U\{\widehat{D}_{ij}^{-1/2}(X_j - \widehat{\theta}_{ij})\} \right]^2,$$

$$\widehat{\sigma}_n^2 = \frac{2\widehat{\mathrm{tr}(R^2)}}{n(n-1)p^2}, \qquad Z_{SS} = T_{SS}/\widehat{\sigma}_n.$$

The p-value is the upper tail of the asymptotic standard normal law. The vector alternative is conventionally labelled `two.sided`, while large positive quadratic evidence is the rejection direction.

Each leave-out fit follows the paper's diagonal HR recursion. Its initial location and scale are the leave-out sample mean and marginal sample variances. Iteration stops only when both the infinity

norm of the mean sign equation and the maximum diagonal-scale equation residual are at most `tol`. Failure to converge, a coincident training observation (for which the inverse-radius location update is undefined), a non-positive marginal scale, or a non-positive feasible variance raises an error. No ridge, absolute-value repair, or numerical floor is used. The paper itself notes that general existence, uniqueness, and convergence of this diagonal HR recursion are not established.

Before fitting, null-centred data are divided by a safe scale separately in every variable. This is an algebraically neutral coordinatewise change of units for the Feng–Sun statistic and protects both very large and very small finite inputs. Returned `leaveout.scale.diagonal` values are mapped back to input-coordinate geometry and canonically normalised so their largest diagonal entry is one; the scale equations identify the diagonal only up to a common multiplier. The corresponding canonical log diagonals are also returned, retaining information if a display-scale entry underflows. `leaveout.location` is in the original input units.

The literal algorithm requires $n(n-1)/2$ separate iterative fits and is therefore computationally intensive. Its compiled implementation is intended for faithful, moderate-sample use rather than silently replacing the published statistic by a full-sample approximation.

Value

An object of class `c("hd_location_test", "htest")`. Components include `T.SS`, the ordered-pair trace estimate, `sigma2.hat`, pairwise numerator and variance contributions, and every leave-two-out location and canonical diagonal-scale fit. Diagnostics report equation residuals, iteration counts, exact zero signs, internal scaling, and the no-repair policy.

References

Feng, L. and Sun, F. (2016). Spatial-sign based high-dimensional location test. *Electronic Journal of Statistics*, **10**, 2420–2434. doi:10.1214/16EJS1176.

Examples

```
set.seed(2608)
x <- matrix(stats::rt(48, df = 5), 8, 6)
feng_sun_one_sample_test(x, tol = 1e-6)
```

`feng_wang_pdq_two_sample_test`

Feng–Wang pairwise-difference-quantile spatial-sign test

Description

Tests equality of two high-dimensional elliptical location vectors using the pairwise-difference-quantile (PDQ) spatial-sign method of Feng and Wang (2026). Observations are rows and variables are columns. The method allows arbitrary within-group correlation and unequal scatter matrices.

Usage

```
feng_wang_pdq_two_sample_test(
  x,
  y,
  quantile_prob = 0.25,
  level = 0.05,
  B = 999L,
  seed = NULL,
  keep_bootstrap = FALSE,
  tol = 1e-08,
  max_iter = 1000L,
  strict = TRUE
)
```

Arguments

x, y	Numeric matrices or data frames containing the two independent samples. They must have the same variables, at least three rows apiece, and at least two columns.
quantile_prob	Probability for the coordinatewise PDQ U-quantile. The default is 0.25; the paper also mentions 0.5.
level	Test level strictly between zero and one. This is the paper's rejection probability (denoted beta there), not quantile_prob.
B	Positive number of Rademacher bootstrap draws.
seed	NULL, or an integer-valued counter-generator seed in [0, 2^32 - 1]. NULL draws and records one seed from R's RNG. An explicit seed makes the bootstrap reproducible without changing R's RNG state.
keep_bootstrap	If TRUE, retain all bootstrap statistics and the corresponding Rademacher multiplier matrix.
tol	Finite positive spatial-median subgradient tolerance.
max_iter	Positive maximum number of modified Weiszfeld updates.
strict	Whether spatial-median non-convergence is an error (TRUE) or a warning followed by use of the last defined iterate (FALSE).

Details

For group k and coordinate j, let $M_k = n_k(n_k-1)/2$ and form the M_k unordered absolute pairwise differences. The PDQ scale is the exact empirical U-quantile

$$q_{kj} = \inf\{t : M_k^{-1} \sum_{i<l} 1(|X_{kij} - X_{klj}| \le t) \ge a\},$$

where a is quantile_prob. Thus the implementation selects sorted difference number $\lceil aM_k \rceil$; it does not use an interpolated sample quantile. The diagonal standardizer has entries q_{kj}^2.

After computing each full-sample spatial median under its PDQ standardizer, the observed full-sample statistic is the cross-centred quantity

$$\widehat{R} = -\frac{1}{n_1 n_2} \sum_{i,j} U\{\widehat{D}_1^{-1/2}(X_{1i} - \widehat{\theta}_2)\}^T U\{\widehat{D}_2^{-1/2}(X_{2j} - \widehat{\theta}_1)\}.$$

This is not replaced by the fitted-sign quadratic expansion. From fitted within-group signs the function constructs the published matrices $\widehat{K}_1, \widehat{K}_2, \widehat{K}_3$ and subtracts the empirical diagonal term

$$\widehat{b} = \sum_{k=1}^{2} n_k^{-2} \sum_i \widehat{S}_{ki}^T \widehat{K}_k \widehat{S}_{ki}.$$

The reported statistic is $T_{PDQ} = \widehat{R} - \widehat{b}$; large values reject the two-sided location null.

Calibration fixes all nuisance estimates. For each bootstrap draw, independent Rademacher multipliers are applied to the fitted within-group signs, the published quadratic form Q^* is evaluated, and $T^* = Q^* - \widehat{b}$ is recorded. The paper's primary rule rejects strictly when T_{PDQ} exceeds the lower empirical $(1 - \text{level})$ quantile of the bootstrap values. Because the paper does not prescribe a finite-B p-value, the `htest` `p.value` uses the explicitly documented conservative convention

$$p_{MC} = \frac{1 + \#\{T^* \geq T_{PDQ}\}}{B + 1}.$$

Ties enter the upper tail. The primary strict-quantile decision and the auxiliary p-value decision are both returned and can differ on the finite bootstrap grid.

The spatial median is computed with a modified Weiszfeld recursion and is accepted when its normalized subgradient residual is at most `tol`. `strict = TRUE` makes non-convergence an error; `strict = FALSE` warns and continues from the last iterate when every published downstream quantity remains defined. A fitted zero residual is always an error because the definition of $\widehat{G}_k$ requires its inverse radius. Zero cross-centred residuals use the paper's convention $U(0) = 0$ and are counted. Zero PDQ scales, singular $\widehat{G}_k$, and non-positive diagonal-deleted bootstrap variance fail explicitly. No ridge, scale floor, generalized inverse, absolute-value correction, or random perturbation is used.

A common coordinatewise affine preconditioning is used internally to protect calculations with extreme but finite units. It is algebraically neutral under the method's common translation and common nonzero coordinatewise scaling invariances. Actual input-unit PDQ diagonals are returned when representable; logarithms and a canonical maximum-one version retain scale ratios when squaring would overflow or underflow.

Value

An object of class `c("hd_location_test", "htest")`. Its `components` retain the PDQ quantiles and exact order-statistic ranks, both spatial-median fits, fitted and cross-centred signs, the $\widehat{\Omega}_k$, $\widehat{G}_k$, bridge and K matrices, observed and fitted quadratic decompositions, diagonal-deletion kernels, bootstrap variance, tail count, and optional draws. `diagnostics` records both finite-B decisions, convergence, zeros, conditioning, numerical representation, seed, and the no-repair policy.

References

Feng, L. and Wang, H. (2026). High-dimensional two-sample test for elliptical symmetry distribution. arXiv:2605.03265. doi:10.48550/arXiv.2605.03265.

Examples

```
v <- rbind(c(1, 2, 3), c(2, -3, 1), c(-4, 1, 2), c(3, 4, -2))
x <- rbind(v, -v)
```

```
y <- rbind(1.1 * v, -1.1 * v) + rep(c(0.2, -0.1, 0.15), each = 8)
feng_wang_pdq_two_sample_test(x, y, B = 19, seed = 2605)
```

`feng_zhang_liu_spatial_rank_test` *Feng–Zhang–Liu high-dimensional two-sample spatial-rank test*

Description

Tests equality of two high-dimensional elliptical location vectors using the leave-two-out spatial-rank statistic of Feng, Zhang, and Liu (2020). Observations are rows and variables are columns. For distinct i, j in group 1 and s, t in group 2, let $\widehat{D}_{(i,j,s,t)}$ be the weighted pool of the two diagonal spatial-rank scale fits after deleting i, j and s, t, respectively. The raw statistic is

$$T_n = \{n_1(n_1-1)n_2(n_2-1)\}^{-1} \sum_{i \neq j} \sum_{s \neq t} U\{\widehat{D}^{-1/2}_{(i,j,s,t)}(X_{1i} - X_{2s})\}^{\mathsf{T}} U\{\widehat{D}^{-1/2}_{(i,j,s,t)}(X_{1j} - X_{2t})\}.$$

Here $U(v) = v/\|v\|$ for nonzero v, and $U(0) = 0$.

Usage

```
feng_zhang_liu_spatial_rank_test(
  x,
  y,
  tol = 1e-07,
  max_iter = 500L,
  scale_identification = c("geometric", "paper_trace")
)
```

Arguments

`x, y`	Numeric matrices or data frames with observations in rows and the same variables in columns. Each sample must have at least six rows.
`tol`	Finite positive convergence tolerance for every full-sample, leave-two-out, and leave-four-out diagonal spatial-rank scale fit.
`max_iter`	Positive integer maximum number of fixed-point updates for every scale fit. Nonconvergence is an error.
`scale_identification`	Scale representative used before pooling group fits. The default `"geometric"` gives exact coordinatewise-scale invariance. `"paper_trace"` reproduces the article's literal trace-normalized recursion for formula auditing.

Details

The p-value uses the feasible null calibration in the original article. Its two within-group trace estimates use all ordered quadruples of mutually distinct observations and their leave-four-out diagonal scale fits. The cross trace uses all ordered within-group pairs, pooled leave-two-out scales, and the published denominator $n_1^2 n_2^2$. If these estimates are denoted by $\widehat{\mathrm{tr}(R_1^2)}$, $\widehat{\mathrm{tr}(R_2^2)}$, and $\widehat{\mathrm{tr}(R_1 R_2)}$, then

$$\widehat{\sigma}_n^2 = \frac{\widehat{\mathrm{tr}(R_1^2)}}{2n_1(n_1-1)p^2} + \frac{\widehat{\mathrm{tr}(R_2^2)}}{2n_2(n_2-1)p^2} + \frac{\widehat{\mathrm{tr}(R_1 R_2)}}{n_1 n_2 p^2}.$$

The p^2 in the second term restores an evident typographical omission in the author TeX: it is required by the symmetric oracle variance and by consistency of the feasible estimator. The reported statistic is $Z = T_n/\widehat{\sigma}_n$; large positive values reject, so the p-value is the upper standard-normal tail. The local-alternative oracle variance is not used.

Each diagonal fit follows the published spatial-rank fixed-point update. `scale_identification = "paper_trace"` applies the article's literal $\mathrm{tr}(D) = p$ normalization. Because separately fitted trace-normalized matrices acquire different scalar representatives after coordinatewise rescaling, their weighted pool is not exactly coordinatewise-scale equivariant in finite samples. The default `"geometric"` instead imposes unit geometric mean. All leave-out fits then acquire the same harmless scalar under a common coordinatewise rescaling, making the complete statistic exactly coordinatewise-scale invariant while leaving the published fixed-point equation unchanged. Both choices and their distinction are recorded in `diagnostics`.

At least six observations per group are necessary: a leave-four-out fit must retain at least two observations. More observations may be needed for a particular data set if a retained subset has zero marginal variance or zero spatial-rank energy. Such degeneracy, nonconvergence, and a nonpositive feasible variance are errors. No ridge, absolute value, numerical floor, or random perturbation is applied. Pairwise ties follow $U(0) = 0$ and are counted.

The formal asymptotic calibration in the paper assumes a common scatter matrix and its stated high-dimensional trace conditions. The article's numerical study considered unequal scatter, but it did not establish the null limit in that setting; this function therefore does not claim an unequal-scatter guarantee and does not switch to a simulation or bootstrap calibration.

Value

An object of class `c("hd_location_test", "htest")`. `components` contains the raw statistic, all three feasible trace and variance terms, exact ordered-sum denominators, block/permutation contributions, and every full/leave-two/leave-four scale fit. `diagnostics` records convergence, zero directions, preprocessing, identification, the corrected author-TeX factor, applicability, and the no-repair policy.

References

Feng, L., Zhang, X., and Liu, B. (2020). A high-dimensional spatial rank test for the two-sample location problem. *Computational Statistics & Data Analysis*, **144**, 106889. doi:10.1016/j.csda.2019.106889.

Examples

```
x <- matrix(c(
  0.7, -1.1, 0.2, -0.4, 0.8, 1.2, 1.1, 0.3, -0.9,
```

```
  -1.2, -0.5, 0.7, 0.2, 1.4, -0.4, 1.5, -0.2, 0.5
), ncol = 3, byrow = TRUE)
y <- matrix(c(
  -0.3, 0.7, -0.1, 1.2, -0.6, 0.8, -1.1, 0.2, 1.3,
  0.5, 1.1, -0.7, 0.9, -1.3, 0.4, -0.6, -0.4, -1.2
), ncol = 3, byrow = TRUE)
feng_zhang_liu_spatial_rank_test(x, y, tol = 1e-6)
```

`feng_zou_wang_two_sample_sign_test`
Feng–Zou–Wang two-sample multivariate-sign test

Description

Tests equality of two multivariate elliptical location vectors with the scalar-invariant spatial-sign procedure of Feng, Zou, and Wang (2016). Observations are rows and variables are columns. If $(\widehat{\theta}_{k,i}, \widehat{D}_{k,i})$ is the diagonal HR fit in group k after deleting observation i, the raw statistic is

$$R_n = -\frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} U\{\widehat{D}_{1,i}^{-1/2}(X_{1i} - \widehat{\theta}_{2,j})\}^{\mathsf{T}} U\{\widehat{D}_{2,j}^{-1/2}(X_{2j} - \widehat{\theta}_{1,i})\}.$$

Thus, each sign uses its own group's leave-one-out scale but the other group's leave-one-out location. This crossed construction removes the high-dimensional location-estimation bias without estimating and subtracting a separate bias term.

Usage

```
feng_zou_wang_two_sample_sign_test(x, y, tol = 1e-07, max_iter = 500L)
```

Arguments

`x, y`	Numeric matrices or data frames with observations in rows and the same variables in columns. Each group needs at least three rows, and each full/leave-one-out fit must have positive marginal variation.
`tol`	Finite positive tolerance for both diagonal HR estimating equations. The default is `1e-7`.
`max_iter`	Positive integer maximum number of HR updates for every full-sample and leave-one-out fit. Nonconvergence is an error.

Details

The p-value uses the feasible null calibration in Proposition 2 of the original article, not the oracle variance for local alternatives. Let $\widetilde{U}_{ki} = U\{\widehat{D}_{k,i}^{-1/2}(X_{ki} - \widehat{\theta}_{k,i})\}$ and $\widehat{c}_k = n_k^{-1} \sum_i \|\widehat{D}_{k,i}^{-1/2}(X_{ki} - \widehat{\theta}_{k,i})\|^{-1}$. With $\widehat{D}_1, \widehat{D}_2$ denoting the two full-sample diagonal HR fits, the three trace estimators are

$$\widehat{\mathrm{tr}(A_1^2)} = \frac{p^2 \widehat{c}_2^2 \widehat{c}_1^{-2}}{n_1(n_1-1)} \sum_{k=1}^{n_1} \sum_{\ell \neq k} (\widetilde{U}_{1\ell}^{\mathsf{T}} \widehat{D}_2^{-1/2} \widehat{D}_1^{1/2} \widetilde{U}_{1k})^2,$$

with the group indices reversed for $\widehat{\mathrm{tr}(A_2^2)}$, and

$$\widehat{\mathrm{tr}(A_3^{\mathsf{T}}A_3)} = \frac{p^2}{n_1 n_2}\sum_{\ell=1}^{n_1}\sum_{k=1}^{n_2}(\widetilde{U}_{1\ell}^{\mathsf{T}}\widetilde{U}_{2k})^2.$$

These are combined as

$$\widehat{\sigma}_n^2 = \frac{2\widehat{\mathrm{tr}(A_1^2)}}{n_1(n_1-1)p^2} + \frac{2\widehat{\mathrm{tr}(A_2^2)}}{n_2(n_2-1)p^2} + \frac{4\widehat{\mathrm{tr}(A_3^{\mathsf{T}}A_3)}}{n_1 n_2 p^2}.$$

The reported statistic is $Z = R_n/\widehat{\sigma}_n$; large positive values reject, so the p-value is the upper standard-normal tail.

Every full-sample and leave-one-out fit follows the published diagonal HR recursion, initialized by the applicable sample mean and marginal sample variances. Convergence requires both estimating-equation residuals to be at most `tol`. A coincident training observation makes the inverse-radius update undefined and is an error. The spatial-sign convention is $U(0) = 0$ in the numerator, but a zero own-sample leave-one-out radius is also an error because its reciprocal is required by $\widehat{c}_k$. Nonconvergence, non-positive scales, and non-positive feasible variance are reported without ridge, absolute-value, floor, or perturbation repairs.

Internally, both groups receive one common, safe coordinatewise affine transformation. This is algebraically neutral under the method's shift and coordinatewise nonzero-scaling invariance and protects extreme finite units. Returned input-coordinate diagonal scales are canonically normalized to have largest diagonal entry one; log diagonals retain ratios that underflow in the display-scale version.

The paper recommends a bootstrap calibration when the dimension is small (it gives $p \leq 50$ as a guide) or grows at order n^2 or faster. This function implements the article's feasible asymptotic-normal test; diagnostics flag those finite-sample regimes. It does not silently switch calibration or reproduce the paper's simulation procedure.

Value

An object of class `c("hd_location_test", "htest")`. In addition to the test result, `components` contains both full-sample fits, all leave-one-out fits and directions, inverse radii, bridge diagonals, ordered/cross pair contributions, three trace estimates, and all three variance contributions. `diagnostics` contains iteration counts, equation residuals, minimum training radii, zero-sign counts, numerical scaling information, asymptotic-regime flags, and the no-repair policy.

References

Feng, L., Zou, C., and Wang, Z. (2016). Multivariate-sign-based high-dimensional tests for the two-sample location problem. *Journal of the American Statistical Association*, **111**(514), 721–735. doi:10.1080/01621459.2015.1035380.

Examples

```
set.seed(2016)
x <- matrix(stats::rt(48, df = 5), 8, 6)
y <- matrix(stats::rt(54, df = 5), 9, 6)
feng_zou_wang_two_sample_sign_test(x, y, tol = 1e-6)
```

`feng_zou_wang_zhu_two_sample_test`
Feng–Zou–Wang–Zhu scale-invariant Behrens–Fisher test

Description

Tests equality of two high-dimensional mean vectors without requiring equal covariance matrices. Observations are rows and variables are columns. Let $\gamma = n_1/n_2$, let $\widehat{\sigma}^2_{sk}$ be the unbiased sample variance of variable k in group s, and define

$$A_k = (\bar{X}_{1k} - \bar{X}_{2k})^2 - \widehat{\sigma}^2_{1k}/n_1 - \widehat{\sigma}^2_{2k}/n_2.$$

The initial statistic is

$$T_{BF} = \sum_k \frac{A_k}{\widehat{\sigma}^2_{1k} + \gamma\widehat{\sigma}^2_{2k}}.$$

Usage

```
feng_zou_wang_zhu_two_sample_test(x, y)
```

Arguments

`x`, `y`	Numeric matrices or data frames with observations in rows and the same variables in columns. Each group must contain at least six observations.

Details

The reported statistic subtracts the Feng–Zou–Wang–Zhu plug-in estimate of the asymptotic null centering and divides by their ratio-consistent null standard deviation. If $\widehat{\kappa}_{sk} = n_s^{-1}\sum_i (X_{sik} - \bar{X}_{sk})^3$ and $D_k = \widehat{\sigma}^2_{1k} + \gamma\widehat{\sigma}^2_{2k}$, the centering is

$$\widehat{\mu}_n = \widehat{b}_1 + \widehat{b}_2,$$

with

$$\widehat{b}_1 = \sum_k \left\{ \frac{2\widehat{\sigma}^4_{1k}}{n_1(n_1-1)D_k^2} + \frac{2\gamma\widehat{\sigma}^4_{2k}}{n_2(n_2-1)D_k^2} \right\},$$

and

$$\widehat{b}_2 = \sum_k \frac{2}{D_k^3}\left(\frac{\widehat{\kappa}_{1k}}{n_1} - \frac{\gamma\widehat{\kappa}_{2k}}{n_2}\right)^2.$$

In particular, the second term of $\widehat{b}_1$ contains one power of γ. This follows the published paper; the current book draft at `ch2_location.tex:685` has an extra power of γ. The displayed population expansion in that draft is also an asymptotic centering up to a smaller-order remainder, not an exact finite-sample expectation.

The two within-group covariance trace estimates use exactly the paper's leave-four-out marginal variances and denominator $2P^4_{n_s}$. The cross trace removes two observations from each group and uses denominator $4P^2_{n_1}P^2_{n_2}$. At least six observations are consequently required in each group. The

C++ kernel sums the 24 orderings associated with each unordered within-group quadruple from a local 4×4 dual Gram matrix. This reduces computation without changing the published estimator and never constructs a $p \times p$ matrix.

Both samples are internally translated by a common anchor and divided by a common scale within each variable. This is algebraically neutral and protects the statistic under very large or small measurement units. Every full-sample and leave-out combined marginal variance, and the final null variance estimate, must have the sign required by the original formulas. No ridge, absolute-value repair, or variance floor is applied.

Value

An object of class `c("hd_location_test", "htest")`. The upper-tail normal p-value is based on $Z = (T_{BF} - \widehat{\mu}_n)/\widehat{\sigma}_n$. `components` contains `T.BF`, `Q3 = n1 * T.BF`, the two centering terms, all three published leave-out trace estimates, their sample-size coefficients and permutation counts, and coordinate-level quantities. `diagnostics` records the exact leave-out policy and the absence of numerical repair.

References

Feng, L., Zou, C., Wang, Z., and Zhu, L. (2015). Two-sample Behrens–Fisher problem for high-dimensional data. *Statistica Sinica*, **25**, 1297–1312. doi:10.5705/ss.2014.048.

Examples

```
set.seed(2411)
x <- matrix(rnorm(56), 8, 7)
y <- matrix(rnorm(63, 0.15), 9, 7)
feng_zou_wang_zhu_two_sample_test(x, y)
```

`fisher_sun_gallagher_sphericity_test`

Fisher–Sun–Gallagher fourth-to-second trace sphericity test

Description

For $N = n + 1$ observations, this implements the primary paper's unbiased estimators $\hat{a}_2$ and $\hat{a}_4$ of $p^{-1}\,\mathrm{tr}(\Sigma^2)$ and $p^{-1}\,\mathrm{tr}(\Sigma^4)$. With $c = p/n$, the statistic

$$\sqrt{\frac{np}{8(8 + 12c + c^2)}}\left(\frac{\hat{a}_4}{\hat{a}_2^2} - 1\right)$$

has an upper standard-normal null calibration. The calculation is performed after a common scale normalization, which is algebraically neutral.

Usage

```
fisher_sun_gallagher_sphericity_test(x)
```

Arguments

x	Numeric matrix with observations in rows; at least five rows.

Value

An hd_covariance_test object retaining every trace and coefficient.

References

Fisher, T. J., Sun, X. and Gallagher, C. M. (2010). *Journal of Multivariate Analysis*, 101, 2554–2570. doi:10.1016/j.jmva.2010.07.004.

Examples

```
set.seed(43)
x <- matrix(rnorm(60), nrow = 15, ncol = 4)
fisher_sun_gallagher_sphericity_test(x)
```

---

gaussian_alpha_combination_test

Feng–Lan–Liu–Ma adaptive Gaussian alpha test

Description

Combines the Pesaran–Yamagata sum p-value and the Feng–Lan–Liu–Ma max p-value with the primary paper's two-test Bonferroni rule

$$p_{\mathrm{COM}} = \min\{1, 2\min(p_{\mathrm{PY}}, p_{\mathrm{MAX}})\}.$$

It deliberately does not implement the generic Cauchy benchmark introduced in the book draft, which is not the combination proposed in the cited Feng–Lan–Liu–Ma method.

Usage

```
gaussian_alpha_combination_test(returns, factors = NULL, p0 = 0.1, delta = 1)
```

Arguments

returns	Finite observation-by-asset numeric matrix or data frame.
factors	NULL, a finite length-T numeric vector, or a finite observation-by-factor numeric matrix or data frame.
p0	Finite thresholding probability in (0,1), default 0.1.
delta	Finite positive threshold exponent, default 1.

Value

An alpha-test object containing both complete component tests.

References

Feng, L., Lan, W., Liu, B., and Ma, Y. (2022). High-dimensional test for alpha in linear factor pricing models. Journal of Econometrics. doi:10.1016/j.jeconom.2021.07.011.

Examples

```
f <- cbind(seq(-1, 1, length.out = 12))
y <- outer(seq_len(12), 1:3, function(i, j) sin(i * j))
gaussian_alpha_combination_test(y, f)
```

`gaussian_covariance_lrt`

Classical Gaussian likelihood-ratio test for a covariance matrix

---

Description

Tests $H_0 : \Sigma = \Sigma_0$ with the Gaussian likelihood-ratio statistic

$$n\{\mathrm{tr}(\Sigma_0^{-1} S_n) - \log|\Sigma_0^{-1} S_n| - p\},$$

where $S_n = n^{-1}\sum_i (X_i - \bar{X})(X_i - \bar{X})'$ when `center = TRUE`. Its fixed-dimensional reference distribution is chi-squared with $p(p+1)/2$ degrees of freedom. The sample covariance must be positive definite; no pseudo-determinant or ridge is substituted.

Usage

```
gaussian_covariance_lrt(x, sigma0, center = TRUE)
```

Arguments

`x`	Numeric matrix with observations in rows.
`sigma0`	Finite symmetric positive-definite null covariance matrix.
`center`	Whether to estimate and remove the mean. `FALSE` implements the known-zero-mean model.

Value

An object of class `c("hd_covariance_test", "htest")`.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Wiley.

Examples

```
x <- matrix(rnorm(80), 20, 4)
gaussian_covariance_lrt(x, diag(4))
```

`gaussian_graphical_lasso` *Gaussian graphical lasso with an off-diagonal penalty*

Description

With the centered empirical covariance S_n, solves the Yuan–Lin off-diagonal graphical-lasso program

$$\min_{\Omega \succ 0} \operatorname{tr}(S_n \Omega) - \log\det(\Omega) + \lambda \sum_{i \neq j} |\omega_{ij}|.$$

Diagonal entries are never penalized. The optimizer uses an SPD-preserving proximal-gradient step with explicit majorization backtracking and returns the full diagonal/off-diagonal subgradient KKT residual. With `lambda = 0`, the exact inverse is returned only when S_n is strictly positive definite; no pseudoinverse or ridge is substituted.

Usage

```
gaussian_graphical_lasso(
  x,
  lambda,
  center = TRUE,
  divisor = c("n", "n-1"),
  solver_tol = 1e-07,
  solver_max_iter = 100000L,
  initial_step = 1,
  max_backtracking = 100L,
  strict = TRUE
)
```

Arguments

`x`	Numeric observation-by-variable matrix or data frame.
`lambda`	Finite non-negative off-diagonal lasso penalty on the correlation scale.
`center`	Whether to subtract column means.
`divisor`	Either `"n"` (formal default) or `"n-1"`.
`solver_tol`	Positive tolerance for every feasibility and KKT certificate.
`solver_max_iter`	Positive ISP/ADMM iteration limit.
`initial_step`	Positive initial proximal-gradient step.
`max_backtracking`	Positive line-search reduction limit per iteration.
`strict`	If `TRUE`, a failed solver certificate is an error. If `FALSE`, the function warns and returns `estimate = NULL` with `valid = FALSE` and the uncertified last iterate in diagnostics.

Value

A gaussian_precision_fit list. estimate is non-NULL only when SPD, objective-descent, relative-update, and KKT certificates all pass.

References

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94, 19–35. doi:10.1093/biomet/asm018.

Examples

```
x <- rbind(c(-2, 0), c(-1, -1), c(1, 1), c(2, 0))
gaussian_graphical_lasso(x, lambda = 0.2)
```

---

gaussian_lda_oracle *Oracle Gaussian linear discriminant classifier*

Description

Constructs the exact Gaussian log-likelihood-ratio score

$$(z - (\mu_1 + \mu_2)/2)'\Omega(\mu_1 - \mu_2) + \log(\pi_1/\pi_2).$$

Exactly one of covariance and precision must be supplied. No ridge, generalized inverse, or positive-definite repair is used.

Usage

```
gaussian_lda_oracle(
  location1,
  location2,
  covariance = NULL,
  precision = NULL,
  prior = "equal",
  levels = c("class1", "class2"),
  feature_names = NULL
)
```

Arguments

location1, location2	Finite class-location vectors.
covariance	Optional common positive-definite covariance matrix.
precision	Optional common positive-definite precision matrix.
prior	"equal" or a positive numeric vector of length two.
levels	Two class labels.
feature_names	Optional unique feature names.

Value

A valid `hd_classifier_fit`.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Wiley.

Examples

```
fit <- gaussian_lda_oracle(c(1, 0), c(-1, 0), covariance = diag(2))
predict(fit, rbind(c(2, 0), c(-2, 0)))
```

`gaussian_mixture_em` *Full-covariance Gaussian-mixture EM*

Description

Implement the Chapter 7 component-specific full-covariance Gaussian EM updates. Means, covariance arrays, and mixing proportions are all required: this function has no hidden start. The E-step uses Cholesky solves and log-sum-exp; no inverse or pseudo-inverse is formed. The M-step covariance has maximum-likelihood divisor `sum(tau[, k])`.

Usage

```
gaussian_mixture_em(
  x,
  means,
  covariances,
  proportions,
  ridge = 0,
  solver_tol = 1e-08,
  monotone_tol = 1e-10,
  rank_tol = 1e-10,
  symmetry_tol = 1e-12,
  solver_max_iter = 200L,
  strict = TRUE,
  keep_responsibilities = TRUE
)
```

Arguments

`x`	Numeric observation matrix.
`means`	Required `K` by `p` initial mean matrix.
`covariances`	Required `p` by `p` by `K` initial covariance array (a matrix is accepted for `K = 1`).
`proportions`	Required strictly positive length-`K` vector summing exactly to one.
`ridge`	Explicit non-negative covariance ridge; zero means none.

`solver_tol` Relative parameter and log-likelihood fixed-point tolerance.

`monotone_tol` Numerical monotonicity tolerance for ordinary EM.

`rank_tol` Relative eigenvalue cutoff used only for raw-rank diagnostics, never for inversion or flooring.

`symmetry_tol` Explicit tolerance for arithmetic symmetrization of supplied covariance matrices.

`solver_max_iter` Maximum EM updates.

`strict` Error rather than return an uncertified iterate.

`keep_responsibilities` Retain the final posterior matrix.

Details

The strict default `ridge = 0` implements ordinary EM. A positive explicit ridge adds `ridge * I` after every raw covariance M-step and is labelled regularized EM; unpenalized likelihood monotonicity is then not claimed. Singular initial or updated covariance matrices error instead of receiving an eigenvalue floor. Gaussian-mixture likelihood is unbounded without further restrictions, so a numerical EM fixed point is not a global-MLE certificate.

Value

A `gaussian_mixture_em_fit` object containing fitted parameters, hard labels (posterior ties use the smallest component), likelihood trace, effective masses, raw covariance ranks, and solver certificates.

References

Feng, L. (2026). *High-Dimensional Data Analysis for Elliptical Symmetric Distributions*, Chapter 7 (book manuscript). The implementation is the explicitly documented standard Gaussian-mixture E/M construction.

Examples

```
x <- matrix(c(-2.2, -2, -1.8, 1.8, 2, 2.2), ncol = 1)
gaussian_mixture_em(
  x, means = matrix(c(-2, 2), ncol = 1),
  covariances = array(c(0.2, 0.2), c(1, 1, 2)),
  proportions = c(0.5, 0.5), solver_tol = 1e-6
)
```

`gaussian_qda_oracle` *Oracle Gaussian quadratic discriminant classifier*

Description

Uses the canonical Gaussian log-likelihood ratio

$$-\tfrac{1}{2}\log|\Sigma_1| + \tfrac{1}{2}\log|\Sigma_2| - \tfrac{1}{2}d_1^2 + \tfrac{1}{2}d_2^2 + \log(\pi_1/\pi_2).$$

The direct quadratic formula displayed later in the book is twice this score. It has the same boundary but is not the same numerical score.

Usage

```
gaussian_qda_oracle(
  location1,
  location2,
  covariance1 = NULL,
  covariance2 = NULL,
  precision1 = NULL,
  precision2 = NULL,
  prior = "equal",
  levels = c("class1", "class2"),
  feature_names = NULL
)
```

Arguments

`location1`, `location2`	Finite class-location vectors.
`covariance1`, `covariance2`	Optional class covariance matrices.
`precision1`, `precision2`	Optional class precision matrices. For each class, supply exactly one covariance or precision matrix.
`prior`	`"equal"` or a positive numeric vector of length two.
`levels`	Two class labels.
`feature_names`	Optional unique feature names.

Value

A valid `hd_classifier_fit` with a canonical-log-ratio score.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Wiley.

Examples

```
fit <- gaussian_qda_oracle(
  c(1, 0), c(-1, 0), covariance1 = diag(2),
  covariance2 = diag(c(2, 1))
)
predict(fit, rbind(c(1, 0), c(-1, 0)), type = "score")
```

`gaussian_wilks_independence_test`

Gaussian Wilks block-independence test

Description

Computes the classical Gaussian likelihood-ratio test for independence of two vector blocks. If $\hat{\rho}_1, \ldots, \hat{\rho}_m$ are the sample canonical correlations, $m = \min(p, q)$, then

$$\Lambda = \prod_{j=1}^{m} (1 - \hat{\rho}_j^2)$$

and the Bartlett statistic is

$$-\{n - 1 - (p + q + 1)/2\} \log \Lambda,$$

calibrated against χ^2_{pq}. This is a fixed-dimensional, Gaussian benchmark; it is not a high-dimensional repair of Wilks' test.

Usage

```
gaussian_wilks_independence_test(x, y)
```

Arguments

`x`	Numeric n by p matrix for the first block.
`y`	Numeric n by q matrix for the second block.

Value

An object inheriting from `htest`. `components` contains Wilks' Lambda, its log value, canonical correlations, and the three sample covariance blocks.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*, 3rd edition, Chapter 8. Wiley.

Examples

```
x <- matrix(c(-2, 0, -1, 2, 0, -1, 1, 1, 2, -2, 3, 0), 6, 2)
y <- matrix(c(1, -2, 0, 2, -1, 3), 6, 1)
gaussian_wilks_independence_test(x, y)
```

`generalized_sign_pca` *Generalized spatial-sign principal component analysis*

Description

Forms the generalized spatial-sign covariance matrix

$$n^{-1}\sum_i g(x_i - T)g(x_i - T)', \qquad g(t) = \xi(\|t\|_2)t,$$

and decomposes it. Available radial multipliers are `"winsor"`, `"quadratic"`, `"ball"`, `"shell"`, `"linear_redescending"`, `"spatial"` (ordinary spatial signs), and `"identity"` (the centered second moment).

Usage

```
generalized_sign_pca(
  x,
  rank = NULL,
 weight = c("winsor", "quadratic", "ball", "shell", "linear_redescending", "spatial",
    "identity"),
  center = c("kstep_lts", "spatial", "mean", "none"),
  cutoff = c("median_mad", "original_h_order", "user"),
  cutoffs = NULL,
  q1_lower_bound = c("zero", "error"),
  zero_mad = c("limit", "error"),
  lts_steps = 2L,
  divisor = c("n", "nonzero"),
  zero_action = c("zero", "error"),
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  keep_operator = TRUE
)
```

Arguments

`x`	Numeric matrix or data frame with observations in rows.
`rank`	Number of loading vectors. `NULL` uses `min(n, p)`.
`weight`	Radial-multiplier family.
`center`	Numeric center or one of `"kstep_lts"`, `"spatial"`, `"mean"`, and `"none"`. K-step LTS starts from the certified Chapter 1 spatial median.
`cutoff`	One of the Chapter 6 book-defined `"median_mad"` rule, the 2019 `"original_h_order"` rule, or `"user"`.
`cutoffs`	For `cutoff = "user"`, four non-negative values named exactly `Q1`, `Q2`, `Q3`, and `Q3_star`; input order is arbitrary.

q1_lower_bound	"zero" explicitly applies the software lower bound $\max(0, m - s)$ before the power 3/2; "error" rejects a negative Wilson–Hilferty base.
zero_mad	"limit" uses the exact coincident-cutoff limit; "error" rejects zero transformed MAD.
lts_steps	Number of deterministic LTS C-steps when center = "kstep_lts".
divisor	"n" uses the Chapter 6 book definition; "nonzero" excludes only zero residuals from the divisor. Observations rejected by a radial weight remain part of either divisor.
zero_action	"zero" maps residuals no larger than zero_tol to zero; "error" rejects such a sample.
tol, max_iter	Spatial-median convergence controls.
zero_tol	Non-negative tolerance for a zero residual. The default detects exact zeros and preserves scale equivariance.
keep_operator	If TRUE, retain the decomposed SSCM.

Details

The default cutoff convention is the ordinary median/raw-MAD construction stated in Chapter 6 of the book. Because the cited 2024 manuscript was not available for formula verification, "median_mad" is documented here as a book-defined convention and is not claimed to reproduce that manuscript. "original_h_order" implements the 2019 order statistic with $h = \lfloor (n + p + 1)/2 \rfloor$ and fails when h > n. "user" requires all four named cutoffs. Ball includes Q2; Shell includes both Q1 and Q3. When the MAD is zero, the default exact limiting rule uses coincident cutoffs and never perturbs a denominator. The "spatial" and "identity" families use no cutoffs and reject an explicitly supplied cutoff or cutoffs argument.

For a general elliptical representation $X - \mu = RAU$, $U = Z/\|Z\|$, and $S = \sum_l \lambda_l Z_l^2$, the population generalized eigenvalue contains the radial variable:

$$E\{K^2(|R|\sqrt{S}/\|Z\|)\lambda_j Z_j^2/S\}.$$

The simpler expression without R is therefore not claimed outside the Gaussian representation. Eigenvector and ordering results require the conditions stated in Chapter 6 for this book-defined construction, not merely a measurable weight.

Value

An object inheriting from hd_pca_fit. eigenvalues contains the full spectrum and loadings contains the requested leading vectors.

References

Raymaekers, J. and Rousseeuw, P. J. (2019). A generalized spatial sign covariance matrix. *Journal of Multivariate Analysis*, 171, 94–111.

Chapter 6 cites Leyder, S., Raymaekers, J., and Verdonck, T. (2024), Generalized spherical principal component analysis, *Statistics and Computing*, 34, 104. That manuscript was unavailable for formula verification; the "median_mad" construction above is therefore attributed to the book rather than asserted as a reproduction of the cited article.

Examples

```
x <- rbind(c(3, 0), c(-3, 0), c(0, 1), c(0, -1))
generalized_sign_pca(
  x, rank = 1, weight = "winsor", center = "none"
)
```

`generic_weighted_hr_location`
Generic weighted Hettmansperger–Randles location estimator

Description

Solves the formula-complete weighted location equation in Chapter 2 while retaining the unweighted diagonal Hettmansperger–Randles (HR) scale equation. At iteration m, let

$$e_i^{(m)} = (D^{(m)})^{-1/2}(X_i - \theta^{(m)}), \quad r_i^{(m)} = \|e_i^{(m)}\|, \quad U_i^{(m)} = e_i^{(m)}/r_i^{(m)}.$$

The updates are

$$\theta^{(m+1)} = \theta^{(m)} + (D^{(m)})^{1/2} \frac{\sum_i K(r_i^{(m)}) U_i^{(m)}}{\sum_i K(r_i^{(m)})/r_i^{(m)}}$$

and

$$D^{(m+1)} = p D^{(m)} \operatorname{diag}\{n^{-1} \sum_i U_i^{(m)} U_i^{(m)T}\}.$$

Thus weights enter the location equation only; the scale update is always the unweighted HR update printed in the manuscript.

Usage

```
generic_weighted_hr_location(
  x,
  K = "constant",
  power = 0,
  initial_location = NULL,
  initial_diagonal = NULL,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  keep_history = FALSE
)
```

Arguments

x	Numeric matrix or data frame with observations in rows.
K	Radial-weight name or scalar R callback.
power	Finite exponent used only when K = "power".
initial_location	Optional finite initial location. The default is the vector of sample column means.
initial_diagonal	Optional strictly positive initial diagonal scale. The default is the vector of unbiased marginal sample variances.
tol	Strictly positive relative iterate tolerance.
max_iter	Positive maximum number of simultaneous location/scale updates.
zero_tol	Non-negative threshold for a singular standardized radius.
keep_history	Whether to retain all location and diagonal iterates.

Details

K may be an R function or one of the auditable built-ins "constant", "inverse_norm", and "power". A user callback is called separately on each scalar radius and must return exactly one finite numeric scalar. The recursion rejects zero radii, a non-positive weighted denominator, non-positive scale iterates, and non-convergence. It never floors a radius, caps a weight, flips a denominator sign, adds a ridge, or returns an unconverged last iterate.

Value

A list of class generic_weighted_hr_location containing the fitted location and diagonal, final directions/radii/weights, equation residuals, and strict convergence diagnostics.

References

Feng, L., Liu, B. and Ma, Y. (2021). An inverse norm sign test for location parameters in high-dimensional data. Journal of Business & Economic Statistics 39, 807–815.

Examples

```
x <- matrix(c(-2, 1, 0, 3, -1, 2, 1, -3, 2, 0, 4, -2), ncol = 2)
generic_weighted_hr_location(x, K = "constant", tol = 1e-6)
```

gqda_classifier *Generalized quadratic discriminant analysis for elliptical populations*

Description

Fits the Bose–Pal–Saha Ray–Nayak generalized QDA rule using classical class moments or two explicitly supplied certified location–scatter fits. The rule assigns an observation to class 1 when

$$\Delta_d^2(x) \geq c \log\{|S_1|/|S_2|\}, \qquad 0 \leq c \leq 1.$$

A supplied `c` is used directly. Otherwise the resubstitution selector (or an explicitly requested leakage-free classical cross-validation selector) enumerates both endpoints, every legal decision breakpoint, and adjacent midpoints. This direct loss calculation remains correct for positive, negative, and zero signed log-determinant contrasts.

Usage

```
gqda_classifier(
  x,
  y,
  class_fits = NULL,
  c = NULL,
  selection = base::c("resubstitution", "fixed", "cross_validation"),
  folds = 5L,
  seed = NULL,
  strict = TRUE
)
```

Arguments

`x`	Numeric training matrix with observations in rows.
`y`	Two-class response. Factor levels define class 1 then class 2; otherwise first appearance defines the order.
`class_fits`	Optional length-two list of certified fits, each providing finite `location` and strictly positive-definite `scatter`. A compatible precision may also be supplied and is checked against the scatter.
`c`	Optional fixed constant in `[0,1]`.
`selection`	One of `"resubstitution"`, `"fixed"`, or `"cross_validation"`. Supplying `c` selects `"fixed"`. Cross-validation is available only for internally refitted classical moments.
`folds`	Number of stratified folds for `"cross_validation"`.
`seed`	Optional seed for stratified fold shuffling. The caller's RNG state is restored. With `NULL`, input-order round-robin folds are used.
`strict`	If `TRUE`, numerical/certificate failure is an error; otherwise an invalid, non-predictable fit is returned with a warning.

Details

This is an equal-prior classifier. The current book's HR-QDA subsection writes an ordinary Gaussian plug-in QDA score, but the cited Yan–Feng–Zhang primary applies the generalized threshold above. No simulation tuning grid, ridge, determinant absolute value, or covariance repair is used.

Value

An `hd_classifier_fit`. Package scores are positive for class 1.

References

Bose, S., Pal, A., SahaRay, R., and Nayak, J. (2015). Generalized quadratic discriminant analysis. *Pattern Recognition*, 48(8), 2676–2684. doi:10.1016/j.patcog.2015.02.016.

Examples

```
x <- rbind(c(-2, 0), c(-1, 1), c(-1, -1), c(-2, 1),
           c(2, 0), c(1, 1), c(1, -1), c(2, -1))
y <- factor(rep(c("left", "right"), each = 4),
            levels = c("left", "right"))
fit <- gqda_classifier(x, y, c = 0)
predict(fit, x)
```

`grs_alpha_test` *Gibbons–Ross–Shanken exact alpha test*

Description

Tests whether all intercepts are zero in an unconditional linear factor-pricing model. Rows of returns and factors are observations and columns are assets and factors. The joint regression contains both an intercept and all factor columns. With unrestricted OLS residual matrix E, Vhat = E' E / T, and h the residual from regressing the all-ones vector on the factors, the implemented primary statistic is

$$[(T - N - K)/N][h'h/T]\hat{\alpha}'\widehat{V}^{-1}\hat{\alpha},$$

with the exact F distribution having N and T-N-K degrees of freedom under the Gaussian GRS assumptions.

Usage

```
grs_alpha_test(returns, factors = NULL)
```

Arguments

`returns`	Finite observation-by-asset numeric matrix or data frame.
`factors`	NULL, a finite length-T numeric vector, or a finite observation-by-factor numeric matrix or data frame.

Details

This differs from two finite-sample expressions in the book draft. Its displayed slope estimator omits the intercept although its residual formula includes one. It also divides the residual covariance by T-K-1 while retaining the primary statistic scaling for a divisor-T covariance, missing the compensating factor T/(T-K-1). This function follows GRS exactly and reports both divisors for audit.

Value

An object of class c("hd_alpha_test", "htest") with the exact statistic, p-value, fitted intercepts, and OLS diagnostics.

References

Gibbons, M. R., Ross, S. A., and Shanken, J. (1989). A test of the efficiency of a given portfolio. Econometrica, 57, 1121–1152. doi:10.2307/1913625.

Examples

```
f <- cbind(seq(-1, 1, length.out = 10))
y <- cbind(0.2 + f[, 1] + sin(seq_len(10)),
           -0.1 - 0.5 * f[, 1] + cos(seq_len(10)))
grs_alpha_test(y, f)
```

`hallin_paindaveine_shape_test`

Hallin–Paindaveine signed-rank test for elliptical shape

Description

For null shape V_0, the observations are sphericized, their radii are ranked, and their directions are weighted by a score `K`. The statistic is Hallin and Paindaveine's equation (4.3),

$$\frac{p(p+2)}{2nE\{K^2(U)\}} \sum_{i,j} K(R_i/(n+1))K(R_j/(n+1))\{(U_i'U_j)^2 - 1/p\}.$$

It has an asymptotic chi-squared reference with $p(p+1)/2-1$ degrees of freedom. Scores are `"sign"` (constant), `"wilcoxon"` (power one), `"spearman"` (power two), and `"vdw"` (chi-squared normal scores). Non-sign scores require distinct radii, as in the paper's continuous elliptical model. No random tie breaking is used.

Usage

```
hallin_paindaveine_shape_test(
  x,
  shape0 = NULL,
  center = NULL,
```

```
  score = c("sign", "wilcoxon", "spearman", "vdw"),
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0
)
```

Arguments

x	Numeric matrix with observations in rows.
shape0	Null shape matrix. Multiplying it by a positive scalar has no effect.
center	A supplied finite center vector. NULL estimates the spatial median.
score	Signed-rank score family.
tol, max_iter, zero_tol	Controls passed to spatial_median() when the center is estimated.

Value

An hd_covariance_test object. components retains scores, ranks, signs, radii on the log scale, and the weighted sign scatter matrix.

References

Hallin, M. and Paindaveine, D. (2006). *Annals of Statistics*, 34, 2707–2756. doi:10.1214/009053606000000731.

Examples

```
set.seed(38)
x <- matrix(rnorm(30), nrow = 15, ncol = 2)
hallin_paindaveine_shape_test(
  x, diag(2), center = c(0, 0), score = "sign"
)
```

---

hd_classifier_fit *Construct a classifier from a scoring function*

Description

This low-level constructor creates a two-class hd_classifier_fit from a user-supplied scoring function. The function must accept a finite numeric matrix with observations in rows and return one finite score per row. Positive scores select the first class. Package methods use transparent structured linear or quadratic score models instead of this custom route.

Usage

```
hd_classifier_fit(
  score,
  n_features,
  levels = c("class1", "class2"),
  feature_names = NULL,
  method = "Custom two-class classifier",
  score_scale = "method_threshold",
  tie = c("class1", "class2")
)
```

Arguments

score	Function mapping an observation matrix to numeric scores.
n_features	Positive number of input features.
levels	Two distinct class labels; the first is selected by a positive score.
feature_names	Optional unique feature names.
method	Descriptive method name.
score_scale	Description of the numerical score scale.
tie	Whether score zero selects the first or second class.

Value

An object of class `hd_classifier_fit`.

References

Feng, L. (2026). *High-Dimensional Data Analysis for Elliptically Symmetric Distributions*, Chapter 5 (book manuscript). This low-level constructor contract is package infrastructure, not a distinct method.

Examples

```
fit <- hd_classifier_fit(function(z) z[, 1] - z[, 2], 2)
predict(fit, rbind(c(2, 1), c(0, 1)))
```

`high_dimensional_hr` *High-dimensional Hettmansperger–Randles estimation*

Description

Implements Algorithm 2 of Yan, Feng, and Zhang (2025). Starting from the sample spatial median and a supplied positive-definite pilot precision, the method jointly updates location and trace-normalized shape. At iteration k, with $e_i^{(k)} = \Sigma_k^{-1/2}(X_i - \mu_k)$, it applies

$$\mu_{k+1} = \mu_k + \Sigma_k^{1/2} \frac{n^{-1}\sum_i U(e_i^{(k)})}{n^{-1}\sum_i \|e_i^{(k)}\|^{-1}}$$

and

$$\Sigma_{k+1} \propto \Sigma_k^{1/2} \mathcal{B}_h \left\{ n^{-1} \sum_i U(e_i^{(k)}) U(e_i^{(k)})^T \right\} \Sigma_k^{1/2}, \qquad \mathrm{tr}(\Sigma_{k+1}) = p.$$

Usage

```
high_dimensional_hr(
  x,
  pilot_precision,
  bandwidth = 3L,
  tol = 1e-08,
  max_iter = 1000L,
  median_tol = 1e-08,
  median_max_iter = 1000L,
  zero_tol = 0,
  scale_estimator = c("paper_qda", "none"),
  strict = TRUE
)
```

Arguments

`x`	Numeric $n \times p$ data matrix.
`pilot_precision`	A symmetric positive-definite $p \times p$ pilot precision matrix, or a valid object returned by `spatial_sign_precision()`. No automatic tuning constant is invented.
`bandwidth`	Non-negative integer less than p. The paper uses 3. If omitted, the resolved default is `min(3, p - 1)`.
`tol`	Positive relative tolerance for the joint location/shape fixed point.
`max_iter`	Positive maximum number of joint updates.
`median_tol`	Positive relative tolerance for the initial sample spatial median.
`median_max_iter`	Positive maximum number of spatial-median updates.
`zero_tol`	Non-negative radius threshold treated as coincidence. The default zero applies no positive floor.
`scale_estimator`	Either `"paper_qda"` for the primary paper's QDA covariance-trace add-on, or `"none"` for shape-only output.
`strict`	If `TRUE`, method failures are errors. If `FALSE`, they are warnings followed by an explicitly invalid fit with `estimate = NULL`.

Details

The paper sets $h = 3$; the formula permits $0 \leq h < p$. When the default is not explicitly supplied and $p \leq 3$, it resolves to `min(3, p - 1)` and records that value. The pilot's common scalar is not identified by spatial signs, so its inverse is normalized to trace p before iteration.

The primary estimator is a shape estimator, not a covariance-scale estimator. With `scale_estimator = "paper_qda"`, this function additionally implements the paper's QDA scale add-on

$$\widehat{\mathrm{tr}(\Xi)} = \{\sum_i \|X_i\|^2 - n\|\bar{X}\|^2\}/(n-1),$$

returning `scatter.scale = trace.hat / p` and `scatter = scatter.scale * shape`. This second-moment scale is not part of the robust sign fixed point and requires its finite double representation.

The paper says to repeat until convergence but does not prescribe a norm. Here, convergence means that the maximum of the relative location and Frobenius shape-map updates is at most `tol`. The spatial-sign score and shape-map residual are returned separately; they are diagnostics rather than an independently claimed estimating-equation certificate.

A supplied `spatial_sign_precision()` fit is accepted only when its feasibility/KKT certificate made it a valid fit. A raw matrix must be symmetric positive definite. Hard banding itself can destroy positive definiteness. Such a failure, a singular pilot, an exact zero standardized residual, overflow, or nonconvergence is reported without a ridge, eigenvalue floor, pseudoinverse, jitter, or perturbation. With `strict = FALSE`, every such method failure returns `estimate = NULL` and `valid = FALSE`; the last iterate appears only under diagnostics.

The book's fixed-pilot score, post-hoc banded raw shape, second banding of the inverse, and HR-centered divisor-n scale are not Algorithm 2. Likewise, the book's displayed $r_n + h^{-\alpha}$ theorem is a review synthesis rather than a finite-sample guarantee stated in this primary.

Value

An object of class `high_dimensional_hr_fit`. A valid object returns robust `location`, the final trace-p `shape`, the unbanded `raw.shape` fixed-point map, `precision = solve(shape)`, the final SSCMs, optional covariance `scatter.scale` and `scatter`, and detailed median, iteration, score, equation, positive-definiteness, reciprocal-condition, scale-identification, and pilot-certificate diagnostics.

References

Yan, G., Feng, L., and Zhang, X. (2025). *High-Dimensional Hettmansperger-Randles Estimator and its Applications*. arXiv:2505.01669. https://arxiv.org/abs/2505.01669

Examples

```
x <- rbind(
  c(2, 0), c(-2, 0), c(0, 1), c(0, -1),
  c(1, 1), c(-1, -1), c(1, -1), c(-1, 1)
)
high_dimensional_hr(x, diag(2), bandwidth = 0)
```

`hotelling_tests` *Classical Hotelling location tests*

Description

Performs the exact Gaussian one- or two-sample Hotelling T^2 test. Observations are rows and variables are columns. The two-sample test uses the pooled covariance matrix and therefore assumes a common population covariance matrix.

Usage

```
hotelling_one_sample_test(x, mu = NULL, tol = sqrt(.Machine$double.eps))

hotelling_two_sample_test(x, y, tol = sqrt(.Machine$double.eps))
```

Arguments

`x`	A numeric matrix or data frame with observations in rows.
`mu`	For the one-sample test, a finite null mean vector with one value per column of `x`. `NULL` uses the zero vector.
`tol`	Reciprocal-condition-number tolerance used after Cholesky factorization. It must be strictly between zero and one.
`y`	For the two-sample test, a second numeric matrix or data frame with observations in rows and the same variables as `x`.

Details

The covariance system is solved by a Cholesky factorization. A generalized inverse is deliberately not used: singular or numerically ill-conditioned covariance matrices invalidate the exact F calibration and produce an error directing the user to a high-dimensional test.

Value

An object of classes `hd_location_test` and `htest`. In addition to the standard `htest` fields, it contains the raw Hotelling `T2`, the covariance estimate, exact null-distribution metadata, and numerical diagnostics including the reciprocal condition number and solver.

References

Hotelling, H. (1931). The generalization of Student's ratio. *Annals of Mathematical Statistics*, **2**, 360–378.

Examples

```
set.seed(11)
x <- matrix(rnorm(60), 20, 3)
hotelling_one_sample_test(x)

y <- matrix(rnorm(75, 0.25), 25, 3)
hotelling_two_sample_test(x, y)
```

`hr_estimator` *Hettmansperger-Randles affine-equivariant location and shape*

Description

Alternates the location and Tyler-type shape updates stated in Chapter 1. The returned shape has trace equal to the data dimension.

Usage

```
hr_estimator(
  x,
  initial_location = NULL,
  initial_shape = NULL,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  warn = TRUE
)
```

Arguments

`x`	Observations in rows.
`initial_location`	Optional starting location. The default begins with the spatial median and deterministically falls back to a noncoincident sample mean or leave-one-out mean when the median equals an observation.
`initial_shape`	Optional positive definite starting shape.
`tol`	Tolerance for both normalized estimating-equation residuals.
`max_iter`	Maximum number of alternating iterations.
`zero_tol`	Tolerance used to detect zero whitened residuals. The default detects exact zeros only.
`warn`	If `TRUE`, warn when the iteration does not converge.

Value

An object of class hd_hr containing location, shape, and convergence diagnostics. In one dimension the method is defined by the conventional sample median and unit shape. This univariate extension does not report the multivariate joint shape-equation residual, which is returned as NA.

References

Hettmansperger, T. P. and Randles, R. H. (2002). A practical affine equivariant multivariate median. *Biometrika*, 89, 851-860.

Examples

```
set.seed(5)
x <- matrix(rt(240, df = 4), 60, 4)
hr_estimator(x)
```

---

hr_gqda *High-dimensional HR generalized quadratic discriminant analysis*

Description

Fits the primary Yan–Feng–Zhang classwise high-dimensional HR estimator, including its separate QDA second-moment scale, and plugs the two certified fits into generalized QDA. This is not the ordinary Gaussian HR-QDA score currently displayed in the book.

Usage

```
hr_gqda(
  x,
  y,
  pilot_precision,
  bandwidth = 3L,
  c = NULL,
  selection = base::c("resubstitution", "fixed", "cross_validation"),
  folds = 5L,
  seed = NULL,
  tol = 1e-08,
  max_iter = 1000L,
  median_tol = 1e-08,
  median_max_iter = 1000L,
  zero_tol = 0,
  strict = TRUE
)

hr_qda(...)
```

Arguments

x	Numeric training matrix with observations in rows.
y	Two-class response. Factor levels define class 1 then class 2; otherwise first appearance defines the order.
pilot_precision	One pilot precision matrix/certified fit shared by both classes, or a length-two list of class-specific pilots.
bandwidth	One or two hard-banding widths passed to high_dimensional_hr().
c	Optional fixed constant in [0,1].
selection	One of "resubstitution", "fixed", or "cross_validation". Supplying c selects "fixed". Cross-validation is available only for internally refitted classical moments.
folds	Number of stratified folds for "cross_validation".
seed	Optional seed for stratified fold shuffling. The caller's RNG state is restored. With NULL, input-order round-robin folds are used.
tol, max_iter, median_tol, median_max_iter, zero_tol	Iteration controls passed to every full-sample and cross-validation HR refit.
strict	If TRUE, numerical/certificate failure is an error; otherwise an invalid, non-predictable fit is returned with a warning.
...	Arguments passed unchanged to hr_gqda().

Value

An hd_classifier_fit containing both certified HR fits.

References

Yan, G., Feng, L., and Zhang, X. (2025). *High-Dimensional Hettmansperger–Randles Estimator and its Applications*. arXiv:2505.01669. https://arxiv.org/abs/2505.01669

Examples

```
x <- rbind(c(-2, 0), c(-1, 1), c(-1, -1), c(-2, 1),
           c(2, 0), c(1, 1), c(1, -1), c(2, -1),
           c(-1.5, .4), c(1.5, -.4), c(-1.7, -.3), c(1.7, .3))
y <- factor(rep(c("left", "right"), each = 6))
hr_gqda(x, y, pilot_precision = diag(2), bandwidth = 0, c = 0)
```

`if_pca` *Influential features PCA clustering*

Description

Implements influential features PCA (IF-PCA): standardize every feature with its n - 1 sample standard deviation, rank features by

$$\psi_{n,j} = \sqrt{n} \sup_t |\widehat{F}_{n,j}(t) - \Phi(t)|,$$

optionally translate and rescale the scores by an empirical null, select features, compute the leading K - 1 left singular vectors, and cluster their rows.

Usage

```
if_pca(
  x,
  K,
  selection = c("fixed", "hct"),
  threshold = NULL,
  empirical_null = c("mean_sd", "median_mad", "none"),
  ks_convention = c("paper", "software"),
  hct_convention = c("paper", "software"),
  null_scores = NULL,
  null_cdf = NULL,
  null_reps = NULL,
  seed = NULL,
  truncate = FALSE,
  rank_tol = sqrt(.Machine$double.eps),
  kmeans_tol = 1e-10,
  kmeans_max_iter = 100L
)
```

Arguments

`x`	Finite numeric matrix with observations in rows.
`K`	Known number of clusters, at least two.
`selection`	Either "fixed" or higher-criticism thresholding ("hct").
`threshold`	Required non-negative adjusted-score threshold for fixed selection.
`empirical_null`	One of "mean_sd" (the primary empirical-null step), "median_mad" (the paper's robust variant), or "none".
`ks_convention`	Either the primary-paper or official-software KS scale.
`hct_convention`	For HCT, "paper" uses rank probability j/p and the first HC maximum; "software" uses j/(p+1) and the last maximum.

null_scores	Optional finite reference scores already on the same final scale as the adjusted observed scores.
null_cdf	Optional CDF function on that final scale.
null_reps	Optional explicit number of null Monte Carlo replicates. There is deliberately no default.
seed	Required when null_reps is supplied. The caller's RNG state is restored after calibration.
truncate	If TRUE, apply the paper's theoretical entrywise bound log(p)/sqrt(n) to the left singular vectors. Numerical work in the primary paper and official software did not use it.
rank_tol	Explicit non-negative relative singular-value tolerance used only to certify that selected data have rank at least K - 1.
kmeans_tol	Positive tolerance for deterministic Lloyd updates.
kmeans_max_iter	Maximum deterministic Lloyd updates.

Details

ks_convention = "paper" evaluates the displayed statistic on the n - 1 standardized data. "software" reproduces the distinct official MATLAB score convention by dividing those standardized values once more by sqrt(1 - 1/n), which is equivalent to using divisor n inside the KS calculation. This distinction does not change the PCA matrix. The book's printed n-divisor standardization is not used because the primary method and software both form the PCA matrix with the n - 1 SD.

With fixed selection, threshold is applied to the adjusted scores using a greater-than-or-equal rule. HCT requires either null_scores already on the same final scale, a supplied null_cdf, or explicit null_reps plus seed. A standard KS CDF is not silently substituted because centering and scale are estimated. The original software used 100*p null replicates and the paper's numerical study used 2000*p; neither computational choice is hidden as a default.

Value

An if_pca_fit object with labels, selected indices and names, standardized data, raw and adjusted KS scores, the embedding, deterministic k-means diagnostics, and full HCT search details when used.

References

Jin, J. and Wang, W. (2016). Influential features PCA for high dimensional clustering. *Annals of Statistics*, **44**, 2323–2359. doi:10.1214/15AOS1423.

Examples

```
x <- cbind(
  c(-3, -2.5, -2, -1.5, 1.5, 2, 2.5, 3),
  c(-1, 0, 1, 0, -1, 0, 1, 0),
  c(0, 1, 0, -1, 0, 1, 0, -1)
)
```

```
if_pca(x, K = 2, selection = "fixed", threshold = 0,
       empirical_null = "none")
```

`independence_classifier` *Diagonal-covariance independence classifier*

Description

Fits the Gaussian independence rule after replacing the common covariance by its pooled diagonal. `variance_divisor = "unbiased"` uses `n - 2`; `"mle"` uses `n`. Zero marginal variances are never floored. They either cause a failure or are dropped explicitly.

Usage

```
independence_classifier(
  x,
  y,
  prior = "equal",
  variance_divisor = c("unbiased", "mle"),
  zero_variance = c("error", "drop"),
  strict = TRUE
)
```

Arguments

`x`	Numeric matrix or all-numeric data frame, observations in rows.
`y`	Two-class label vector or factor.
`prior`	`"equal"`, `"empirical"`, or a positive numeric vector of length two. Named entries must match the class levels.
`variance_divisor`	Pooled marginal-variance convention.
`zero_variance`	Whether an exactly non-positive pooled marginal variance is an error or is explicitly dropped.
`strict`	If `TRUE`, a method failure is an error. If `FALSE`, it is a warning followed by an invalid `hd_classifier_fit`.

Value

An `hd_classifier_fit`.

References

Bickel, P. J. and Levina, E. (2004). Some theory for Fisher's linear discriminant function, naive Bayes, and some alternatives when there are many more variables than observations. *Bernoulli*, 10, 989–1010.

Examples

```
x <- rbind(c(2, 1), c(1, 2), c(3, 0),
           c(-2, -1), c(-1, -2), c(-3, 0))
independence_classifier(x, rep(c("A", "B"), each = 3))
```

`inst_one_sample_test` *Feng–Liu–Ma one-sample inverse norm sign test*

Description

Tests $H_0 : \theta = \mu_0$ against an unrestricted multivariate location alternative with the inverse norm sign test (INST) of Feng, Liu, and Ma (2021). For every unordered pair $i < j$, a location and positive diagonal scale are jointly fitted from the sample with observations i, j removed. Writing the fitted scale as $\widehat{D}_{ij}$, $r_{ij,k} = \|\widehat{D}_{ij}^{-1/2}(X_k - \mu_0)\|$, and $U_{ij,k} = U\{\widehat{D}_{ij}^{-1/2}(X_k - \mu_0)\}$, the primary raw statistic is

$$T_{INST} = \frac{2}{n(n-1)} \sum_{i<j} r_{ij,i}^{-1} r_{ij,j}^{-1} U_{ij,i}^{\mathsf{T}} U_{ij,j}.$$

The pair-specific fitted location is used only to estimate $\widehat{D}_{ij}$. It is deliberately not subtracted from either endpoint in this statistic.

Usage

```
inst_one_sample_test(
  x,
  mu = NULL,
  alpha = 0.05,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

`x`	A numeric matrix or data frame with observations in rows and variables in columns. At least four observations are required.
`mu`	A finite numeric vector giving the null location. The default is the zero vector.
`alpha`	Significance level for the returned upper-tail rejection rule.
`tol`	Positive tolerance for relative location and log-diagonal updates in every leave-two-out fit.
`max_iter`	Positive maximum number of updates for each fit.
`zero_tol`	Non-negative tolerance below which a standardized radius is treated as singular. The default detects exact zeros only.
`strict`	If `TRUE`, error when any leave-two-out iteration is not stable. If `FALSE`, warn and return the last iterates.

Details

The operational calibration is the direct feasible variance in Section S.3 of the paper's official supplement. If $\widetilde{\mu}_{ij} = (n-2)^{-1}\sum_{k\neq i,j} U_{ij,k}$, then

$$\widehat{\sigma}_n^2 = 2n^{-4}\sum_{i\neq j} r_{ij,i}^{-2} r_{ij,j}^{-2}\{(U_{ij,i}-\widetilde{\mu}_{ij})^{\mathsf{T}}U_{ij,j}\}\{(U_{ij,j}-\widetilde{\mu}_{ij})^{\mathsf{T}}U_{ij,i}\}.$$

This is an ordered-pair sum with the published $2n^{-4}$ multiplier; it is not replaced by a finite-sample pair-count denominator. The reported p-value is based only on $T_{INST}/\widehat{\sigma}_n$.

For interpretation, the result additionally reports cross-fitted estimates of $E(r^{-1})$, $\nu_{2,IN} = E(r^{-2})$, $c_{0,IN} = E(r^{-2})$, and $\mathrm{tr}(R^2)$. The displayed oracle-factorized variance $2\widehat{\nu}_{2,IN}^2\widehat{\mathrm{tr}(R^2)}/\{n(n-1)p^2\}$ is diagnostic only and never replaces the primary direct variance.

Each diagonal HR recursion starts at the leave-two-out sample mean and marginal sample variances. The common scale from that initialization is retained because the fixed-point equations identify only relative diagonal scales. `iteration.stable` means that both the relative location update and log-diagonal update are at most `tol`; the estimating-equation `score.residual` is returned separately. With `strict = TRUE`, any required fit that is not stable after `max_iter` updates is an error. With `strict = FALSE`, the last iterates are used with a warning and complete diagnostics.

Exact or tolerance-defined zero radii make inverse norm weighting singular. Such a case is an error. A non-finite or non-positive published variance is also an error. No radius perturbation, weight cap, ridge, absolute value, numerical floor, or variance substitution is applied.

Null-centred residuals are internally divided by a positive scale in each coordinate. This is algebraically neutral under a common translation and nonsingular diagonal change of units, and protects the calculation from avoidable overflow. Returned input-coordinate diagonals are canonical displays with largest entry one; their canonical log values retain scale ratio information when a displayed entry underflows. The unnormalised internal diagonals retain the initialization's common scale.

Value

An object of class `c("hd_location_test", "htest")`. Its `components` field contains the primary statistic and direct variance, cross-fitted radial moments and trace diagnostics, every pair's kernels, radii, fitted locations, and diagonal scales. Its `diagnostics` field records iteration stability, score residuals, centring roles, rejection rule, numerical scaling, and the no-repair contract.

References

Feng, L., Liu, B., and Ma, Y. (2021). An inverse norm sign test of location parameter for high-dimensional data. *Journal of Business & Economic Statistics*, **39**, 807–815. doi:10.1080/07350015.2020.1736084.

Feng, L., Liu, B., and Ma, Y. (2020). Supplementary material for *An inverse norm sign test of location parameter for high-dimensional data*. doi:10.6084/m9.figshare.11914095.v2.

Examples

```
set.seed(2021)
x <- matrix(stats::rt(40, df = 5), 10, 4)
inst_one_sample_test(x, tol = 1e-6)
```

`jiang_da_qda` *Jiang-Wang-Leng direct sparse QDA*

Description

Implements the direct quadratic-discriminant estimator of Jiang, Wang and Leng. The interaction estimate minimizes a penalized quadratic trace loss, is symmetrized after optimization, and the linear coefficient minimizes its lasso quadratic loss. This is deliberately not the Dantzig program sometimes misattributed to this paper. The intercept exhaustively checks sorted raw training-score breakpoints and intervening intervals for minimum 0-1 loss.

Usage

```
jiang_da_qda(
  x,
  y,
  lambda_interaction = NULL,
  lambda_linear = NULL,
  parameter_grid = NULL,
  folds = 5L,
  rho = 1,
  solver_tol = 1e-07,
  solver_max_iter = 10000L,
  strict = TRUE
)
```

Arguments

`x`	Numeric training matrix with observations in rows.
`y`	Two-class response. Its first observed level is class 1.
`lambda_interaction`, `lambda_linear`	Explicit non-negative penalties. Supply both, or instead provide `parameter_grid`.
`parameter_grid`	Paired `lambda_interaction` and `lambda_linear` columns for deterministic stratified joint CV.
`folds`	Number of folds used only with a parameter grid.
`rho`	Positive ADMM penalty for the interaction loss.
`solver_tol`	Positive optimization and certificate tolerance.
`solver_max_iter`	Positive optimization iteration limit.
`strict`	Whether numerical failure is an error; otherwise an invalid `hd_classifier_fit` is returned with a warning.

Value

An `hd_classifier_fit`. Strictly positive scores select class 1; exact zero selects class 2, following the primary decision rule.

References

Jiang, B., Wang, X. and Leng, C. (2018). A direct approach for sparse quadratic discriminant analysis. *Journal of Machine Learning Research*, 19(31), 1-37.

Examples

```
x <- rbind(
  c(-2, 0), c(-1, 1), c(-1, -1), c(-2, 1),
  c(2, 0), c(1, 2), c(1, -2), c(2, 1)
)
y <- factor(rep(c("left", "right"), each = 4))
fit <- jiang_da_qda(x, y, 0.2, 0.2, solver_tol = 1e-6)
```

---

john_sphericity_test *John's classical trace test of Gaussian sphericity*

Description

Computes $U = p\,\mathrm{tr}(S^2)/\,\mathrm{tr}^2(S) - 1$ and uses $mpU/2$, with residual degrees of freedom m, as a chi-squared statistic with $(p-1)(p+2)/2$ degrees of freedom. The multiplier is p; the $p+2$ multiplier in the book draft belongs to the spatial-sign analogue, not John's covariance statistic.

Usage

```
john_sphericity_test(x, center = TRUE)
```

Arguments

x	Numeric matrix with observations in rows.
center	Whether to estimate and remove the mean. FALSE implements the known-zero-mean model.

Value

An hd_covariance_test object.

References

John, S. (1971). Some optimal multivariate tests. *Biometrika*, 58, 123–127.

Examples

```
set.seed(36)
x <- matrix(rnorm(60), nrow = 20, ncol = 3)
john_sphericity_test(x)
```

k_spatial_median *K-spatial-median clustering*

Description

Alternates exact sample spatial-median center updates with deterministic nearest-center assignments. The fitted objective is the sum of unsquared Euclidean distances. Empty-cluster repair is disabled by default.

Usage

```
k_spatial_median(
  x,
  K,
  init = "maxmin",
  first_index = 1L,
  tol = 1e-08,
  max_iter = 100L,
  spatial_max_iter = 500L,
  zero_tol = 0,
  empty_action = c("error", "farthest"),
  cycle_action = c("error", "return"),
  ties = "first",
  keep_distances = FALSE
)
```

Arguments

x	Numeric observation-by-variable matrix or data frame.
K	Number of clusters.
init	Either "maxmin", K distinct row indices, or a finite K by p center matrix.
first_index	First max-min seed when init = "maxmin".
tol	Positive spatial-median equation tolerance.
max_iter	Positive outer iteration limit.
spatial_max_iter	Positive modified-Weiszfeld iteration limit.
zero_tol	Non-negative zero-residual tolerance.
empty_action	"error" or the explicit deterministic "farthest" repair. The latter only donates from a cluster of size at least two.
cycle_action	Whether a detected repeated state is an "error" or is returned with a failed convergence certificate.
ties	Deterministic tie rule; currently only "first" is supported.
keep_distances	Whether to retain the final full distance matrix.

Value

A k_spatial_median_fit object with labels, centers, objective, and explicit convergence, tie, and repair diagnostics.

References

Zhao, P., Zhuang, D., and Feng, L. (2026). Sparse K-spatial-median clustering for high-dimensional data. arXiv:2605.00598.

Examples

```
x <- matrix(c(0, 2, 8, 10), ncol = 1)
k_spatial_median(x, K = 2, init = c(1, 4))
```

---

kendall_factor_number *Robust factor-number selection from multivariate Kendall eigenvalues*

Description

Implements the modified Kendall eigenvalue-ratio (MKER) and transformed contribution-ratio (MKTCR) selectors. With m = min(N, T) for N = p variables and T = n observations, the stabilized eigenvalues are

$$\widetilde{\lambda}_j = \lambda_j + c/\sqrt{m}.$$

MKER maximizes tilde_lambda[j] / tilde_lambda[j + 1]. For MKTCR,

$$V_j = \sum_{i=j+1}^{m} \widetilde{\lambda}_i, \qquad \frac{\log(1 + \widetilde{\lambda}_j / V_{j-1})}{\log(1 + \widetilde{\lambda}_{j+1} / V_j)}$$

is maximized. kmax and strictly positive c are always supplied, and an exact tie selects the smallest index.

Usage

```
kendall_factor_number(
  x,
  kmax,
  c,
  method = c("mker", "mktcr"),
  zero_tol = 0,
  zero_pair_action = c("error", "zero"),
  eigen_tol = sqrt(.Machine$double.eps)
)
```

Arguments

x	Numeric matrix or data frame with observations in rows.
kmax	Supplied positive upper bound, at most min(n, p) - 1.
c	Supplied strictly positive stabilization constant.
method	Either "mker" or "mktcr".
zero_tol	Non-negative tolerance for tied pairwise differences.
zero_pair_action	Either error on ties or retain their explicit zero contribution under the all-pairs denominator.
eigen_tol	Positive relative PSD tolerance.

Value

A kendall_factor_number object with the selected factor number, raw and stabilized eigenvalues, criterion path, and MKTCR tail sums.

References

Yu, L., He, Y. and Zhang, X. (2019). Robust factor number specification for large-dimensional elliptical factor models. arXiv:1808.09107.

Examples

```
x <- rbind(c(-3, -1, 0), c(-2, 1, 1), c(-1, -2, 0),
           c(1, 2, 0), c(2, -1, -1), c(3, 1, 0))
kendall_factor_number(x, kmax = 1, c = 0.1, method = "mker")
```

---

kendall_pca *Multivariate-Kendall principal component analysis*

Description

Decomposes the exact multivariate Kendall U-statistic over all unordered pairs. Its default tie convention matches spatial_kendall(): a tied difference contributes the zero matrix while the divisor remains choose(n, 2). The operator is translation invariant; center controls only the reported scores.

Usage

```
kendall_pca(
  x,
  rank = NULL,
  center = c("mean", "spatial", "none"),
  divisor = c("all_pairs", "nonzero_pairs"),
  ties = c("zero", "error"),
  tol = 1e-08,
```

```
  max_iter = 500L,
  zero_tol = 0,
  keep_operator = TRUE
)
```

Arguments

x	Numeric matrix or data frame with observations in rows.
rank	Number of loading vectors. NULL uses min(n, p).
center	A numeric score center or one of "mean", "spatial", and "none". It does not enter the Kendall operator.
divisor	"all_pairs" uses every unordered pair; "nonzero_pairs" excludes tied pairs from the divisor.
ties	"zero" gives tied differences zero contribution; "error" rejects any tied pair.
tol, max_iter	Spatial-median convergence controls.
zero_tol	Non-negative tolerance for a tied pairwise difference.
keep_operator	If TRUE, retain the decomposed SSCM.

Value

An object inheriting from hd_pca_fit. eigenvalues contains the full spectrum and loadings contains the requested leading vectors.

References

Han, F. and Liu, H. (2018). ECA: High-dimensional elliptical component analysis in non-Gaussian distributions. *Journal of the American Statistical Association*, 113, 252–268.

Examples

```
x <- rbind(c(2, 0), c(-2, 0), c(0, 1), c(0, -1))
kendall_pca(x, rank = 1)
```

---

li_chen_covariance_test

Li–Chen two-sample high-dimensional covariance equality test

Description

Computes the exact location-invariant leave-four-out statistics A_{n_1}, A_{n_2}, and the two-by-two leave-out cross statistic $C_{n_1 n_2}$ from Li and Chen (2012). Their combination

$$T_{LC} = A_{n_1} + A_{n_2} - 2C_{n_1 n_2}$$

is unbiased for $\mathrm{tr}(\Sigma_1 - \Sigma_2)^2$. Under the equality null, the primary feasible standard-error estimator is

$$\hat{\sigma}_0 = 2A_{n_1}/n_2 + 2A_{n_2}/n_1.$$

The published article labels this quantity once as $\hat{\sigma}_0^2$; its units, subsequent ratio-consistency theorem, and rejection rule show that it is the standard error. A non-positive estimate is rejected without an absolute value or floor.

Usage

```
li_chen_covariance_test(x, y)
```

Arguments

x, y	Numeric matrices with observations in rows and matching columns; each group needs at least four observations.

Value

An `hd_covariance_test` object retaining original-unit and internally scaled trace estimates.

References

Li, J. and Chen, S. X. (2012). *Annals of Statistics*, 40, 908–940. doi:10.1214/12AOS993.

Examples

```
set.seed(44)
x <- matrix(rnorm(24), nrow = 8, ncol = 3)
y <- matrix(rnorm(27, mean = 0.1), nrow = 9, ncol = 3)
li_chen_covariance_test(x, y)
```

---

li_shao_sparse_qda *Li-Shao thresholded sparse QDA*

Description

Fits the three-threshold rule of Li and Shao. Class covariances use the maximum-likelihood divisor n_k. Mean differences are retained only when strictly above the first threshold. Entries whose class difference is at most the second threshold are pooled; off-diagonal entries are then kept only when strictly above the third threshold. Diagonals are not removed.

Usage

```
li_shao_sparse_qda(
  x,
  y,
  threshold_mean = NULL,
  threshold_difference = NULL,
  threshold_covariance = NULL,
  selection = c("specified", "paper_bisection"),
  bisection_tol = NULL,
  ridge = 0,
  strict = TRUE
)
```

Arguments

x	Numeric training matrix with observations in rows.
y	Two-class response. Its first observed level is class 1.
threshold_mean, threshold_difference, threshold_covariance	Three finite non-negative thresholds, required with specified selection.
selection	Either "specified" or "paper_bisection".
bisection_tol	Explicit positive bisection stopping tolerance.
ridge	Explicit non-negative diagonal ridge after thresholding.
strict	Whether numerical failure is an error; otherwise an invalid hd_classifier_fit is returned with a warning.

Details

Paper bisection uses deterministic leave-one-out endpoint searches. Its tolerance must be supplied. A ridge is used only when explicitly positive.

Value

An hd_classifier_fit; non-negative scores select class 1.

References

Li, J. and Shao, J. (2015). Sparse quadratic discriminant analysis for high dimensional data. *Statistica Sinica*, 25, 457-473.

Examples

```
x <- rbind(
  c(-2, 0), c(-1, 1), c(-1, -1), c(-2, 1),
  c(2, 0), c(1, 2), c(1, -2), c(2, 1)
)
y <- factor(rep(c("left", "right"), each = 4))
fit <- li_shao_sparse_qda(
  x, y, threshold_mean = 0, threshold_difference = 0,
```

```
  threshold_covariance = 0, ridge = 0.1
)
```

`li_wang_zou_two_sample_sign_test` *Li–Wang–Zou simpler spatial-sign-based two-sample test*

Description

Tests equality of two multivariate location vectors using the simplified bias-corrected spatial-sign test (SST) of Li, Wang, and Zou (2016). For group $k = 1, 2$, let $(\widehat{\theta}_k, \widehat{D}_k)$ be the full-sample diagonal Hettmansperger–Randles fit satisfying

$$n_k^{-1} \sum_i U\{\widehat{D}_k^{-1/2}(X_{ki} - \widehat{\theta}_k)\} = 0$$

and

$$p n_k^{-1} \mathrm{diag}\left(\sum_i U_{ki} U_{ki}^{\mathsf{T}}\right) = I_p.$$

Unlike the earlier Feng–Zou–Wang statistic, this method uses no leave-out fit. Its uncorrected statistic is

$$T_n = -\frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} U\{\widehat{D}_1^{-1/2}(X_{1i} - \widehat{\theta}_2)\}^{\mathsf{T}} U\{\widehat{D}_2^{-1/2}(X_{2j} - \widehat{\theta}_1)\}.$$

Usage

```
li_wang_zou_two_sample_sign_test(
  x,
  y,
  alpha = 0.05,
  tol = 1e-07,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

`x`, `y`	Numeric matrices or data frames with observations in rows and the same variables in columns. The literal feasible formulas require at least two observations in each group; every group-specific marginal sample variance must be positive.
`alpha`	Significance level for the returned upper-tail rejection rule.
`tol`	Finite positive tolerance for both full-sample diagonal HR estimating equations.
`max_iter`	Positive maximum number of HR updates per group.
`zero_tol`	Non-negative tolerance below which a standardized radius is treated as zero. The default detects exact zeros only.
`strict`	If `TRUE`, error unless both fits satisfy the estimating equations within `tol`; if `FALSE`, warn and return the last iterates.

Details

Write $\widetilde{U}_{ki} = U\{\widehat{D}_k^{-1/2}(X_{ki} - \widehat{\theta}_k)\}$, $\widehat{c}_k = n_k^{-1}\sum_i \|\widehat{D}_k^{-1/2}(X_{ki} - \widehat{\theta}_k)\|^{-1}$, $\widehat{c}_{nk} = \widehat{c}_{3-k}/\widehat{c}_k$, and $\widehat{D}_n = \widehat{D}_1^{1/2}\widehat{D}_2^{1/2}$. The feasible bias terms in Proposition 1 are

$$\widehat{\mathrm{tr}(A_k)} = \widehat{c}_{nk}\,\mathrm{tr}(\widehat{D}_k\widehat{D}_n^{-1}), \qquad \widehat{\mu}_n = \frac{\widehat{\mathrm{tr}(A_1)}}{n_1 p} + \frac{\widehat{\mathrm{tr}(A_2)}}{n_2 p}.$$

The within-group squared-trace estimators are

$$\widehat{\mathrm{tr}(A_k^2)} = \frac{p^2\widehat{c}_{nk}^2}{n_k(n_k-1)}\sum_{a=1}^{n_k}\sum_{b\neq a}\{\widetilde{U}_{kb}^{\mathsf{T}}\widehat{D}_k\widehat{D}_n^{-1}\widetilde{U}_{ka}\}^2,$$

and the cross trace is

$$\widehat{\mathrm{tr}(A_3^{\mathsf{T}}A_3)} = \frac{p^2}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}(\widetilde{U}_{1i}^{\mathsf{T}}\widetilde{U}_{2j})^2.$$

The published feasible variance is

$$\widehat{\sigma}_n^2 = \frac{2\widehat{\mathrm{tr}(A_1^2)}}{n_1^2p^2} + \frac{2\widehat{\mathrm{tr}(A_2^2)}}{n_2^2p^2} + \frac{4\widehat{\mathrm{tr}(A_3^{\mathsf{T}}A_3)}}{n_1 n_2 p^2}.$$

In particular, the first two outer denominators are n_k^2; only the ordered-pair trace estimators use $n_k(n_k-1)$. The reported statistic is $Z = (T_n - \widehat{\mu}_n)/\widehat{\sigma}_n$.

The scientific alternative is two-sided location inequality, but the SST is quadratic in the location difference. Consequently, the original article rejects for large positive Z, and this function reports the upper standard-normal tail.

Each full-sample fit starts at its sample mean and marginal sample variances and iterates the two published estimating equations. The diagonal equations identify scale only up to a common multiplier; a unit geometric mean is imposed internally. All bias and variance combinations are invariant to that identification. `iteration.stable` means that both estimating-equation residuals are at most `tol`. With `strict = TRUE`, an unstable fit is an error; otherwise the last iterate is returned with a warning and full diagnostics.

The spatial-sign convention is $U(0) = 0$ for numerator terms. A zero training radius is nevertheless an error because the recursive location update and $\widehat{c}_k$ require inverse radii. No radius perturbation, weight cap, ridge, pseudoinverse, absolute-value repair, numerical floor, or variance substitution is applied.

Value

An object of class `c("hd_location_test", "htest")`. Its `components` field contains the uncorrected and bias-corrected statistics, bias and variance estimates, radial moments, three trace estimates, ordered/cross-pair kernels, both full-sample fits, signs, radii, and scale bridges. `diagnostics` records convergence, denominator conventions, upper-tail calibration, zero-sign use, numerical scaling, and the no-repair contract.

References

Li, Y., Wang, Z., and Zou, C. (2016). A simpler spatial-sign-based two-sample test for high-dimensional data. *Journal of Multivariate Analysis*, **149**, 192–198. doi:10.1016/j.jmva.2016.04.004.

Examples

```
set.seed(2016)
x <- matrix(stats::rt(32, df = 5), 8, 4)
y <- matrix(stats::rt(36, df = 5), 9, 4)
li_wang_zou_two_sample_sign_test(x, y, tol = 1e-6)
```

`linear_pool_covariance`

Linear or convex pooling of covariance matrices

Description

Estimates every group covariance as a non-negative linear combination of all unbiased group SCMs and, optionally, the identity. For target group `k`, coefficients solve

$$\min_{a} \ \frac{1}{2} a^{\mathsf{T}}(C + \Delta)a - C_{\cdot k}^{\mathsf{T}} a,$$

subject to the requested lower bounds. `method = "convex"` additionally imposes $1^{\mathsf{T}} a = 1$.

Usage

```
linear_pool_covariance(
  data,
  method = c("linear", "convex"),
  identity = TRUE,
  identity_lower = 0,
  tol = 1e-08,
  max_iter = 1000L,
  solver_tol = 1e-10,
  solver_max_iter = 10000L,
  strict = TRUE,
  keep_signs = FALSE
)
```

Arguments

`data`	A non-empty list of finite numeric sample matrices. A single matrix is accepted and wrapped as one group. Rows are observations; every group needs at least four rows, a common column dimension, matching column names, positive marginal variation, and positive spatial residual radii.
`method`	Either non-negative `"linear"` pooling or `"convex"` pooling with coefficients summing to one.
`identity`	Whether to add the identity as an additional target.
`identity_lower`	A non-negative scalar or one value per target group, giving the identity coefficient lower bound. The default zero does not reproduce the official code's numerical `1e-8` choice.

`tol`, `max_iter`	Spatial-median convergence controls.
`solver_tol`	Positive projected-gradient/KKT solver tolerance.
`solver_max_iter`	Positive QP iteration limit.
`strict`	Whether spatial-median or QP nonconvergence is an error.
`keep_signs`	Whether to retain each group's fitted sign matrix.

Details

Each SCM uses divisor $n_k - 1$. For group `k`, the spatial-median SSCM and positive residual radii give

$$\widehat{\gamma}_{0k} = \frac{pn_k}{n_k - 1}\{\mathrm{tr}(S^2_{sgn,k}) - 1/n_k\}, \qquad \widehat{\gamma}_k = \Pi_{[1,p]}(\widehat{\gamma}_{0k} - p\widehat{\delta}_k),$$

with the Zou et al. inverse-radial bias correction. Corrected marginal kurtosis follows `ollila_raninen_shrinkage_covaria` The real-valued diagonal MSE term is

$$\Delta_{kk} = p^{-1}\{\tau_{1k}\,\mathrm{tr}(S_k)^2 + (\tau_{1k} + \tau_{2k})p\eta_k^2\gamma_k\},$$

where $\tau_{1k} = 1/(n_k - 1) + \kappa_k/n_k$ and $\tau_{2k} = \kappa_k/n_k$. The diagonal of `C` is $\eta_k^2\gamma_k$; off-diagonal entries use $\mathrm{tr}(S_{sgn,k}S_{sgn,l})\,\mathrm{tr}(S_k)\,\mathrm{tr}(S_l)/p$.

The independent projected-FISTA solver adds no ridge or pseudoinverse. It returns feasibility, projected-gradient, KKT, complementarity-gap, eigenvalue and restart diagnostics. Singular convex objectives are allowed. Nonconvergence is an error by default; `strict = FALSE` returns the last feasible iterate with a warning and labels it uncalibrated.

Value

A `linear_pool_covariance` list with coefficient matrix, pooled estimates, SCMs, SSCMs, centers, radii, eta/gamma/kappa, `C`, `Delta`, scaled objective matrices, and per-target solver diagnostics.

References

Raninen, E., Tyler, D. E., and Ollila, E. (2022). Linear pooling of sample covariance matrices. *IEEE Transactions on Signal Processing*, 70, 659–672. doi:10.1109/TSP.2021.3139207.

Examples

```
x1 <- matrix(c(-2, 0, -1, 2, 0, -2, 1, 1, 2, -1, 3, 2,
               -1, -2, 2, 0), ncol = 2, byrow = TRUE)
x2 <- matrix(c(-1, 1, 0, 3, 1, -2, 2, 2, 3, 0, -2, -1,
               1, 2, 0, -3), ncol = 2, byrow = TRUE)
linear_pool_covariance(list(first = x1, second = x2))
```

`liu_feng_ma_spatial_sign_alpha_test` *Liu–Feng–Ma robust spatial-sign sum alpha test*

Description

Implements the primary heavy-tailed alpha test. Restricted factor residuals are standardized by the full-sample diagonal spatial-sign fixed point. If U_t denotes the resulting sign and h is the residualized intercept, the quadratic form is

$$Q = N(h'h)^{-1} \sum_{s \neq t} h_s h_t U_s' U_t.$$

The variance component is the primary split leave-two-out estimator. After deleting a pair, the remaining rows are split into chronological first and second halves, separate restricted slopes are fitted, and the two deleted residuals use the corresponding slopes. Its denominator is $(h'h)(h'h-1)$, not $(h'h)^2$.

Usage

```
liu_feng_ma_spatial_sign_alpha_test(
  returns,
  factors = NULL,
  bias = c("wild_bootstrap", "none", "supplied"),
  delta_q = NULL,
  bootstrap_reps = 100L,
  seed = NULL,
  keep_bootstrap = FALSE,
  tol = 1e-07,
  max_iter = 500L,
  zero_tol = 0
)
```

Arguments

`returns`	Finite observation-by-asset numeric matrix or data frame.
`factors`	NULL, a finite length-T numeric vector, or a finite observation-by-factor numeric matrix or data frame.
`bias`	Bias calibration: primary "wild_bootstrap", literal zero "none" for formula auditing, or a user "supplied" value.
`delta_q`	Finite supplied bias when bias is "supplied"; otherwise must be NULL.
`bootstrap_reps`	Positive number of Rademacher replicates. The primary recommendation is 100.
`seed`	NULL to use and advance the current random stream, or a non-negative integer for an isolated deterministic bootstrap.
`keep_bootstrap`	Whether to retain all bootstrap Q values.

`tol`	Positive diagonal fixed-point tolerance.
`max_iter`	Positive integer update limit.
`zero_tol`	Non-negative standardized zero-radius threshold; default zero uses the literal spatial-sign convention.

Details

The recommended feasible statistic is

$$(Q - \widehat{\delta}_Q)/\{2\widehat{\mathrm{tr}(R^2)}\}^{1/2}.$$

By default, $\widehat{\delta}_Q$ is the mean of bootstrap Q values formed by multiplying every unrestricted OLS residual entry by an independent asset-time Rademacher sign and adding the restricted fitted values. This is the method's inferential bias calibration, not a reproduction of its simulation study. An explicit seed is locally isolated: the caller's random-number state is restored even after an error. With a NULL seed the current stream advances.

The book draft omits the primary wild-bootstrap bias term, uses $(h'h)^2$ in the trace denominator, and describes pair-deleted scale fits rather than the paper's full-sample diagonal scale plus split pair-deleted slope fits. These choices are not silently mixed here. The book's weighted/INST construction splices this incompatible trace into a paper whose complete feasible calibration is not given in the cited material, so weighted/INST is intentionally review-only. Likewise, the dependent and Lq paragraphs do not specify a complete feasible centering, long-run variance, and calibration and are not reconstructed.

Value

An asymptotic upper-tail normal alpha-test object with every quadratic, trace, bias, scale, convergence, and bootstrap diagnostic.

References

Liu, B., Feng, L., and Ma, Y. (2023). High-dimensional alpha test of linear factor pricing models with heavy-tailed distributions. Statistica Sinica, 33, 1389–1410. doi:10.5705/ss.202021.0134.

Examples

```
f <- cbind(seq(-1, 1, length.out = 10))
y <- outer(seq_len(10), 1:3, function(i, j) sin(i + j / 2))
liu_feng_ma_spatial_sign_alpha_test(
  y, f, bootstrap_reps = 2, seed = 1
)
```

`lloyd_kmeans` *Lloyd's deterministic K-means algorithm*

Description

Minimize the Chapter 7 within-cluster sum of squared Euclidean distances by alternating exact sample-mean and nearest-center updates. Exact assignment ties use the smallest cluster index. The default `empty_action = "error"` applies no repair. The optional `"farthest"` rule moves the smallest-row maximizer of current within-cluster squared distance from a donor of size at least two, then recomputes centers.

Usage

```
lloyd_kmeans(
  x,
  clusters,
  initial = NULL,
  initialization = c("maxmin", "random"),
  first_index = 1L,
  seed = NULL,
  ties = "first",
  empty_action = c("error", "farthest"),
  solver_tol = 1e-10,
  solver_max_iter = 100L,
  strict = TRUE,
  keep_distances = FALSE
)
```

Arguments

`x`	Numeric observation matrix, with observations in rows.
`clusters`	Number of clusters.
`initial`	Optional center matrix or vector of distinct row indices.
`initialization`	Max-min or seeded random initialization when `initial` is `NULL`.
`first_index`	First max-min seed row.
`seed`	Explicit seed required for random initialization; otherwise `NULL`.
`ties`	Exact tie rule; only deterministic `"first"` is implemented.
`empty_action`	Error, or the explicit farthest-observation repair.
`solver_tol`	Tolerance used only to certify objective monotonicity.
`solver_max_iter`	Maximum Lloyd iterations.
`strict`	Error rather than return an uncertified iterate.
`keep_distances`	Retain the final squared-distance matrix.

Details

`initial` may be a `clusters` by `p` center matrix or `clusters` distinct row indices. Otherwise initialization is either deterministic max-min from the explicit `first_index`, or seeded sampling without replacement. Seeded initialization restores the caller's RNG state.

Value

A `lloyd_kmeans_fit` object with labels, centers, sizes, objective trace, initialization record, and convergence/repair certificates.

References

Feng, L. (2026). *High-Dimensional Data Analysis for Elliptical Symmetric Distributions*, Chapter 7 (book manuscript). This implementation follows the manuscript's explicit deterministic Lloyd-update contract.

For a foundational k-means formulation, see MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium*, 1, 281–297.

Examples

```
x <- rbind(c(-2, 0), c(-1, 0), c(2, 0), c(3, 0))
lloyd_kmeans(x, clusters = 2, initial = c(1, 3))
```

---

lpd_classifier *Linear programming discriminant classifier*

Description

Fits the Cai–Liu linear programming discriminant (LPD) direction

$$\widehat{\gamma} \in \arg\min_{\gamma} \|\gamma\|_1 : \|A\gamma - (\bar{x}_1 - \bar{x}_2)\|_\infty \le \lambda,$$

where A is the unbiased pooled within-class covariance plus the explicitly supplied $ridgeI$. The package score is

$$(z - (\bar{x}_1 + \bar{x}_2)/2)^\top \widehat{\gamma} + \log(\pi_1/\pi_2),$$

so a positive score selects class 1. The Dantzig solution is returned only after primal feasibility, dual feasibility, l1 stationarity, and a primal–dual gap are all certified.

Usage

```
lpd_classifier(
  x,
  y,
  lambda,
  prior = NULL,
```

```
  ridge = 0,
  solver_tol = 1e-07,
  solver_max_iter = 100000L,
  strict = TRUE,
  tie = c("class1", "class2")
)
```

Arguments

x	Numeric training matrix with observations in rows.
y	Binary response. Its first factor level is class 1.
lambda	Positive Dantzig constraint radius; it is never selected implicitly.
prior	Two positive class probabilities summing to one. NULL uses the shared classifier default.
ridge	Explicit non-negative ridge added to the pooled covariance. Zero implements the unmodified paper operator.
solver_tol	Positive tolerance required by every solver certificate.
solver_max_iter	Positive maximum number of primal–dual iterations.
strict	If TRUE, a failed certificate is an error. If FALSE, an explicitly invalid, non-predictable fit is returned with a warning.
tie	Which class receives a score exactly equal to zero.

Value

An object of class hd_classifier_fit.

References

Cai, T. and Liu, W. (2011). A direct estimation approach to sparse linear discriminant analysis. Journal of the American Statistical Association, 106, 1566–1577. doi:10.1198/jasa.2011.tm11199.

Examples

```
x <- rbind(
  c(2, 1), c(1, 2), c(2, 2), c(3, 1),
  c(-2, -1), c(-1, -2), c(-2, -2), c(-3, -1)
)
y <- factor(rep(c("first", "second"), each = 4))
fit <- lpd_classifier(x, y, lambda = 0.2, ridge = 0.1)
predict(fit, x, type = "score")
```

`ma_feng_wang_bao_conditional_alpha_test`
Ma–Feng–Wang–Bao conditional maximum and adaptive alpha test

Description

For a supplied restricted sieve fit, the primary marginal statistics are

$$t_i^2 = T^{-1}\widehat{\sigma}_{ii}^{-1}(\widehat{e}_{i\cdot}'1_T)^2, \qquad \widehat{\sigma}_{ij} = \widehat{e}_{i\cdot}'\widehat{e}_{j\cdot}/(T-d-1),$$

where `d` is the observed factor count, not the number of sieve columns. The maximum is centered by $2\log N - \log\log N$ and calibrated with cdf $F(x) = \exp\{-\pi^{-1/2}\exp(-x/2)\}$.

Usage

```
ma_feng_wang_bao_conditional_alpha_test(
  fit,
  factor_count = NULL,
  component = c("max", "sum", "adaptive")
)
```

Arguments

`fit`	A valid object from `conditional_alpha_sieve_fit()`.
`factor_count`	Optional non-negative observed factor count. It is inferred from a design constructed by `conditional_alpha_sieve_design()`; otherwise it is required.
`component`	One of `"max"`, `"sum"`, or `"adaptive"`.

Details

The adaptive test combines this maximum p-value and the complete `ma_lan_su_tsai_conditional_alpha_sum_test()` p-value by the primary Fisher statistic $-2(\log p_M + \log p_S)$ with a chi-squared distribution on four degrees of freedom. The book draft, and a later review paragraph, incorrectly attribute a Cauchy combination to this paper.

Value

A `conditional_alpha_test/htest` object retaining both component statistics, log p-values, the primary marginal divisor, and calibration diagnostics.

References

Ma, H., Feng, L., Wang, Z. and Bao, J. (2024). Adaptive testing for alphas in conditional factor models with high dimensional assets. *Journal of Business & Economic Statistics*, 42, 1356–1366. doi:10.1080/07350015.2024.2313543. Preprint: https://arxiv.org/abs/2307.09397.

Examples

```
tt <- seq(0, 1, length.out = 18)
f <- cbind(market = sin(1:18 / 3))
z <- conditional_alpha_sieve_design(cbind(tt, tt^2), f)
y <- cbind(sin(1:18), cos(1:18), sin(1:18 / 2))
fit <- conditional_alpha_sieve_fit(y, z)
ma_feng_wang_bao_conditional_alpha_test(fit, component = "max")
```

---

ma_lan_su_tsai_conditional_alpha_sum_test

Ma–Lan–Su–Tsai conditional high-dimensional alpha sum test

Description

Implements the complete feasible HDA statistic for a reusable restricted sieve fit. With $h = M_Z 1_T$ and restricted residuals $\widehat{e}_{it}$,

$$S_{NT} = \frac{1}{NT}\sum_i (\widehat{e}_{i\cdot}' 1_T)^2, \qquad \widehat{\mu}_{NT} = \frac{1}{NT}\sum_{i,t} \widehat{e}_{it}^2 h_t^2.$$

If $q = \mathrm{ncol}(Z)$ and $\widehat{\Sigma} = T^{-1}\sum_t (\widehat{e}_t - \bar{e})(\widehat{e}_t - \bar{e})'$, the implemented primary trace correction is

$$\widehat{\mathrm{tr}(\Sigma^2)} = \frac{T^2}{(T+q-1)(T-q)}\left\{\mathrm{tr}(\widehat{\Sigma}^2) - \frac{\mathrm{tr}^2(\widehat{\Sigma})}{T-q}\right\}.$$

The variance estimate is

$$\widehat{\sigma}_{NT}^2 = \frac{2\widehat{\mathrm{tr}(\Sigma^2)}}{N^2T^2}\sum_{t\neq s} h_t^2 h_s^2.$$

Usage

```
ma_lan_su_tsai_conditional_alpha_sum_test(fit)
```

Arguments

`fit`	A valid object from `conditional_alpha_sieve_fit()`.

Details

The book draft's expression using only oracle $N^{-1}\,\mathrm{tr}(\Omega)$ and $2N^{-2}\,\mathrm{tr}(\Omega^2)$ omits h and the feasible finite-sample trace correction. This function uses the primary estimator and rejects non-positive estimates; it never takes an absolute value or applies a floor.

Value

A `conditional_alpha_test/htest` object with the upper-tail normal calibration and every feasible centering/variance component.

References

Ma, S., Lan, W., Su, L. and Tsai, C.-L. (2020). Testing alphas in conditional time-varying factor models with high-dimensional assets. *Journal of Business & Economic Statistics*, 38, 214–227. doi:10.1080/07350015.2018.1482758.

Examples

```
tt <- seq(0, 1, length.out = 18)
z <- conditional_alpha_sieve_design(cbind(tt, tt^2))
y <- cbind(sin(1:18), cos(1:18), sin(1:18 / 2))
fit <- conditional_alpha_sieve_fit(y, z)
ma_lan_su_tsai_conditional_alpha_sum_test(fit)
```

---

mauchly_sphericity_test

Mauchly–Box likelihood-ratio test of Gaussian sphericity

Description

With residual Wishart degrees of freedom m, this function computes

$$V = |S| / (\mathrm{tr}(S)/p)^p$$

and the Box–Bartlett statistic

$$-\left\{ m - \frac{2p^2 + p + 2}{6p} \right\} \log V.$$

The denominator in the correction is p, not $p + 1$. The latter appears in the accompanying book draft and is a transcription error.

Usage

```
mauchly_sphericity_test(x, center = TRUE)
```

Arguments

`x`	Numeric matrix with observations in rows.
`center`	Whether to estimate and remove the mean. `FALSE` implements the known-zero-mean model.

Value

An `hd_covariance_test` object.

References

Mauchly, J. W. (1940). *Annals of Mathematical Statistics*, 11, 204–209. Wang, Q. and Yao, J. (2013). *Electronic Journal of Statistics*, 7, 2164–2192.

Examples

```
set.seed(35)
x <- matrix(rnorm(60), nrow = 20, ncol = 3)
mauchly_sphericity_test(x)
```

`nagao_identity_test` *Nagao's classical Gaussian identity-covariance test*

Description

Tests $H_0 : \Sigma = I_p$ with

$$\frac{m}{2}\,\mathrm{tr}(S - I_p)^2,$$

using the covariance with divisor equal to the residual Wishart degrees of freedom m. Its fixed-dimensional reference has $p(p+1)/2$ degrees of freedom. The factor one-half is missing from the book draft.

Usage

```
nagao_identity_test(x, center = TRUE)
```

Arguments

`x`	Numeric matrix with observations in rows.
`center`	Whether to estimate and remove the mean. `FALSE` implements the known-zero-mean model.

Value

An `hd_covariance_test` object.

References

Nagao, H. (1973). On some test criteria for covariance matrix. *Annals of Statistics*, 1, 700–709.

Examples

```
set.seed(37)
x <- matrix(rnorm(60), nrow = 20, ncol = 3)
nagao_identity_test(x)
```

normalize_shape *Normalize a shape matrix*

Description

Removes the unidentified scalar from a positive definite shape matrix.

Usage

```
normalize_shape(shape, method = c("trace", "determinant"))
```

Arguments

shape	A finite square numeric matrix.
method	"trace" gives trace equal to the dimension; "determinant" gives determinant one.

Value

The symmetrized, normalized matrix.

References

Fang, K.-T. and Anderson, T. W. (1990). *Statistical Inference in Elliptically Contoured and Related Distributions*. Allerton Press.

Examples

```
shape <- matrix(c(4, 1, 1, 1), 2, 2)
normalize_shape(shape, method = "determinant")
```

ollila_raninen_shrinkage_covariance *Ollila–Raninen elliptical shrinkage covariance estimator*

Description

Estimates a covariance matrix by

$$\widehat{\Sigma} = \widehat{\beta} S + (1 - \widehat{\beta}) \widehat{\eta} I_p,$$

where S is the unknown-mean unbiased sample covariance (divisor $n - 1$), $\widehat{\eta} = \mathrm{tr}(S)/p$, and

$$\widehat{\beta} = \frac{\widehat{\gamma} - 1}{\widehat{\gamma} - 1 + \widehat{\kappa}(2\widehat{\gamma} + p)/n + (\widehat{\gamma} + p)/(n - 1)}$$

.

Usage

```
ollila_raninen_shrinkage_covariance(
  x,
  sphericity = c("ell1", "ell2", "ell3"),
  tol = 1e-08,
  max_iter = 1000L,
  strict = TRUE,
  keep_signs = FALSE
)
```

Arguments

x	A finite real numeric matrix or data frame; observations are rows. At least four observations are required and every marginal variable must have positive sample variation.
sphericity	One of "ell1", "ell2", or "ell3".
tol, max_iter	Convergence controls passed to spatial_median() when Ell1 is needed.
strict	Whether spatial-median nonconvergence is an error. With FALSE, the last finite iterate is used with a warning.
keep_signs	Whether to retain fitted spatial signs for Ell1/Ell3.

Details

The corrected marginal excess kurtoses are

$$K_j = \frac{n-1}{(n-2)(n-3)}\{(n+1)g_{2j}+6\},$$

and $\widehat{\kappa} = \max\{-2/(p+2), p^{-1}\sum_j K_j/3\}$. Equality at the theoretical lower bound is retained and diagnosed; the 0.99 boundary modification in the authors' MATLAB code is deliberately not used.

sphericity = "ell1" uses the spatial-median SSCM estimator

$$\widehat{\gamma}_1^* = \frac{n}{n-1}\{p\,\mathrm{tr}(S_{sgn}^2) - p/n\}.$$

"ell2" uses

$$\widehat{\gamma}_2^* = b_n\left\{\frac{p\,\mathrm{tr}(S^2)}{\mathrm{tr}(S)^2} - a_n\frac{p}{n}\right\},$$

where $a_n = n(n/(n-1)+\widehat{\kappa})/(n+\widehat{\kappa})$ and $b_n = (n+\widehat{\kappa})(n-1)^2/[(n-2)\{3\widehat{\kappa}(n-1)+n(n+1)\}]$. Both are projected to [1, p] as prescribed in the publication. "ell3" selects the smaller projected sphericity, with ties assigned to Ell2.

Value

An hd_covariance_estimator containing the estimate, unbiased SCM, scale, both raw/projected sphericities, corrected kurtosis, data weight, shrinkage intensity, and no-repair diagnostics.

References

Ollila, E. and Raninen, E. (2019). Optimal shrinkage covariance matrix estimation under random sampling from elliptical distributions. *IEEE Transactions on Signal Processing*, 67, 2707–2719. doi:10.1109/TSP.2019.2908144.

Examples

```
x <- matrix(c(-2, 0, 1, -1, 2, 0, 0, -2, 1, 1, 1, -1,
              2, 1, 0, 3, -1, 2, -2, -1, -1, 1, 3, 1),
            ncol = 3, byrow = TRUE)
ollila_raninen_shrinkage_covariance(x, sphericity = "ell3")
```

`oracle_weighted_sign_sum_test` *Oracle generic weighted-sign sum statistic*

Description

Evaluates the Chapter 2 oracle statistic from a supplied reference location and diagonal shape. With

$$r_i = \|D^{-1/2}(X_i - \theta)\|, \quad V_i(K) = K(r_i)U\{D^{-1/2}(X_i - \theta)\},$$

the raw score is

$$T_n(K) = \frac{2}{n(n-1)} \sum_{i<j} V_i(K)^T V_j(K).$$

Usage

```
oracle_weighted_sign_sum_test(
  x,
  theta,
  diagonal,
  K = "constant",
  power = 0,
  null_sd = NULL,
  nu2 = NULL,
  trace_R2 = NULL,
  zero_tol = 0,
  keep_scores = FALSE
)
```

Arguments

`x`	Numeric matrix or data frame with observations in rows.
`theta`	Required finite reference location vector.
`diagonal`	Required strictly positive reference diagonal shape.
`K`	Radial-weight name or scalar R callback, as in `generic_weighted_hr_location()`.

power	Finite exponent used only when K = "power".
null_sd	Optional strictly positive supplied null standard deviation.
nu2	Optional strictly positive supplied population moment $\nu_{2,K} = E\{K^2(r_i)\}$.
trace_R2	Optional strictly positive supplied $\mathrm{tr}(R^2)$. It must be supplied together with nu2.
zero_tol	Non-negative threshold for a singular standardized radius.
keep_scores	Whether to retain directions, radii, weights, and weighted score means.

Details

This function deliberately says oracle. It does not estimate the reference location, diagonal, radial moment, or correlation trace. A valid upper-tail normal p-value is returned only when the caller supplies either null_sd, or both nu2 and trace_R2, in which case

$$\sigma_{n,K} = \left\{ \frac{2\nu_{2,K}^2 \,\mathrm{tr}(R^2)}{n(n-1)p^2} \right\}^{1/2}.$$

With no calibration inputs the raw score remains available and calibrated is FALSE; no feasible leave-out or plug-in estimator is guessed.

Value

A list of class oracle_weighted_sign_sum_test. The p.value field is absent unless explicit calibration is complete.

References

Feng, L., Liu, B. and Ma, Y. (2021). An inverse norm sign test for location parameters in high-dimensional data. Journal of Business & Economic Statistics 39, 807–815.

Examples

```
x <- matrix(c(-2, 1, 0, 3, -1, 2, 1, -3, 2, 0, 4, -2), ncol = 2)
oracle_weighted_sign_sum_test(
  x, theta = c(0, 0), diagonal = c(1, 1), K = "constant"
)
```

---

park_ayyala_one_sample_test

Park–Ayyala one-sample high-dimensional mean test

Description

Tests $H_0 : \mu = \mu_0$ with the diagonal leave-two-out statistic of Park and Ayyala (2013). For every ordered pair of observations, the diagonal covariance used in the quadratic product is estimated after removing that pair. The estimated null variance uses the matching leave-two-out means. At least six observations are required: below this threshold the inverse-chi-square bias correction is not valid (and at $n = 5$ the corrected numerator is identically zero). The algebraic API therefore permits $n = 6, 7$, but squared inverse leave-out variances have no finite Gaussian expectation at those sizes; their asymptotic normal calibration is especially fragile, so $n \geq 8$ is strongly preferred.

Usage

```
park_ayyala_one_sample_test(x, mu = NULL)
```

Arguments

`x`	A numeric matrix or data frame with observations in rows and at least six rows.
`mu`	A finite null-mean vector. `NULL` uses the zero vector.

Details

Each residual `x - mu` is formed in extended precision and divided by a common scale within its variable. This is algebraically neutral for the Park–Ayyala statistic and protects its leave-out moments from avoidable overflow, underflow, and cancellation under a large common translation. No zero variance is replaced, no ridge is added, and a non-positive estimated null variance is reported as an error.

Value

An object of class `c("hd_location_test", "htest")`. The `components` field contains the two ordered-pair sums, bias-corrected raw statistic, and estimated null variance. `finite.sample.correction` is $(n-5)/(n-3)$, whereas `bias.factor` is the full coefficient $(n-5)/\{n(n-1)(n-3)\}$ multiplying the ordered-pair sum.

References

Park, J. and Ayyala, D. N. (2013). A test for the mean vector in large dimension and small samples. *Journal of Statistical Planning and Inference*, 143, 929–943.

Examples

```
set.seed(23)
x <- matrix(rnorm(24), nrow = 8, ncol = 3)
park_ayyala_one_sample_test(x)
```

`pesaran_cd_test` *Pesaran CD test for cross-sectional independence*

Description

Given a T by N residual matrix, computes

$$CD = \sqrt{2T/\{N(N-1)\}} \sum_{i<j} \hat{\rho}_{ij}.$$

The standard-normal calibration is two-sided. Residuals are used exactly as supplied: include an intercept in the preceding unit regressions when centering is required by the model.

Usage

```
pesaran_cd_test(residuals, keep_correlations = FALSE)
```

Arguments

`residuals`	Numeric T by N matrix, with time in rows and panel units in columns.
`keep_correlations`	Whether to retain the N by N sample residual-correlation matrix.

Value

An object inheriting from `htest`.

References

Pesaran, M. H. (2004). General Diagnostic Tests for Cross Section Dependence in Panels. IZA Discussion Paper 1240. https://docs.iza.org/dp1240.pdf

Examples

```
e <- matrix(c(-2, 1, 0, 2, -1, 3, 1, -2, 2, 1, -3, 1), 4, 3)
pesaran_cd_test(e)
```

`pesaran_yamagata_alpha_test` *Pesaran–Yamagata large-N sum alpha test*

Description

Implements the feasible statistic of Pesaran and Yamagata. Let v=T-K-1 and t_i be the usual unrestricted OLS intercept t statistic. Residual correlations are retained only when $|\sqrt{v}\,\hat{\rho}_{ij}| > \Phi^{-1}\{1 - p_0/(2N^{\delta})\}$. If $\widetilde{\rho}^2$ is twice the sum of squared retained upper-triangular correlations divided by N(N-1), the primary statistic is

$$\frac{N^{-1/2}\sum_i\{t_i^2 - v/(v-2)\}}{[v/(v-2)]\{2(v-1)(1+(N-1)\widetilde{\rho}^2)/(v-4)\}^{1/2}}.$$

Usage

```
pesaran_yamagata_alpha_test(returns, factors = NULL, p0 = 0.1, delta = 1)
```

Arguments

`returns`	Finite observation-by-asset numeric matrix or data frame.
`factors`	NULL, a finite length-T numeric vector, or a finite observation-by-factor numeric matrix or data frame.
`p0`	Finite thresholding probability in (0,1), default 0.1.
`delta`	Finite positive threshold exponent, default 1.

Details

The generic dense statistic in the book draft is not this feasible finite-v test: it omits the t-square centering and the multiple-testing threshold estimator. This function follows the primary paper.

Value

An asymptotic upper-tail normal alpha-test object.

References

Pesaran, M. H. and Yamagata, T. (2023). Testing for alpha in linear factor pricing models with a large number of securities. Journal of Financial Econometrics. doi:10.1093/jjfinec/nbad002.

Examples

```
f <- cbind(seq(-1, 1, length.out = 12))
y <- outer(seq_len(12), 1:3, function(i, j) sin(i + j)) +
  f %*% matrix(1:3, nrow = 1)
pesaran_yamagata_alpha_test(y, f)
```

`pmd_sparse_cca` *Penalized-matrix-decomposition sparse CCA*

Description

Solves the PMD sparse CCA problem

$$\max_{u,v} u'Mv, \quad \|u\|_2, \|v\|_2 \le 1, \quad \|u\|_1 \le c_x, \quad \|v\|_1 \le c_y$$

with the cross-covariance operator applied matrix-free. Later pairs use `M <- M - d u v'`. Both actual l1 bounds must be supplied; no permutation tuning or default grid is generated.

Usage

```
pmd_sparse_cca(
  x,
  y,
  l1_x,
  l1_y,
  components = 1L,
  center = TRUE,
  scale = TRUE,
  covariance_divisor = c("n", "n-1"),
  initial_x = NULL,
  initial_y = NULL,
  solver_tol = 1e-08,
  solver_max_iter = 1000L,
  strict = TRUE,
  keep_operator = TRUE
)
```

Arguments

`x`, `y`	Numeric paired data matrices with observations in rows.
`l1_x`, `l1_y`	Required actual l1 bounds, scalar or one per component.
`components`	Number of canonical pairs.
`center`	Logical or supplied centers. A length-two logical vector may control x and y separately.
`scale`	Logical or supplied list `list(x=..., y=...)` of positive scales. Logical `TRUE` uses sample standard deviations.
`covariance_divisor`	Common cross-covariance divisor.
`initial_x`, `initial_y`	Optional deterministic coefficient starts; both must be supplied together.
`solver_tol`, `solver_max_iter`	Alternating-solver controls.

strict	If TRUE, failure is an error; otherwise return an invalid fit.
keep_operator	Retain the explicit cross-covariance matrix. The solver itself remains matrix-free.

Details

For the shared hd_pca_fit contract, loadings, scores, center, p, and variable.names refer to the x view. The y-view counterparts are returned separately.

Value

A pmd_sparse_cca_fit inheriting from hd_pca_fit.

References

Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10, 515–534.

Examples

```
x <- cbind(a = c(-2, -1, 1, 2), b = c(1, -1, -1, 1))
y <- cbind(c = x[, 1], d = c(-1, 1, 1, -1))
pmd_sparse_cca(x, y, l1_x = 1, l1_y = 1, scale = FALSE)
```

---

pmd_sparse_pca *Penalized-matrix-decomposition sparse PCA*

Description

For each residual data matrix R, solves the PMD rank-one problem

$$\max_{u,v} u'Rv, \quad \|u\|_2 \le 1, \quad \|v\|_2 \le 1, \quad \|v\|_1 \le c,$$

then applies R <- R - d u v'. The supplied l1_bound is the actual $c \in [1, \sqrt{p}]$, not a rescaled UI fraction. The alternating solver reports both block-KKT and fixed-point residuals.

Usage

```
pmd_sparse_pca(
  x,
  l1_bound,
  components = 1L,
  center = c("mean", "none"),
  scale = FALSE,
  covariance_divisor = c("n", "n-1"),
  initial = NULL,
  solver_tol = 1e-08,
```

```
  solver_max_iter = 1000L,
  symmetry_tol = sqrt(.Machine$double.eps),
  strict = TRUE,
  keep_operator = TRUE
)
```

Arguments

x	Numeric data with observations in rows.
l1_bound	Required PMD loading bound, scalar or one per component.
components	Number of components.
center	Numeric center or "mean"/"none".
scale	Logical or supplied positive scale vector.
covariance_divisor	Divisor used to report component variances.
initial	Optional deterministic loading starts.
solver_tol, solver_max_iter	Alternating-solver controls.
symmetry_tol	Positive covariance certification tolerance.
strict	If TRUE, failure is an error; otherwise return an invalid fit.
keep_operator	Retain the processed sample covariance.

Value

A pmd_sparse_pca_fit inheriting from hd_pca_fit.

References

Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10, 515–534.

Examples

```
x <- rbind(c(3, 0), c(-3, 0), c(0, 1), c(0, -1))
pmd_sparse_pca(x, l1_bound = 1)
```

`poet_covariance` *POET covariance estimator with a supplied factor count*

Description

Implements Principal Orthogonal complEment Thresholding (POET). The first `factors` sample principal components form the low-rank part; a generalized threshold is applied only to the off-diagonal entries of the orthogonal residual covariance, and the two parts are added back. The factor count is deliberately required: the cited display in the book does not specify a unique data-driven selector.

Usage

```
poet_covariance(
  x,
  factors,
  threshold,
  threshold_type = c("correlation", "variance"),
  rule = c("hard", "soft", "scad", "adaptive_lasso"),
  scad_a = 3.7,
  adaptive_eta = 1,
  center = TRUE
)
```

Arguments

`x`	Numeric matrix with observations in rows.
`factors`	A supplied non-negative integer number of factors.
`threshold`	Finite non-negative τ or C, according to `threshold_type`.
`threshold_type`	Either `"correlation"` or `"variance"`.
`rule`, `scad_a`, `adaptive_eta`	Generalized thresholding rule controls.
`center`	Whether to remove column means.

Details

`threshold_type = "correlation"` uses primary equation (2.6), $\lambda_{ij} = \tau\sqrt{r_{ii}r_{jj}}$. `"variance"` uses equation (3.2), $C\sqrt{\hat{\theta}_{ij}}\{p^{-1/2} + \sqrt{\log(p)/n}\}$. Residual diagonal entries are always retained. No positive-definiteness repair, ridge, or automatic tuning is applied.

Value

An `hd_covariance_estimator` object containing the low-rank part, raw and thresholded orthogonal complements, eigenvalues/eigenvectors, and thresholds.

References

Fan, J., Liao, Y. and Mincheva, M. (2013). *Journal of the Royal Statistical Society: Series B*, 75, 603–680. arXiv:1201.0175.

Examples

```
set.seed(34)
x <- matrix(rnorm(60), nrow = 15, ncol = 4)
poet_covariance(x, factors = 1, threshold = 0.2)
```

`poet_tme` *One-step Tyler POET estimator for an elliptical factor model*

Description

Implements the primary POET-TME construction. With a positive-definite preliminary inverse shape $\widehat{V}_S$, it first computes

$$\widehat{\Sigma}_T = \frac{p}{n} \sum_i \frac{(X_i - \widehat{\mu})(X_i - \widehat{\mu})^T}{(X_i - \widehat{\mu})^T \widehat{V}_S (X_i - \widehat{\mu})}$$

and then applies the same supplied-factor POET decomposition once. This is the one-step estimator recommended in the primary paper, not a new convergence iteration.

Usage

```
poet_tme(
  x,
  factors,
  threshold = NULL,
  constant = 1,
  rule = c("hard", "soft", "scad", "adaptive_lasso"),
  scad_a = 3.7,
  adaptive_eta = 1,
  preliminary_precision = NULL,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

`x`	Numeric matrix or data frame with observations in rows.
`factors`	Supplied non-negative integer number of factors.
`threshold`	Optional common threshold in trace-normalized scatter units. `NULL` uses the primary rate with `constant`.

constant	Non-negative multiplier for the primary rate threshold.
rule	One of "hard", "soft", "scad", or "adaptive_lasso".
scad_a, adaptive_eta	Generalized-threshold rule parameters.
preliminary_precision	Optional positive-definite preliminary inverse shape matrix or a valid elliptical_factor_precision_f
tol, max_iter, zero_tol	Spatial-median controls.
strict	If TRUE, fail when the spatial median is not certified; otherwise return an explicitly invalid diagnostic object.

Details

If preliminary_precision = NULL, a POET-SS fit with the same factor and threshold controls is constructed and inverted only when it is strictly positive definite. A supplied matrix, or the certified estimate from an elliptical_factor_precision_fit, may be used instead. No generalized inverse, ridge, or positive-definiteness repair is performed.

Value

An elliptical_factor_fit whose pilot is the one-step Tyler matrix.

References

Xu, X., Ma, H., Wang, H. and Feng, L. (2025). arXiv:2512.19325.

Examples

```
x <- rbind(c(-3, -2), c(-2, -1), c(-1, 1),
           c(1, -1), c(2, 1), c(3, 2))
poet_tme(x, factors = 0, threshold = 0.2,
         preliminary_precision = diag(2))
```

---

predict.hd_classifier_fit

Predict from a Chapter 5 classifier

Description

Predict from a Chapter 5 classifier

Usage

```
## S3 method for class 'hd_classifier_fit'
predict(object, newdata, type = c("class", "score"), ...)
```

Arguments

object	A valid object inheriting from hd_classifier_fit.
newdata	Numeric matrix or all-numeric data frame with observations in rows.
type	Return class labels or numerical scores.
...	Reserved for method compatibility; currently unused.

Value

For type = "score", a numeric vector. For type = "class", a factor with the training class levels.

---

predict.semc_fit *Predict from a semiparametric elliptical mixture fit*

Description

Predict from a semiparametric elliptical mixture fit

Usage

```
## S3 method for class 'semc_fit'
predict(object, newdata = NULL, type = c("class", "posterior"), ...)
```

Arguments

object	A valid object returned by semc_fit().
newdata	Optional numeric matrix. NULL returns fitted predictions.
type	Either "class" or "posterior".
...	Unused.

Value

Integer class labels or a posterior-probability matrix.

`regularized_spatial_sign_covariance` *Regularized spatial-sign covariance (RSSCM)*

Description

Forms $V = pS_{sgn}$, lets $a = \mathrm{tr}(V^2)/p$, and estimates the data weight by

$$\widehat{\alpha}_{raw} = \frac{\{n/(n-1)\}(a - p/n) - 1}{a - 1}.$$

The published estimator projects this value to `[0, 1]` and returns $\widehat{\alpha}V + (1 - \widehat{\alpha})I_p$. When `a` is exactly one, the formula is `0/0`; this implementation returns data weight zero and explicitly labels the signal-free case instead of producing `NaN`.

Usage

```
regularized_spatial_sign_covariance(
  x,
  center = c("spatial", "mean", "none"),
  tol = 1e-08,
  max_iter = 1000L,
  strict = TRUE,
  keep_signs = FALSE
)
```

Arguments

`x`	A finite numeric matrix or data frame, observations in rows.
`center`	One of `"spatial"`, `"mean"`, `"none"`, or a supplied finite numeric center. `"none"` treats the data as already centered.
`tol`, `max_iter`, `strict`	Spatial-median convergence controls.
`keep_signs`	Whether to retain the fitted sign matrix.

Details

Spatial-median centering is the paper-facing default. `center = "mean"`, `"none"`, or a numeric vector are practical alternatives and are labelled as such in diagnostics. Exact zero residuals are errors: the official code's observation deletion is not reproduced.

Value

An `hd_covariance_estimator` whose estimate has trace `p`, with SSCM, raw/projected data weight, shrinkage intensity, centre, and diagnostics.

References

Raninen, E. and Ollila, E. (2022). Bias adjusted sign covariance matrix. *IEEE Signal Processing Letters*, 29, 339–343. doi:10.1109/LSP.2021.3134940.

Examples

```
x <- matrix(c(-2, 0, -1, 2, 0, -2, 1, 1, 2, -1, 3, 2),
            ncol = 2, byrow = TRUE)
regularized_spatial_sign_covariance(x)
```

`relliptical` *Simulate an elliptically symmetric sample*

Description

Uses the stochastic representation $X = \mu + \xi AU$, where `shape` is AA^T, U is uniform on the unit sphere, and `radial` supplies the non-negative radii. The default chi radius yields a multivariate Gaussian sample with covariance `shape`.

Usage

```
relliptical(n, location = 0, shape = diag(length(location)), radial = NULL)
```

Arguments

`n`	Number of observations.
`location`	Location vector. A scalar zero is expanded when `shape` has dimension greater than one.
`shape`	Positive semidefinite scatter matrix.
`radial`	Either `NULL`, a function accepting `n`, or a non-negative numeric vector of length `n`.

Value

An `n` by `p` numeric matrix.

References

Fang, K.-T. and Anderson, T. W. (1990). *Statistical Inference in Elliptically Contoured and Related Distributions*. Allerton Press.

Examples

```
set.seed(6)
relliptical(5, location = c(1, -1), shape = diag(c(1, 4)))
```

`robust_factor_subspace` *Robust principal subspace for an elliptical factor model*

Description

Estimates a supplied-dimensional principal subspace from either the sample spatial-sign covariance matrix or the exact multivariate Kendall matrix. The result is deliberately called a `principal_subspace`: a finite-sample equality with the span of an unknown factor-loading matrix is not asserted.

Usage

```
robust_factor_subspace(
  x,
  factors,
  method = c("spatial_sign", "kendall"),
  center = c("spatial", "mean", "none"),
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  zero_action = c("error", "zero"),
  eigen_tol = sqrt(.Machine$double.eps)
)
```

Arguments

`x`	Numeric matrix or data frame with observations in rows.
`factors`	Supplied positive principal-subspace dimension.
`method`	Either `"spatial_sign"` or `"kendall"`.
`center`	`"spatial"`, `"mean"`, `"none"`, or a supplied finite center. Kendall's operator is translation invariant, but this center still defines the returned component scores.
`tol`, `max_iter`	Spatial-median iteration controls.
`zero_tol`	Non-negative zero-residual or tied-pair tolerance.
`zero_action`	Either error on zero directions or retain their explicit zero contribution under the primary denominator.
`eigen_tol`	Positive relative eigengap and PSD tolerance.

Details

The requested boundary must have a certified eigengap. Signs are anchored deterministically, while directions within a repeated eigenvalue block are represented by their projector. Zero spatial residuals or tied Kendall pairs are errors unless their zero contribution is explicitly requested.

Value

A robust_factor_subspace_fit and hd_factor_fit object containing the orthonormal principal-subspace basis, projector, robust operator, and centered component scores.

References

Han, F. and Liu, H. (2018). ECA: High-dimensional elliptical component analysis in non-Gaussian distributions. *JASA*, 113, 252–268.

Examples

```
x <- rbind(c(-3, -1), c(-2, 1), c(-1, -2),
           c(1, 2), c(2, -1), c(3, 1))
fit <- robust_factor_subspace(x, factors = 1, method = "kendall",
                              center = "mean")
fit$principal_subspace
```

---

robust_gqda *Generalized QDA from certified robust class fits*

Description

This adapter applies the audited GQDA decision and tuning rule to two certified robust location–scatter fits. It deliberately does not invent a new M-, MVE-, MCD-, S-, or SD-estimation algorithm.

Usage

```
robust_gqda(
  x,
  y,
  class_fits,
  c = NULL,
  selection = base::c("resubstitution", "fixed"),
  strict = TRUE
)
```

Arguments

x	Numeric training matrix with observations in rows.
y	Two-class response. Factor levels define class 1 then class 2; otherwise first appearance defines the order.
class_fits	Optional length-two list of certified fits, each providing finite location and strictly positive-definite scatter. A compatible precision may also be supplied and is checked against the scatter.
c	Optional fixed constant in [0,1].

`selection`	One of `"resubstitution"`, `"fixed"`, or `"cross_validation"`. Supplying `c` selects `"fixed"`. Cross-validation is available only for internally refitted classical moments.
`strict`	If `TRUE`, numerical/certificate failure is an error; otherwise an invalid, non-predictable fit is returned with a warning.

Value

An `hd_classifier_fit`.

References

Bose, S., Pal, A., SahaRay, R., and Nayak, J. (2015). Generalized quadratic discriminant analysis. *Pattern Recognition*, 48(8), 2676–2684. doi:10.1016/j.patcog.2015.02.016.

Examples

```
x <- rbind(c(-2, 0), c(-1, 1), c(-1, -1), c(-2, 1),
           c(2, 0), c(1, 1), c(1, -1), c(2, -1))
y <- factor(rep(c("left", "right"), each = 4))
fits <- lapply(split(seq_len(nrow(x)), y), function(ii) {
  list(valid = TRUE, location = colMeans(x[ii, , drop = FALSE]),
       scatter = stats::cov(x[ii, , drop = FALSE]))
})
robust_gqda(x, y, fits, c = 0)
```

`rothman_levina_zhu_covariance_threshold` *Rothman–Levina–Zhu generalized covariance thresholding*

Description

Applies a generalized thresholding map entrywise to the divisor-n centered sample covariance. The four rules in Rothman, Levina and Zhu (2009) are hard, soft, SCAD, and adaptive lasso. Their adaptive-lasso rule is

$$\operatorname{sign}(z)(|z| - \lambda^{\eta+1}|z|^{-\eta})_+.$$

The accompanying book calls the fourth rule MCP; that attribution is incorrect, so this API does not silently rename adaptive lasso as MCP.

Usage

```
rothman_levina_zhu_covariance_threshold(
  x,
  threshold = NULL,
  constant = 1,
  rule = c("hard", "soft", "scad", "adaptive_lasso"),
```

```
  scad_a = 3.7,
  adaptive_eta = 1,
  center = TRUE,
  divisor = c("n", "n-1"),
  threshold_diagonal = TRUE
)
```

Arguments

x	Numeric matrix with observations in rows.
threshold	A finite non-negative threshold in covariance units. NULL uses the rate threshold controlled by constant.
constant	A finite non-negative multiplier for the rate threshold.
rule	One of "hard", "soft", "scad", or "adaptive_lasso".
scad_a	SCAD shape parameter, greater than two.
adaptive_eta	Non-negative adaptive-lasso exponent.
center	Whether to remove column means.
divisor	Either "n" (primary-paper default) or "n-1".
threshold_diagonal	Whether to apply the rule to diagonal entries.

Value

An hd_covariance_estimator object.

References

Rothman, A. J., Levina, E. and Zhu, J. (2009). *Journal of the American Statistical Association*, 104, 177–186. doi:10.1198/jasa.2009.0101.

Examples

```
set.seed(32)
x <- matrix(rnorm(40), nrow = 10, ncol = 4)
rothman_levina_zhu_covariance_threshold(
  x, threshold = 0.2, rule = "soft"
)
```

rspherical *Simulate a spherically symmetric sample*

Description

Simulate a spherically symmetric sample

Usage

```
rspherical(n, p, location = rep.int(0, p), radial = NULL)
```

Arguments

n	Number of observations.
p	Dimension.
location	Location vector.
radial	Radial specification passed to relliptical().

Value

An n by p numeric matrix.

References

Fang, K.-T. and Anderson, T. W. (1990). *Statistical Inference in Elliptically Contoured and Related Distributions*. Allerton Press.

Examples

```
set.seed(7)
rspherical(4, p = 3, radial = rep(1, 4))
```

rts_factor *Robust two-step factor estimator based on multivariate Kendall's tau*

Description

Implements the primary RTS estimator. If Gamma contains the supplied top Kendall eigenvectors, loadings are sqrt(p) * Gamma, factor scores are the cross-sectional OLS estimates X_centered %*% loadings / p, and the common component is factor_scores %*% t(loadings). Centering affects the latter two quantities and is therefore explicit even though the Kendall operator is translation invariant.

Usage

```
rts_factor(
  x,
  factors,
  center = c("mean", "none"),
  zero_tol = 0,
  zero_pair_action = c("error", "zero"),
  eigen_tol = sqrt(.Machine$double.eps)
)
```

Arguments

x	Numeric matrix or data frame with time/observations in rows and cross-sectional variables in columns.
factors	Supplied positive factor number.
center	"mean", "none", or a supplied finite center vector.
zero_tol	Non-negative tolerance for tied pairwise differences.
zero_pair_action	Either error on ties or retain their explicit zero contribution under the all-pairs Kendall denominator.
eigen_tol	Positive relative PSD, numerical-rank, and eigengap tolerance.

Value

An rts_factor_fit and hd_factor_fit object with loading matrix, OLS factor scores, common component, residuals, fitted data, principal subspace, and complete Kendall diagnostics.

References

He, Y., Kong, X., Yu, L. and Zhang, X. (2022). Large-dimensional factor analysis without moment constraints. *Journal of Business & Economic Statistics*, 40, 302–312.

Examples

```
x <- rbind(c(-3, -1, 0), c(-2, 1, 1), c(-1, -2, 0),
           c(1, 2, 0), c(2, -1, -1), c(3, 1, 0))
fit <- rts_factor(x, factors = 1)
crossprod(fit$loadings) / ncol(x)
```

`scaled_spatial_median` *Scaled spatial median with diagonal HR standardisation*

Description

Fits the full-sample diagonal Hettmansperger–Randles system used by Liu, Feng, Zhao and Wang:

$$n^{-1}\sum_i U\{D^{-1/2}(X_i-\theta)\}=0,$$

$$(p/n)\,\mathrm{diag}\sum_i U\{D^{-1/2}(X_i-\theta)\}U\{D^{-1/2}(X_i-\theta)\}^{\mathsf{T}}=I_p.$$

Starting with the sample mean and marginal sample variances, the function applies the paper's simultaneous location and diagonal-scale recursion.

Usage

```
scaled_spatial_median(
  x,
  tol = 1e-07,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

`x`	A numeric matrix or data frame with observations in rows and variables in columns. At least two observations and positive marginal sample variation are required.
`tol`	A finite positive estimating-equation tolerance.
`max_iter`	A positive integer maximum number of recursion updates.
`zero_tol`	A non-negative threshold for declaring an internally standardised residual radius singular. The literal default is zero.
`strict`	Whether non-convergence is an error (`TRUE`) or warning with the last finite iterate returned (`FALSE`).

Details

The equations identify `D` only up to a common positive multiplier. Internally, `D` has unit geometric mean after safe coordinate standardisation; the returned input-coordinate diagonal is divided by its largest entry. Both identifications leave the location and signs unchanged. Radial quantities are explicitly labelled as belonging to the internal unit-geometric-mean identification. The original paper notes that general existence, uniqueness, and convergence of this recursion are not proved.

With `strict = TRUE`, failure of the relative location/log-diagonal update to reach `tol` within `max_iter` updates is an error. With `strict = FALSE`, the last finite iterate is returned with a warning and

`iteration.stable = FALSE`. Both equation residuals are reported separately and are not mislabelled as the paper's unspecified stopping rule. Coincident residuals, non-positive marginal variation, and non-finite updates are errors; no ridge, floor, perturbation, or pseudoinverse is used.

Value

A list containing `location`, the identified scale diagonal and its logarithm, spatial directions, radial quantities, and convergence and equation diagnostics.

References

Liu, B., Feng, L., Zhao, P. and Wang, Z. Spatial-sign based maxsum test for high-dimensional location parameters. *Statistica Sinica*, accepted. doi:10.5705/ss.202024.0051.

Examples

```
x <- matrix(c(0.7, -0.4, 1.1, -1.2, 0.2, 1.5, -0.8,
              -1.1, 0.8, 0.3, -0.5, 1.4, -0.2, 0.6), 7, 2)
scaled_spatial_median(x, tol = 1e-6)
```

---

sdar_qda *Sparse discriminant analysis with quadratic interactions*

Description

Fits the SDAR rule through two Dantzig programs. The interaction operator is applied directly as one half of $S_1 D S_2 + S_2 D S_1$; no Kronecker matrix is formed. The interaction is symmetrized and its feasibility is then checked again. The method has an equal-prior contract. The signed determinant of $I + D S_1$ must be positive.

Usage

```
sdar_qda(
  x,
  y,
  lambda_D = NULL,
  lambda_beta = NULL,
  parameter_grid = NULL,
  folds = 5L,
  solver_tol = 1e-07,
  feasibility_tol = 1e-07,
  solver_max_iter = 10000L,
  strict = TRUE
)
```

Arguments

x	Numeric training matrix with observations in rows.
y	Two-class response. Its first observed level is class 1.
lambda_D, lambda_beta	Explicit non-negative Dantzig bounds.
parameter_grid	Paired lambda_D and lambda_beta columns for deterministic stratified joint cross-validation.
folds	Number of folds used only with a parameter grid.
solver_tol	Positive optimization and certificate tolerance.
feasibility_tol	Positive tolerance for final constraint feasibility.
solver_max_iter	Positive optimization iteration limit.
strict	Whether numerical failure is an error; otherwise an invalid hd_classifier_fit is returned with a warning.

Value

An hd_classifier_fit; non-negative scores select class 1.

References

Cai, T. T. and Zhang, L. (2021). A convex optimization approach to high-dimensional sparse quadratic discriminant analysis. *Annals of Statistics*, 49, 1537-1568.

Examples

```
x <- rbind(
  c(-2, 0), c(-1, 1), c(-1, -1), c(-2, 1),
  c(2, 0), c(1, 2), c(1, -2), c(2, 1)
)
y <- factor(rep(c("left", "right"), each = 4))
fit <- sdar_qda(x, y, lambda_D = 0.5, lambda_beta = 0.5,
                solver_tol = 1e-6)
```

---

semc_fit *Semiparametric elliptical mixture clustering*

Description

Fits the Feng–Zhuang semiparametric generalized-EM model with common radial generator and common trace-normalized shape. The nested shape pipelines are "tyler" (sign pilot, POET initialization, weighted Tyler), "poet" (the preceding pipeline plus a second POET step), and "glasso" (the full paper pipeline plus off-diagonal graphical lasso).

Usage

```
semc_fit(
  x,
  K,
  shape = c("glasso", "poet", "tyler"),
  initialization = c("maxmin", "software_random", "labels", "centers"),
  initial_labels = NULL,
  initial_centers = NULL,
  first_index = 1L,
  init_tau = 0,
  init_tau_grid = NULL,
  init_B = 5L,
  init_nstart = 1L,
  init_max_iter = 50L,
  init_empty_action = c("error", "farthest"),
  init_empty_active = c("error", "all"),
  init_dispersion_floor = 1e-08,
  outer_nstart = 1L,
  eta_mu = 0.7,
  eta_precision = 0.7,
  mixing_floor = 1e-10,
  center_weight_floor = 1e-10,
  max_iter = 25L,
  convergence_tol = 0.002,
  bandwidth = NULL,
  bandwidth_min = 0.05,
  generator_grid_size = 250L,
  generator_radius_floor = 1e-04,
  generator_density_floor = 1e-10,
  generator_score_clip = c(0.001, 50),
  generator_spline_spar = 0.55,
  generator_extrapolation = c("constant", "error"),
  radial_floor = 1e-06,
  poet_factor_selection = c("supplied", "software_gr"),
  poet_factors = 0L,
  poet_max_factors = 8L,
  factor_ratio_floor = 1e-08,
  poet_threshold = NULL,
  poet_threshold_scale = 0.55,
  poet_ridge = 0,
  tyler_ridge = 0.02,
  tyler_tol = 1e-05,
  tyler_max_iter = 200L,
  glasso_lambda = NULL,
  glasso_lambda_grid = NULL,
  glasso_lambda_scale = 0.28,
  glasso_ebic_gamma = 0.5,
  glasso_tol = 1e-06,
```

```
  glasso_max_iter = 10000L,
  glasso_initial_step = 1,
  glasso_max_backtracking = 100L,
  glasso_majorization_tol = 1e-12,
  spd_tol = 0,
  symmetry_tol = 1e-10,
  seed = NULL,
  strict = TRUE,
  keep_path = TRUE
)
```

Arguments

x	Numeric matrix with observations in rows.
K	Integer number of mixture components, at least two and less than the number of observations.
shape	One of "glasso", "poet", or "tyler".
initialization	Initial sparse-K-median start: deterministic "maxmin", official-software-style "software_random", or supplied "labels" or "centers".
initial_labels, initial_centers	Supplied initialization used by the corresponding initialization choice.
first_index	First row for deterministic max-min initialization.
init_tau	Nonnegative sparse-K-median feature threshold. NULL activates the official-software quantile-grid/permutation selector.
init_tau_grid	Optional explicit grid when init_tau is NULL.
init_B	Number of initialization reference permutations.
init_nstart, init_max_iter	Sparse-K-median start and iteration limits.
init_empty_action	Explicit empty-cluster policy.
init_empty_active	Explicit all-features fallback policy when a threshold excludes every feature.
init_dispersion_floor	Positive floor used only to reject degenerate initialization Gap logarithms; no objective is silently altered.
outer_nstart	Number of full GEM starts.
eta_mu, eta_precision	Damping constants in (0,1].
mixing_floor	Explicit positive mixing-weight floor.
center_weight_floor	Explicit positive denominator boundary for center updates.
max_iter, convergence_tol	Outer GEM controls.
bandwidth	Optional transformed-radius KDE bandwidth.

`bandwidth_min` Explicit lower bound for an automatic or supplied bandwidth.

`generator_grid_size`
: Number of transformed-radius grid points.

`generator_radius_floor`, `generator_density_floor`
: Explicit positive generator reconstruction floors.

`generator_score_clip`
: Two positive radial-score clipping endpoints.

`generator_spline_spar`
: Fixed smoothing-spline parameter.

`generator_extrapolation`
: Either constant endpoint extrapolation (official software contract) or strict error.

`radial_floor` Positive denominator floor in sign and Tyler maps.

`poet_factor_selection`
: Supplied factor count or the official-software GR selector.

`poet_factors`, `poet_max_factors`
: Factor controls.

`factor_ratio_floor`
: Explicit GR ratio floor.

`poet_threshold` Optional POET threshold. NULL uses poet_threshold_scale * sqrt(log(p) / n_eff).

`poet_threshold_scale`
: Explicit software-rate constant.

`poet_ridge` Explicit nonnegative POET diagonal ridge; zero means none.

`tyler_ridge` Explicit ridge in the paper's ridge-stabilized Tyler map.

`tyler_tol`, `tyler_max_iter`
: Tyler certificate controls.

`glasso_lambda` Optional singleton off-diagonal graphical-lasso penalty.

`glasso_lambda_grid`
: Optional explicit EBIC grid.

`glasso_lambda_scale`
: Software-rate constant when no penalty is supplied.

`glasso_ebic_gamma`
: EBIC gamma for a multi-penalty grid.

`glasso_tol`, `glasso_max_iter`, `glasso_initial_step`
: Graphical-lasso convergence and initial-step controls.

`glasso_max_backtracking`
: Maximum number of graphical-lasso SPD backtracking steps per iteration.

`glasso_majorization_tol`
: Nonnegative tolerance for the graphical-lasso majorization/KKT certificate.

`spd_tol`, `symmetry_tol`
: Strict shape/precision certificate tolerances.

`seed` Optional local seed. It is required for stochastic starts or automatic initialization-threshold selection and is restored on exit.

`strict` If TRUE, fail when the selected outer fit does not converge.

`keep_path` Retain the deterministic outer-iteration audit path.

Details

The paper does not uniquely determine initialization, bandwidth constants, POET factor selection, penalty constants, generator clipping, or all stopping rules. Defaults identified below as software contracts reproduce the named choices at the pinned official implementation, while every numerical floor, ridge, damping constant, and fallback policy is an explicit argument. There is no positive-definite projection, pseudoinverse, hidden ridge, or graphical-lasso fallback.

Value

A semc_fit object containing labels, posterior probabilities, mixing weights, centers, trace-p shape, precision, generator grid, initialization and shape certificates, explicit controls, and provenance.

References

Feng, L. and Zhuang, D. (2026). Semiparametric Elliptical Mixture Clustering for High-Dimensional Data. arXiv:2605.08995. https://arxiv.org/abs/2605.08995.

Examples

```
semc_x <- matrix(c(
  -3, -2, -2, -3, -2, -1, -1, -2, -2.5, -2, -1.5, -2.2,
   3,  2,  2,  3,  2,  1,  1,  2,  2.5,  2,  1.5,  2.2
), ncol = 2L, byrow = TRUE)
semc_labels <- rep(1:2, each = 6L)
semc_model <- semc_fit(
  semc_x, K = 2L, shape = "tyler",
  initialization = "labels", initial_labels = semc_labels,
  first_index = 1L, init_tau = 0, init_tau_grid = NULL, init_B = 2L,
  init_nstart = 1L, init_max_iter = 20L,
  init_empty_action = "error", init_empty_active = "error",
  init_dispersion_floor = 1e-8, outer_nstart = 1L,
  eta_mu = 0.7, eta_precision = 0.7,
  mixing_floor = 1e-10, center_weight_floor = 1e-10,
  max_iter = 10L, convergence_tol = 0.2,
  bandwidth = 0.2, bandwidth_min = 0.05,
  generator_grid_size = 40L, generator_radius_floor = 1e-4,
  generator_density_floor = 1e-10,
  generator_score_clip = c(1e-3, 50),
  generator_spline_spar = 0.55, generator_extrapolation = "constant",
  radial_floor = 1e-6, poet_factor_selection = "supplied",
  poet_factors = 0L, poet_max_factors = 0L,
  factor_ratio_floor = 1e-8, poet_threshold = 0.1,
  poet_threshold_scale = 0.55, poet_ridge = 0.05,
  tyler_ridge = 0.1, tyler_tol = 1e-5, tyler_max_iter = 200L,
  spd_tol = 0, symmetry_tol = 1e-10, seed = 11L,
  strict = TRUE, keep_path = FALSE
)
predict(semc_model)
```

`semc_select_k_gap` *Select the SEMC cluster count by Gap-LSE*

Description

The default follows equations (2.18)–(2.21) of Feng and Zhuang: hard fitted labels, radial dispersion mean(log(1 + delta)), independently column-permuted reference samples, and the lower-complexity one-standard- error rule. The pinned official software instead defaults to posterior-weighted sum(tau * delta); that distinct contract is available only through the explicit dispersion = "software_soft_delta" choice.

Usage

```
semc_select_k_gap(
  x,
  k_grid = 2:5,
  B = 20L,
  control = list(),
  dispersion = c("paper_hard_log1p", "software_soft_delta"),
  rule = c("lse", "max"),
  permutation_indices = NULL,
  seed,
  dispersion_floor = 1e-12,
  keep_reference_fits = FALSE,
  keep_permutations = FALSE
)
```

Arguments

`x`	Numeric matrix with observations in rows.
`k_grid`	Sorted candidate component counts.
`B`	At least two reference permutations.
`control`	Named arguments passed to semc_fit(); x, K, and seed are controlled by this selector.
`dispersion`	Paper hard-log1p or official-software soft-delta contract.
`rule`	Lower-complexity one-standard-error or maximum-Gap selection.
`permutation_indices`	Optional integer n by p by B permutation array.
`seed`	Explicit local seed for permutation and fit seeds.
`dispersion_floor`	Positive rejection boundary for logarithms.
`keep_reference_fits`	Retain every reference fit.
`keep_permutations`	Retain the permutation array.

Value

A semc_gap_selection object with both LSE and maximum-Gap choices, observed fits, all formula ingredients, local seeds, and provenance.

References

Feng, L. and Zhuang, D. (2026). Semiparametric Elliptical Mixture Clustering for High-Dimensional Data. arXiv:2605.08995. https://arxiv.org/abs/2605.08995.

Examples

```
semc_x <- matrix(c(
  -3, -2, -2, -3, -2, -1, -1, -2, -2.5, -2, -1.5, -2.2,
   3,  2,  2,  3,  2,  1,  1,  2,  2.5,  2,  1.5,  2.2
), ncol = 2L, byrow = TRUE)
semc_labels <- rep(1:2, each = 6L)
semc_B <- 2L
semc_plan <- array(
  rep(seq_len(nrow(semc_x)), ncol(semc_x) * semc_B),
  dim = c(nrow(semc_x), ncol(semc_x), semc_B)
)
semc_control <- list(
  shape = "tyler", initialization = "labels",
  initial_labels = semc_labels, first_index = 1L,
  init_tau = 0, init_tau_grid = NULL, init_B = 2L,
  init_nstart = 1L, init_max_iter = 20L,
  init_empty_action = "error", init_empty_active = "error",
  init_dispersion_floor = 1e-8, outer_nstart = 1L,
  eta_mu = 0.7, eta_precision = 0.7,
  mixing_floor = 1e-10, center_weight_floor = 1e-10,
  max_iter = 10L, convergence_tol = 0.2,
  bandwidth = 0.2, bandwidth_min = 0.05,
  generator_grid_size = 40L, generator_radius_floor = 1e-4,
  generator_density_floor = 1e-10,
  generator_score_clip = c(1e-3, 50),
  generator_spline_spar = 0.55, generator_extrapolation = "constant",
  radial_floor = 1e-6, poet_factor_selection = "supplied",
  poet_factors = 0L, poet_max_factors = 0L,
  factor_ratio_floor = 1e-8, poet_threshold = 0.1,
  poet_threshold_scale = 0.55, poet_ridge = 0.05,
  tyler_ridge = 0.1, tyler_tol = 1e-5, tyler_max_iter = 200L,
  spd_tol = 0, symmetry_tol = 1e-10,
  strict = TRUE, keep_path = FALSE
)
semc_gap <- semc_select_k_gap(
  semc_x, k_grid = 2L, B = semc_B, control = semc_control,
  dispersion = "paper_hard_log1p", rule = "lse",
  permutation_indices = semc_plan, seed = 19L,
  dispersion_floor = 1e-12, keep_reference_fits = FALSE,
  keep_permutations = FALSE
)
semc_gap$selected.k
```

shao_threshold_lda *Shao–Wang–Deng–Wang thresholded sparse LDA*

Description

Implements the primary hard-threshold LDA benchmark. The pooled covariance uses divisor n; only its off-diagonal entries are retained when $|s_{ij}| > M_{cov}\sqrt{\log(p)/n}$, while the diagonal is always retained. Mean differences are retained when $|\bar{X}_j - \bar{Y}_j| > M_{mean}\{\log(p)/n\}^{\alpha}$, with $0 < \alpha < 1/2$. Equality is thresholded out in both cases.

Usage

```
shao_threshold_lda(
  x,
  y,
  M_cov = NULL,
  M_mean = NULL,
  alpha,
  parameter_grid = NULL,
  cv = "loo",
  prior = "equal",
  strict = TRUE
)
```

Arguments

x	Numeric matrix or all-numeric data frame, observations in rows.
y	Two-class label vector or factor.
M_cov, M_mean	Explicit non-negative threshold constants.
alpha	Exponent strictly between zero and one-half.
parameter_grid	Optional data frame or numeric matrix with columns M_cov and M_mean. No default grid is invented.
cv	The primary leave-one-out scheme; currently only "loo".
prior	"equal", "empirical", or a positive numeric vector of length two. Named entries must match the class levels.
strict	If TRUE, a method failure is an error. If FALSE, it is a warning followed by an invalid hd_classifier_fit.

Details

Supply either both constants or an explicit two-column grid. A grid is selected by exact leave-one-out correct classification count, recomputing all moments and thresholds in every fold. Candidates whose thresholded covariance is not positive definite in any fold are ineligible. A score tie selects class 1; a tuning tie selects the first supplied grid row.

Value

An hd_classifier_fit.

References

Shao, J., Wang, Y., Deng, X., and Wang, S. (2011). Sparse linear discriminant analysis by thresholding for high dimensional data. *Annals of Statistics*, 39, 1241–1265.

Examples

```
x <- rbind(c(3, 1), c(2, 2), c(4, -0.5),
           c(-3, -1), c(-2, -2), c(-4, 0.5))
shao_threshold_lda(
  x, rep(c("A", "B"), each = 3), M_cov = 0, M_mean = 0,
  alpha = 0.25
)
```

---

sign_whitened_sparse_cca

Book sign-whitened sparse CCA variant

Description

Implements the constrained program printed in Chapter 6 of the book. It first whitens the two diagonal blocks of the joint sample spatial-sign covariance and then solves

$$\max_{u,v} u^T K_S v, \quad \|u\|_2 \le 1,\ \|v\|_2 \le 1,\ \|u\|_1 \le c_x,\ \|v\|_1 \le c_y.$$

Usage

```
sign_whitened_sparse_cca(
  x,
  y,
  c_x,
  c_y,
  ridge_x = 0,
  ridge_y = 0,
  tol = 1e-07,
  max_iter = 500L,
  median_tol = 1e-08,
  median_max_iter = 500L,
  zero_tol = 0,
  zero_action = c("zero", "error"),
  rank_tol = sqrt(.Machine$double.eps),
  strict = TRUE
)
```

Arguments

x, y	Paired finite numeric matrices, observations in rows.
c_x, c_y	Explicit l1 bounds in [1, sqrt(ncol(block))].
ridge_x, ridge_y	Explicit non-negative diagonal additions used only in the two whitening metrics.
tol, max_iter	Alternating PMD update controls.
median_tol, median_max_iter	Joint spatial-median controls.
zero_tol, zero_action	Spatial-sign zero convention; see sscca().
rank_tol	Positive relative SPD tolerance for whitening. No pseudoinverse or eigenvalue floor is used.
strict	If TRUE, method failures are errors; otherwise an invalid fit is returned without exposing an uncertified estimate.

Details

This is intentionally a separately named book variant: it is not the penalized metric program proposed in Qian, Liu, and Feng (2025), which is implemented by sscca(). Any ridge used to make a whitening block SPD is explicit and changes the reported operator; the defaults apply no ridge.

Value

A sign_whitened_sparse_cca_fit and hd_cca_fit object containing whitened and original-coordinate directions, scores, constraints and convergence diagnostics.

References

Feng, L. (2026). *High-Dimensional Data Analysis for Elliptical Symmetric Distributions*, Chapter 6 (book manuscript).

For the distinct primary metric-penalized method, see Qian, J., Liu, W., and Feng, L. (2025). High dimensional sparse canonical correlation analysis for elliptical symmetric distributions. https://arxiv.org/abs/2504.13018.

Examples

```
t <- seq_len(12)
x <- cbind(x1 = sin(t), x2 = cos(t / 2))
y <- cbind(y1 = sin(t) + 0.2 * cos(t), y2 = cos(t / 2) - 0.1 * sin(t))
fit <- sign_whitened_sparse_cca(x, y, c_x = sqrt(2), c_y = sqrt(2))
fit$canonical.correlations
```

sm_sscm *Spatial-median clustering with a common SSCM metric*

Description

Uses full-dimensional spatial medians and the explicitly regularized common spatial-sign covariance matrix `crossprod(U) / n + lambda * I`. The inverse is an exact SPD inverse; no pseudoinverse or eigenvalue repair is used.

Usage

```
sm_sscm(
  x,
  K,
  lambda,
  init = "maxmin",
  first_index = 1L,
  tol = 1e-08,
  max_iter = 100L,
  spatial_max_iter = 500L,
  zero_tol = 0,
  empty_action = c("error", "farthest"),
  cycle_action = c("error", "return"),
  ties = "first",
  keep_distances = FALSE,
  keep_signs = FALSE
)
```

Arguments

`x`	Numeric observation-by-variable matrix or data frame.
`K`	Number of clusters.
`lambda`	Strictly positive ridge appearing in the stated SSCM method.
`init`	Either `"maxmin"`, `K` distinct row indices, or a finite `K` by `p` center matrix.
`first_index`	First max-min seed when `init = "maxmin"`.
`tol`	Positive spatial-median equation tolerance.
`max_iter`	Positive outer iteration limit.
`spatial_max_iter`	Positive modified-Weiszfeld iteration limit.
`zero_tol`	Non-negative zero-residual tolerance.
`empty_action`	`"error"` or the explicit deterministic `"farthest"` repair. The latter only donates from a cluster of size at least two.
`cycle_action`	Whether a detected repeated state is an `"error"` or is returned with a failed convergence certificate.

ties	Deterministic tie rule; currently only "first" is supported.
keep_distances	Whether to retain the final full distance matrix.
keep_signs	Whether to retain final residual spatial signs.

Value

An sm_sscm_fit object. Its diagnostics explicitly do not claim monotonicity of a global objective.

References

Zhao, P., Zhuang, D., and Feng, L. (2026). Sparse K-spatial-median clustering for high-dimensional data. arXiv:2605.00598. https://arxiv.org/abs/2605.00598.

Examples

```
x <- rbind(c(-2, 0), c(-1, 0), c(1, 0), c(2, 0))
sm_sscm(x, K = 2, lambda = 0.1, init = c(1, 4))
```

---

sparse_k_spatial_median

Sparse K-spatial-median clustering

Description

Updates full-dimensional spatial medians, hard-screens coordinates by across-center separation, and assigns observations in the retained subspace. This procedure is a deterministic block-coordinate heuristic and is not represented as optimizing a single global objective.

Usage

```
sparse_k_spatial_median(
  x,
  K,
  tau,
  init = "maxmin",
  first_index = 1L,
  tol = 1e-08,
  max_iter = 100L,
  spatial_max_iter = 500L,
  zero_tol = 0,
  empty_action = c("error", "farthest"),
  empty_active = c("error", "largest", "all"),
  cycle_action = c("error", "return"),
  ties = "first",
  reset_excluded = c("none", "overall_spatial_median"),
  keep_distances = FALSE
)
```

Arguments

x	Numeric observation-by-variable matrix or data frame.
K	Number of clusters.
tau	Finite non-negative hard-screening threshold. Equality is retained: a coordinate is active when its score is at least tau.
init	Either "maxmin", K distinct row indices, or a finite K by p center matrix.
first_index	First max-min seed when init = "maxmin".
tol	Positive spatial-median equation tolerance.
max_iter	Positive outer iteration limit.
spatial_max_iter	Positive modified-Weiszfeld iteration limit.
zero_tol	Non-negative zero-residual tolerance.
empty_action	"error" or the explicit deterministic "farthest" repair. The latter only donates from a cluster of size at least two.
empty_active	Action when no score reaches tau: "error", the first "largest" score, or "all" coordinates.
cycle_action	Whether a detected repeated state is an "error" or is returned with a failed convergence certificate.
ties	Deterministic tie rule; currently only "first" is supported.
reset_excluded	Output-only treatment of excluded center coordinates. "overall_spatial_median" never feeds reset values back into iteration.
keep_distances	Whether to retain the final full distance matrix.

Value

A sparse_k_spatial_median_fit object with both reported centers and untouched algorithm_centers, the active set, scores, and diagnostics.

References

Zhao, P., Zhuang, D., and Feng, L. (2026). Sparse K-spatial-median clustering for high-dimensional data. arXiv:2605.00598. https://arxiv.org/abs/2605.00598.

Examples

```
x <- rbind(c(-2, 0), c(-1, 0), c(1, 0), c(2, 0))
sparse_k_spatial_median(x, K = 2, tau = 0.5, init = c(1, 4))
```

`sparse_kmeans` *Sparse K-means with ordered-pair BCSS weights*

Description

Optimize the Witten–Tibshirani sparse K-means criterion from Chapter 7. For the returned partition, `feature_scores[j]` is exactly the displayed ordered-pair score `sum_ii' (x_ij-x_i'j)^2/n - sum_k sum_(i,i' in Ck) (` hence it equals twice `TSS[j] - WCSS[j]`. The half-scaled values are returned separately in diagnostics.

Usage

```
sparse_kmeans(
  x,
  clusters,
  s,
  initial = NULL,
  initialization = c("maxmin", "random"),
  first_index = 1L,
  seed = NULL,
  ties = "first",
  empty_action = c("error", "farthest"),
  weight_tol = 1e-10,
  weight_max_iter = 200L,
  solver_tol = 1e-10,
  solver_max_iter = 100L,
  strict = TRUE,
  keep_distances = FALSE
)
```

Arguments

`x`	Numeric observation matrix, with observations in rows.
`clusters`	Number of clusters.
`s`	Required L1 weight bound in `[1, sqrt(p)]`.
`initial`	Optional center matrix or vector of distinct row indices.
`initialization`	Max-min or seeded random initialization when `initial` is `NULL`.
`first_index`	First max-min seed row.
`seed`	Explicit seed required for random initialization; otherwise `NULL`.
`ties`	Exact tie rule; only deterministic `"first"` is implemented.
`empty_action`	Error, or the explicit farthest-observation repair.
`weight_tol`	Weight-threshold root and KKT tolerance.
`weight_max_iter`	Maximum bisection iterations for a weight update.

solver_tol	Tolerance used only to certify objective monotonicity.
solver_max_iter	Maximum Lloyd iterations.
strict	Error rather than return an uncertified iterate.
keep_distances	Retain the final squared-distance matrix.

Details

The weight block is the normalized positive soft-threshold of these scores under `norm(w, 2) <= 1`, `sum(w) <= s`, and `w >= 0`. At `s = 1`, a tied maximum score selects the smallest feature index. All-zero scores error rather than create uniform weights. Weighted assignment uses squared distance after multiplying each column by `sqrt(w)`.

Value

A `sparse_kmeans_fit` object with feature scores, weights, labels, centers, weighted objective, and solver certificates.

References

Witten, D. M. and Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105, 713–726. doi:10.1198/jasa.2010.tm09415.

Examples

```
x <- rbind(c(-3, 0), c(-2, 0), c(2, 0), c(3, 0))
sparse_kmeans(x, clusters = 2, s = 1, initial = c(1, 3))
```

---

sparse_kmeans_select_s

Select the sparse K-means L1 bound by permutation Gap calibration

---

Description

For every supplied `s_grid` value this function fits sparse K-means and records its optimized weighted BCSS `O(s)`. Each reference data set independently permutes the rows of every feature, preserving marginal scales and tails while breaking joint clustering. The reported criterion is `Gap(s) = log(O_obs(s)) - mean_b(log(O_perm,b(s)))`; every objective must be finite and strictly positive.

Usage

```
sparse_kmeans_select_s(
  x,
  clusters,
  s_grid,
  permutation_indices = NULL,
```

```
  n_permutations = NULL,
  permutation_seed = NULL,
  selection = c("max", "one_se"),
  initial = NULL,
  initialization = c("maxmin", "random"),
  first_index = 1L,
  clustering_seed = NULL,
  ties = "first",
  empty_action = c("error", "farthest"),
  weight_tol = 1e-10,
  weight_max_iter = 200L,
  solver_tol = 1e-10,
  solver_max_iter = 100L,
  strict = TRUE,
  keep_permutations = FALSE
)
```

Arguments

`x`	Numeric observation matrix, with observations in rows.
`clusters`	Number of clusters.
`s_grid`	Finite distinct candidate bounds in `[1, sqrt(p)]`.
`permutation_indices`	Optional explicit `n` by `p` by `B` array; each feature column in each slice must be a permutation of `1:n`.
`n_permutations`	Number of generated references when indices are not supplied.
`permutation_seed`	Explicit seed for generated permutation indices.
`selection`	Maximum-Gap or the stated one-SE rule.
`initial`	Optional center matrix or vector of distinct row indices.
`initialization`	Max-min or seeded random initialization when `initial` is `NULL`.
`first_index`	First max-min seed row.
`clustering_seed`	Seed passed only to random clustering initialization, separate from `permutation_seed`.
`ties`	Exact tie rule; only deterministic `"first"` is implemented.
`empty_action`	Error, or the explicit farthest-observation repair.
`weight_tol`	Weight-threshold root and KKT tolerance.
`weight_max_iter`	Maximum bisection iterations for a weight update.
`solver_tol`	Tolerance used only to certify objective monotonicity.
`solver_max_iter`	Maximum Lloyd iterations.
`strict`	Error rather than return an uncertified iterate.
`keep_permutations`	Retain permutation objectives and index array.

Details

`selection = "max"` chooses the smallest `s` attaining the largest Gap. `"one_se"` uses `se(s) = sqrt(1 + 1/B) * sd_b(log(O_perm,b(s)))` and chooses the smallest `s` whose Gap is at least `max(Gap) - se(s_max)`, where `s_max` is the smallest Gap maximizer. This fully states the implemented one-SE convention; no paper simulation is reproduced.

Value

A `sparse_kmeans_selection` object with all observed fits, Gap/SE table, selected bound and fit, and optional reference details.

References

Witten, D. M. and Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105, 713–726. doi:10.1198/jasa.2010.tm09415.

Examples

```
x <- rbind(c(-3, 0), c(-2, 1), c(2, 0), c(3, 1))
sparse_kmeans_select_s(
  x, clusters = 2, s_grid = c(1, sqrt(2)), n_permutations = 2,
  permutation_seed = 9, initial = c(1, 3), selection = "max"
)
```

---

sparse_kmedian *Coordinatewise sparse K-median clustering*

Description

Implement the Chapter 7 sparse K-median baseline. Cluster centers and the global center are coordinatewise medians, using the midpoint for even samples. Feature `j` receives improvement `D_j = sum_i |x_ij-median_j(x)| - sum_k sum_(i in Ck) |x_ij-median_j(Ck)|`. The non-negative weight block obeys the same L2/L1 constraints and soft-threshold rule as sparse K-means; assignment minimizes weighted L1 distance.

Usage

```
sparse_kmedian(
  x,
  clusters,
  s_w,
  initial = NULL,
  initialization = c("maxmin", "random"),
  first_index = 1L,
  seed = NULL,
  ties = "first",
  empty_action = c("error", "farthest"),
```

```
  score_tol = 1e-12,
  weight_tol = 1e-10,
  weight_max_iter = 200L,
  solver_tol = 1e-10,
  solver_max_iter = 100L,
  strict = TRUE,
  keep_distances = FALSE
)
```

Arguments

x	Numeric observation matrix, with observations in rows.
clusters	Number of clusters.
s_w	Required L1 weight bound in [1, sqrt(p)].
initial	Optional center matrix or vector of distinct row indices.
initialization	Max-min or seeded random initialization when initial is NULL.
first_index	First max-min seed row.
seed	Explicit seed required for random initialization; otherwise NULL.
ties	Exact tie rule; only deterministic "first" is implemented.
empty_action	Error, or the explicit farthest-observation repair.
score_tol	Explicit relative roundoff tolerance for the theoretically non-negative median-improvement scores.
weight_tol	Weight-threshold root and KKT tolerance.
weight_max_iter	Maximum bisection iterations for a weight update.
solver_tol	Tolerance used only to certify objective monotonicity.
solver_max_iter	Maximum Lloyd iterations.
strict	Error rather than return an uncertified iterate.
keep_distances	Retain the final squared-distance matrix.

Details

This is the book-defined axis-aligned baseline, not a claim of a unique primary sparse-median algorithm and not a rotation-equivariant spatial- median method. All-zero improvements error. A negative score within the explicit score_tol roundoff bound is set to zero and counted; a larger violation errors.

Value

A sparse_kmedian_fit object with midpoint medians, feature improvements, weights, labels, weighted-L1 objective and certificates.

References

Feng, L. (2026). *High-Dimensional Data Analysis for Elliptical Symmetric Distributions*, Chapter 7 (book manuscript). This is the book-defined axis-aligned baseline, not a separately attributed primary method.

Examples

```
x <- rbind(c(-4, 0), c(-2, 0), c(2, 0), c(8, 0))
sparse_kmedian(x, clusters = 2, s_w = 1, initial = c(1, 3))
```

`sparse_plugin_lda` *Sparse matrix plug-in LDA classifier*

Description

Applies the Gaussian plug-in LDA score using exactly one supplied sparse precision or covariance matrix and classwise sample means (or two explicit locations). A supplied covariance is inverted only after an ordinary Cholesky factorization proves it is strictly positive definite. A supplied precision must itself be strictly positive definite. The function never uses a pseudoinverse, ridge, symmetrization, or eigenvalue clipping.

Usage

```
sparse_plugin_lda(
  x,
  y,
  precision = NULL,
  covariance = NULL,
  locations = NULL,
  prior = NULL,
  tie = c("class1", "class2")
)
```

Arguments

`x`	Numeric training matrix with observations in rows.
`y`	Binary response. Its first factor level is class 1.
`precision`	Optional supplied sparse precision matrix.
`covariance`	Optional supplied sparse covariance matrix. Exactly one of precision and covariance must be supplied.
`locations`	NULL for classwise sample means, or a list of two finite classwise location vectors.
`prior`	Two positive class probabilities summing to one. NULL uses the shared classifier default.
`tie`	Which class receives a score exactly equal to zero.

Value

An object of class hd_classifier_fit.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Wiley.

Feng, L. (2026). *High-Dimensional Data Analysis for Elliptical Symmetric Distributions*, Chapter 5 (book manuscript). The supplied sparse-matrix adapter is a package construction, not a separate primary method.

Examples

```
x <- rbind(
  c(2, 1), c(1, 2), c(2, 2), c(3, 1),
  c(-2, -1), c(-1, -2), c(-2, -2), c(-3, -1)
)
y <- factor(rep(c("first", "second"), each = 4))
sparse_plugin_lda(x, y, precision = diag(2))
```

---

sparse_plugin_qda *Certified sparse plug-in QDA*

Description

Constructs a quadratic discriminant rule from explicitly supplied class means and covariance matrices. Both covariance matrices must pass strict symmetry, Cholesky, and reciprocal-condition certificates. The function never estimates or repairs a covariance, making it suitable for plugging in externally obtained sparse positive-definite estimators.

Usage

```
sparse_plugin_qda(
  x,
  y,
  mean1,
  mean2,
  covariance1,
  covariance2,
  prior = "equal",
  strict = TRUE
)
```

Arguments

x	Numeric training matrix used to define features and class levels.
y	Two-class response.
mean1, mean2	Explicit finite class mean vectors.
covariance1, covariance2	Explicit symmetric positive-definite class covariance matrices.
prior	Equal, empirical, or explicitly supplied positive class prior.
strict	Whether covariance-certificate failure is an error; otherwise return an invalid fit with a warning.

Value

An hd_classifier_fit whose score is twice the log posterior density ratio of class 1 to class 2.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Wiley.

Feng, L. (2026). *High-Dimensional Data Analysis for Elliptical Symmetric Distributions*, Chapter 5 (book manuscript). The supplied sparse-matrix adapter is a package construction, not a separate primary method.

Examples

```
x <- rbind(c(-1, 0), c(-2, 1), c(1, 0), c(2, -1))
y <- factor(rep(c("left", "right"), each = 2))
fit <- sparse_plugin_qda(
  x, y, mean1 = c(-1.5, 0.5), mean2 = c(1.5, -0.5),
  covariance1 = diag(2), covariance2 = diag(c(1, 2))
)
```

---

sparse_sm_select_k *Select K for Sparse–SM by the BWDM rule*

Description

Retunes tau for every candidate K, then evaluates average between-median and within-median distances in the retained subspace. The selected K maximizes the degree-of-freedom adjusted BWDM index.

Usage

```
sparse_sm_select_k(
  x,
  k_grid,
  tau_grid,
  B = NULL,
  permutations = NULL,
  seed = NULL,
  init = "maxmin",
  first_index = 1L,
  tol = 1e-08,
  max_iter = 100L,
  spatial_max_iter = 500L,
  zero_tol = 0,
  empty_action = c("error", "farthest"),
  empty_active = c("error", "largest", "all"),
  cycle_action = c("error", "return"),
  ties = "first",
  selection_ties = "smallest",
  keep_selections = FALSE,
  keep_permutations = FALSE
)
```

Arguments

x	Numeric observation-by-variable matrix or data frame.
k_grid	Candidate integers satisfying 2 <= K < n.
tau_grid	Finite non-negative threshold grid.
B	Number of reference permutations when permutations is absent.
permutations	Optional list of n by p column-index permutation matrices, or an n by p by B array.
seed	Required explicit RNG seed when permutations are generated.
init	Either "maxmin", K distinct row indices, or a finite K by p center matrix.
first_index	First max-min seed when init = "maxmin".
tol	Positive spatial-median equation tolerance.
max_iter	Positive outer iteration limit.
spatial_max_iter	Positive modified-Weiszfeld iteration limit.
zero_tol	Non-negative zero-residual tolerance.
empty_action	"error" or the explicit deterministic "farthest" repair. The latter only donates from a cluster of size at least two.
empty_active	Action when no score reaches tau: "error", the first "largest" score, or "all" coordinates.
cycle_action	Whether a detected repeated state is an "error" or is returned with a failed convergence certificate.

`ties` Deterministic tie rule; currently only `"first"` is supported.

`selection_ties` Tie rule for equal Gap values; only `"smallest"` is supported.

`keep_selections`
Whether to retain every candidate's threshold selection object.

`keep_permutations`
Whether to retain the complete permutation plan.

Value

A `sparse_sm_k_selection` object with the selected `K`, selected fit, and exact ABDM, AWDM, and BWDM ingredients.

References

Zhao, P., Zhuang, D., and Feng, L. (2026). Sparse K-spatial-median clustering for high-dimensional data. arXiv:2605.00598. https://arxiv.org/abs/2605.00598.

Examples

```
x <- rbind(
  c(-1, -0.3), c(0, 0.1), c(1, 0.4),
  c(5, -0.2), c(6, 0.2), c(7, 0.5)
)
permutations <- list(cbind(
  c(2:6, 1), c(4:6, 1:3)
))
sparse_sm_select_k(
  x, k_grid = c(2, 3), tau_grid = 0, permutations = permutations,
  init = "maxmin", tol = 1e-6, spatial_max_iter = 2000,
  empty_action = "farthest"
)
```

---

sparse_sm_select_tau *Select the Sparse–SM threshold by the permutation Gap criterion*

Description

Fits Sparse–SM over a finite threshold grid and maximizes `log(O(tau)) - mean(log(O_perm(tau)))`, where every separation criterion is evaluated in that fit's retained subspace. This is method-internal calibration, not reproduction of a paper simulation.

Usage

```
sparse_sm_select_tau(
  x,
  K,
  tau_grid,
  B = NULL,
  permutations = NULL,
  seed = NULL,
  init = "maxmin",
  first_index = 1L,
  tol = 1e-08,
  max_iter = 100L,
  spatial_max_iter = 500L,
  zero_tol = 0,
  empty_action = c("error", "farthest"),
  empty_active = c("error", "largest", "all"),
  cycle_action = c("error", "return"),
  ties = "first",
  selection_ties = "smallest",
  keep_fits = FALSE,
  keep_permutations = FALSE
)
```

Arguments

x	Numeric observation-by-variable matrix or data frame.
K	Number of clusters.
tau_grid	Finite non-negative threshold grid.
B	Number of reference permutations when permutations is absent.
permutations	Optional list of n by p column-index permutation matrices, or an n by p by B array.
seed	Required explicit RNG seed when permutations are generated.
init	Either "maxmin", K distinct row indices, or a finite K by p center matrix.
first_index	First max-min seed when init = "maxmin".
tol	Positive spatial-median equation tolerance.
max_iter	Positive outer iteration limit.
spatial_max_iter	Positive modified-Weiszfeld iteration limit.
zero_tol	Non-negative zero-residual tolerance.
empty_action	"error" or the explicit deterministic "farthest" repair. The latter only donates from a cluster of size at least two.
empty_active	Action when no score reaches tau: "error", the first "largest" score, or "all" coordinates.
cycle_action	Whether a detected repeated state is an "error" or is returned with a failed convergence certificate.

`ties`	Deterministic tie rule; currently only `"first"` is supported.
`selection_ties`	Tie rule for equal Gap values; only `"smallest"` is supported.
`keep_fits`	Whether to retain every observed-data fit.
`keep_permutations`	Whether to retain the complete permutation plan.

Value

A `sparse_sm_tau_selection` object containing the selected threshold, selected fit, and all observed/reference Gap ingredients.

References

Zhao, P., Zhuang, D., and Feng, L. (2026). Sparse K-spatial-median clustering for high-dimensional data. arXiv:2605.00598. https://arxiv.org/abs/2605.00598.

Examples

```
x <- rbind(
  c(-4, 0, 2), c(-3, 1, 2), c(-2, 2, 2),
  c(2, 0.2, -2), c(3, 1.2, -2), c(4, 2.2, -2)
)
permutations <- list(
  cbind(6:1, c(3:6, 1:2), c(2:6, 1)),
  cbind(c(2:6, 1), 6:1, c(4:6, 1:3))
)
sparse_sm_select_tau(
  x, K = 2, tau_grid = c(0, 1), permutations = permutations,
  init = c(1, 6), tol = 1e-6, spatial_max_iter = 2000,
  empty_active = "largest"
)
```

`sparse_spatial_sign_pca` *Sparse spatial-sign principal component analysis*

Description

Forms the Chapter 1 sample spatial-sign covariance matrix and applies the certified truncated-power solver. Zero residuals have the same explicit semantics as `spatial_sign_pca()`; scores project the centered original observations rather than their signs.

Usage

```
sparse_spatial_sign_pca(
  x,
  sparsity,
  components = 1L,
  center = c("spatial", "mean", "none"),
  divisor = c("n", "nonzero"),
  zero_action = c("zero", "error"),
  initial = NULL,
  median_tol = 1e-08,
  median_max_iter = 500L,
  zero_tol = 0,
  solver_tol = 1e-08,
  solver_max_iter = 1000L,
  symmetry_tol = sqrt(.Machine$double.eps),
  strict = TRUE,
  keep_operator = TRUE
)
```

Arguments

`x`	Numeric matrix or data frame with observations in rows.
`sparsity`	Required support size, scalar or one value per component.
`components`	Number of sparse components.
`center`	Numeric center or one of `"spatial"`, `"mean"`, and `"none"`.
`divisor`	`"n"` or the number of nonzero residuals.
`zero_action`	Either retain exact zero sign contributions or reject.
`initial`	Optional deterministic loading starts.
`median_tol`, `median_max_iter`	Spatial-median controls.
`zero_tol`	Non-negative zero-residual tolerance.
`solver_tol`, `solver_max_iter`	Truncated-power controls.
`symmetry_tol`	Positive relative operator certification tolerance.
`strict`	If `TRUE`, an uncertified location or sparse solver stops; otherwise an invalid fit is returned with a warning.
`keep_operator`	Retain the SSCM.

Value

A `sparse_spatial_sign_pca_fit` inheriting from `hd_pca_fit`.

References

Zhao, Y., Wang, L., and Feng, L. (2024). Spatial-sign based high-dimensional principal component analysis. arXiv:2409.13267.

Examples

```
x <- rbind(c(4, 0, 0), c(-4, 0, 0), c(0, 2, 0), c(0, -2, 0))
sparse_spatial_sign_pca(x, sparsity = 1, center = "none")
```

`spatial_kendall` *Multivariate spatial Kendall matrix*

Description

Computes the exact U-statistic average of outer products of spatial signs of all pairwise differences. Pairwise differencing makes the estimator translation invariant and removes the need to estimate location.

Usage

```
spatial_kendall(x, zero_tol = 0)
```

Arguments

`x`	Observations in rows.
`zero_tol`	Tolerance for tied pairwise differences.

Value

A positive semidefinite matrix with `n_pairs` and `n_zero_pairs` attributes. Its trace is `1 - n_zero_pairs / n_pairs`; in particular, it has unit trace when there are no tied pairs.

References

Han, F. and Liu, H. (2018). ECA: High-dimensional elliptical component analysis in non-Gaussian distributions. *Journal of the American Statistical Association*, 113, 252-268.

Examples

```
set.seed(2)
spatial_kendall(matrix(rnorm(40), 10, 4))
```

spatial_median *Spatial median*

Description

Minimizes the weighted average Euclidean distance to the observations. The implementation uses a modified Weiszfeld iteration that remains valid when an iterate coincides with an observation.

Usage

```
spatial_median(
  x,
  weights = NULL,
  initial = NULL,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  warn = TRUE
)
```

Arguments

x	A numeric matrix or data frame with observations in rows.
weights	Optional non-negative observation weights.
initial	Optional starting location. Coordinatewise weighted medians are used by default.
tol	Tolerance for the normalized subgradient-equation residual.
max_iter	Maximum number of modified Weiszfeld iterations.
zero_tol	Non-negative tolerance for coincident observations.
warn	If TRUE, warn when the iteration does not converge.

Value

A numeric location vector. Convergence diagnostics are stored in the objective, iterations, converged, relative_change, and equation_residual attributes.

References

Oja, H. (2010). *Multivariate Nonparametric Methods with R*. Springer.

Examples

```
x <- rbind(c(0, 0), c(1, 0), c(0, 1), c(20, 20))
spatial_median(x)
```

spatial_rank *Empirical spatial ranks*

Description

For every row x_i, computes

$$m^{-1} \sum_{j=1}^{m} U(x_i - y_j)$$

with respect to the rows y_j of a reference sample. When `reference` is omitted, the reference is `x`, including the zero self-difference exactly as in the book definition.

Usage

```
spatial_rank(x, reference = NULL, zero_tol = 0)
```

Arguments

`x`	Evaluation observations in rows.
`reference`	Optional reference observations in rows.
`zero_tol`	Non-negative tolerance below which differences are zero.

Value

A matrix of empirical spatial ranks.

References

Oja, H. (2010). *Multivariate Nonparametric Methods with R: An Approach Based on Spatial Signs and Ranks*. Springer. doi:10.1007/9781441904683.

Examples

```
spatial_rank(rbind(c(0, 0), c(1, 0), c(0, 1)))
```

spatial_rank_covariance *Spatial rank covariance matrix*

Description

Computes the average outer product of the empirical spatial ranks defined with denominator `n`, including zero self-differences.

Usage

```
spatial_rank_covariance(x, zero_tol = 0)
```

Arguments

x	Evaluation observations in rows.
zero_tol	Non-negative tolerance below which differences are zero.

Value

A positive semidefinite matrix.

References

Oja, H. (2010). *Multivariate Nonparametric Methods with R: An Approach Based on Spatial Signs and Ranks*. Springer. doi:10.1007/9781441904683.

Examples

```
set.seed(3)
spatial_rank_covariance(matrix(rnorm(30), 10, 3))
```

---

spatial_sign *Spatial signs*

Description

Computes the spatial sign map $U(x) = x/\|x\|_2$, with the zero vector mapped to zero. For a matrix, rows are observations and columns are variables.

Usage

```
spatial_sign(x, center = NULL, zero_tol = 0)
```

Arguments

x	A numeric vector, matrix, or data frame.
center	Optional center to subtract from every observation.
zero_tol	Non-negative tolerance below which a norm is treated as zero. The default detects exact zeros only and therefore preserves scale equivariance.

Value

A vector when x is a vector, otherwise a matrix. The result carries norms and n_zero attributes.

References

Oja, H. (2010). *Multivariate Nonparametric Methods with R: An Approach Based on Spatial Signs and Ranks*. Springer. doi:10.1007/9781441904683.

Examples

```
spatial_sign(c(3, 4))
spatial_sign(rbind(c(3, 4), c(0, 0)))
```

`spatial_sign_change_point_test` *Spatial-sign max-Linf, max-L2, and adaptive change-point test*

Description

Implements the feasible procedure of Liu, Feng, Peng, and Wang (2025). Endpoint blocks of size `floor(n * endpoint_fraction)` supply two joint scaled-spatial-median/diagonal fits. Their diagonals are normalized by their first entries and averaged,

$$\widehat{D} = \{\widehat{D}_1/\widehat{d}^2_{1,1} + \widehat{D}_2/\widehat{d}^2_{2,1}\}/2,$$

exactly as in the primary paper. The same stable endpoint blocks estimate ζ_1 and $\mathrm{tr}(R^2)$. This differs from the book's full-sample shortcut.

Usage

```
spatial_sign_change_point_test(
  x,
  lambda,
  endpoint_fraction = 0.2,
  variant = c("unweighted", "weighted"),
  combination = c("fisher", "none"),
  fv_draws = 4999L,
  alpha = 0.05,
  seed = NULL,
  tol = 1e-08,
  max_iter = 1000L,
  zero_tol = 0,
  keep_reference = FALSE,
  strict = TRUE
)
```

Arguments

`x`	Numeric `n` by `p` matrix.
`lambda`	Integer boundary removal parameter.

`endpoint_fraction`
: Stable endpoint proportion in `(0, 1/2)`.

`variant`
: Unweighted or weighted primary pair of max-Linf/max-L2 tests.

`combination`
: Fisher adaptive test, or `"none"` to report components with the max-Linf p-value as the `htest` p-value.

`fv_draws`
: Intrinsic Gaussian-process draws for the unweighted max-L2 null CDF.

`alpha`
: Test level.

`seed`
: Optional calibration seed; previous RNG state is restored.

`tol`, `max_iter`, `zero_tol`
: Joint fixed-point controls. The default `zero_tol = 0` follows the exact formula; positive values are explicit user-requested failure tolerances.

`keep_reference`
: Whether to retain Gaussian-process maxima.

`strict`
: Failure contract; no failed segment fit is silently dropped.

Details

For every candidate k, separate joint fits give the two segment location estimates. With the common endpoint diagonal, the max-Linf statistic is based on

$$C_\gamma(k) = \{u(1-u)\}^{1-\gamma}\sqrt{n}\,\widehat{D}^{-1/2}(\widehat{\theta}_{1:k} - \widehat{\theta}_{k+1:n}).$$

The max-L2 statistic uses the full-sample sign partial sums. For `variant = "unweighted"`, the primary normalization is `S / sqrt(2 * trace_hat)`, not the book's `p*S/(2*trace_hat)`. Its Gaussian-process CDF is approximated on the actual trimmed scan grid. For `variant = "weighted"`, the primary pivot uses `abs(S_dagger) / sqrt(2 * trace_hat)`; the absolute value and square root are both essential.

Endpoint trace sums are ordered pairs: each block contributes

$$\frac{p^2}{2m(m-1)}\sum_{i\neq j}(U_i^T U_j)^2.$$

The adaptive p-value is the primary Fisher combination. No paper size or power simulation is reproduced; the Gaussian-process draws are intrinsic null calibration only.

Value

An `htest` object containing both primary component statistics, feasible radial/trace quantities, and worst-iteration diagnostics.

References

Liu, J., Feng, L., Peng, L. and Wang, Z. (2025), arXiv:2504.19306.

Examples

```
x <- rbind(
  c(-2, 0), c(2, 0), c(0, -2), c(0, 2),
  c(-1, -1), c(1, 1), c(-1, 1), c(1, -1),
  c(-2, 1), c(2, -1), c(-1, 2), c(1, -2)
)
spatial_sign_change_point_test(
  x, lambda = 3, endpoint_fraction = 0.25,
  fv_draws = 99, seed = 1
)
```

---

spatial_sign_max_test *Spatial-sign max test for a high-dimensional location*

Description

Tests $H_0 : \theta = \mu$ with the feasible max statistic of Liu, Feng, Zhao and Wang. The full-sample scaled spatial median and diagonal HR scale are fitted first. With $\widehat{r}_i = \|\widehat{D}^{-1/2}(X_i - \widehat{\theta})\|$ and $\widehat{\zeta}_1 = n^{-1}\sum_i \widehat{r}_i^{-1}$,

$$T_{\mathrm{MAX}} = n\|\widehat{D}^{-1/2}(\widehat{\theta} - \mu)\|_\infty^2 \, p\widehat{\zeta}_1^2(1 - n^{-1/2}).$$

Every factor shown is multiplicative, including the unsquared finite-sample factor. Under the paper's high-dimensional conditions,

$$T_{\mathrm{MAX}} - 2\log p + \log\log p$$

has limiting cdf $F(t) = \exp\{-\pi^{-1/2}\exp(-t/2)\}$. The vector alternative is scientifically two-sided, while large max statistics use the upper tail.

Usage

```
spatial_sign_max_test(
  x,
  mu = NULL,
  alpha = 0.05,
  tol = 1e-07,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

x	A numeric matrix or data frame with observations in rows and variables in columns. At least two observations and positive marginal sample variation are required.
mu	A finite numeric null-location vector. The default is zero.

alpha	A finite test level strictly between zero and one.
tol	A finite positive estimating-equation tolerance.
max_iter	A positive integer maximum number of recursion updates.
zero_tol	A non-negative threshold for declaring an internally standardised residual radius singular. The literal default is zero.
strict	Whether non-convergence is an error (TRUE) or warning with the last finite iterate returned (FALSE).

Details

strict, singular residual handling, and diagonal-scale identification are as described in scaled_spatial_median(). At least two variables are required for the literal $\log\log p$ centering.

Value

An object of class c("hd_location_test", "htest"). It contains the raw max statistic, centered Gumbel statistic, upper-tail probability and log tails, critical values, the full-sample fit, radial moment, and convergence diagnostics.

References

Liu, B., Feng, L., Zhao, P. and Wang, Z. Spatial-sign based maxsum test for high-dimensional location parameters. *Statistica Sinica*, accepted. doi:10.5705/ss.202024.0051.

Examples

```
x <- matrix(c(-2, -1, 0, 1, 2, 3, -1, 2, 1, -2, 3, 0,
              1, 0, -1, 2, -2, 1), 6, 3)
spatial_sign_max_test(x, tol = 1e-6)
```

---

spatial_sign_maxsum_test

Spatial-sign max-sum test for a high-dimensional location

Description

Combines the feasible spatial-sign max p-value from spatial_sign_max_test() with the feasible Feng–Sun sum p-value returned directly by feng_sun_one_sample_test(). For component p-values p_{MAX} and p_{SUM}, the published combination is

$$1 - G[0.5\tan\{\pi(0.5 - p_{\mathrm{MAX}})\} + 0.5\tan\{\pi(0.5 - p_{\mathrm{SUM}})\}],$$

where G is the standard Cauchy cdf.

Usage

```
spatial_sign_maxsum_test(
  x,
  mu = NULL,
  alpha = 0.05,
  tol = 1e-07,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

x	A numeric matrix or data frame with observations in rows and variables in columns. At least two observations and positive marginal sample variation are required.
mu	A finite numeric null-location vector. The default is zero.
alpha	A finite test level strictly between zero and one.
tol	A finite positive estimating-equation tolerance.
max_iter	A positive integer maximum number of recursion updates.
zero_tol	A non-negative threshold for declaring an internally standardised residual radius singular. The literal default is zero.
strict	Whether non-convergence is an error (TRUE) or warning with the last finite iterate returned (FALSE).

Details

The implementation retains both component log tails and evaluates the combination in signed-log form, finishing with atan2. It therefore does not clip component probabilities. Exact equal endpoints map to the same endpoint; exact reverse endpoints 0 and 1 are reported as mathematically indeterminate rather than silently set to one half. The sum component uses literal leave-two-out fits and consequently requires at least four observations.

Value

An object of class c("hd_location_test", "htest") containing the combined p-value, both complete component-test objects, their ordinary and logarithmic p-value tails, and signed-log Cauchy diagnostics.

References

Liu, B., Feng, L., Zhao, P. and Wang, Z. Spatial-sign based maxsum test for high-dimensional location parameters. *Statistica Sinica*, accepted. doi:10.5705/ss.202024.0051.

Examples

```
x <- matrix(c(-2, -1, 0, 1, 2, 3, 1, -1,
              -1, 2, 1, -2, 3, 0, -2, 1,
              1, 0, -1, 2, -2, 1, 3, -1), 8, 3)
spatial_sign_maxsum_test(x, tol = 1e-6)
```

---

spatial_sign_pca *Spatial-sign principal component analysis*

Description

Decomposes the sample spatial-sign covariance matrix (SSCM). When the spatial-median center is certified, the default operator conventions match `sscm()`: zero residuals map to zero and the divisor is `n`. Scores are projections of the centered original observations, not projections of their spatial signs.

Usage

```
spatial_sign_pca(
  x,
  rank = NULL,
  center = c("spatial", "mean", "none"),
  divisor = c("n", "nonzero"),
  zero_action = c("zero", "error"),
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  keep_operator = TRUE
)
```

Arguments

`x`	Numeric matrix or data frame with observations in rows.
`rank`	Number of loading vectors. `NULL` uses `min(n, p)`.
`center`	A numeric center or one of `"spatial"`, `"mean"`, and `"none"`.
`divisor`	Either `"n"`, the book and Chapter 1 definition, or `"nonzero"`, which divides by the number of nonzero residual signs.
`zero_action`	`"zero"` maps residuals no larger than `zero_tol` to zero; `"error"` rejects such a sample.
`tol, max_iter`	Spatial-median convergence controls.
`zero_tol`	Non-negative tolerance for a zero residual. The default detects exact zeros and preserves scale equivariance.
`keep_operator`	If `TRUE`, retain the decomposed SSCM.

Value

An object inheriting from `hd_pca_fit`. `eigenvalues` contains the full spectrum and `loadings` contains the requested leading vectors.

References

Taskinen, S., Kankainen, A., and Oja, H. (2012). Sign covariance matrix estimate with an application to principal components. *Statistics & Probability Letters*, 82, 1153–1161.

Examples

```
x <- rbind(c(3, 0), c(-3, 0), c(0, 1), c(0, -1))
spatial_sign_pca(x, rank = 1, center = "none")
```

`spatial_sign_poet` *Spatial-sign POET estimator for an elliptical factor model*

Description

Implements POET-SS from Xu et al. The pilot is

$$\widehat{\Sigma}_0 = \frac{p}{n} \sum_i U(X_i - \widehat{\mu}) U(X_i - \widehat{\mu})^T,$$

its supplied leading `factors` eigenpairs form the low-rank component, and a generalized thresholding rule is applied only to off-diagonal entries of the raw idiosyncratic complement. When `threshold = NULL`, the common threshold is `constant * (sqrt(log(p)/n) + sqrt(log(n)/n))`. The unknown sufficiently-large theoretical constant is not estimated from the same data or borrowed from a paper simulation.

Usage

```
spatial_sign_poet(
  x,
  factors,
  threshold = NULL,
  constant = 1,
  rule = c("hard", "soft", "scad", "adaptive_lasso"),
  scad_a = 3.7,
  adaptive_eta = 1,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

x	Numeric matrix or data frame with observations in rows.
factors	Supplied non-negative integer number of factors.
threshold	Optional common threshold in trace-normalized scatter units. NULL uses the primary rate with constant.
constant	Non-negative multiplier for the primary rate threshold.
rule	One of "hard", "soft", "scad", or "adaptive_lasso".
scad_a, adaptive_eta	Generalized-threshold rule parameters.
tol, max_iter, zero_tol	Spatial-median controls.
strict	If TRUE, fail when the spatial median is not certified; otherwise return an explicitly invalid diagnostic object.

Details

The result may be indefinite in finite samples because thresholding is used literally. No eigenvalue floor, ridge, nearest-PD projection, or pseudoinverse is applied.

Value

An elliptical_factor_fit with the pilot, factor eigensystem, raw and thresholded idiosyncratic components, and reconstructed scatter.

References

Xu, X., Ma, H., Wang, H. and Feng, L. (2025). arXiv:2512.19325.

Examples

```
x <- rbind(c(-3, -2, 0), c(-2, -1, 1), c(-1, 0, -1),
           c(1, 0, 1), c(2, 1, -1), c(3, 2, 0))
spatial_sign_poet(x, factors = 1, threshold = 0.1)
```

---

spatial_sign_precision

Robust spatial-sign SCLIME and SGLASSO precision estimation

Description

Estimates the inverse trace-normalized shape $V_0 = \Lambda_0^{-1} = \{tr(\Sigma_0)/p\}\Sigma_0^{-1}$ using the spatial-sign covariance matrix $\widehat{S}$ centered at the ordinary sample spatial median. Set $A = p\widehat{S}$. With method = "sclime" the paper's problem

$$\min_V \|V\|_1 \quad \text{subject to} \quad \|AV - I\|_\infty \le \lambda$$

is solved column by column. Every column must pass primal feasibility, dual feasibility, l1 stationarity, and relative primal-dual gap checks. The raw solution is then symmetrized exactly as in Lu and Feng (2025): for each transposed pair, retain the entry with smaller absolute value.

Usage

```
spatial_sign_precision(
  x,
  lambda,
  method = c("sclime", "sglasso"),
  median_tol = 1e-08,
  median_max_iter = 500L,
  zero_tol = 0,
  solver_tol = 1e-07,
  solver_max_iter = 100000L,
  initial_step = 1,
  max_backtracking = 100L,
  strict = TRUE
)
```

Arguments

x	Numeric matrix or data frame with observations in rows.
lambda	Positive tuning parameter in the primary-paper objective.
method	Either "sclime" or "sglasso".
median_tol, median_max_iter, zero_tol	Spatial-median and zero-sign controls.
solver_tol	Positive tolerance required for every solver certificate.
solver_max_iter	Positive maximum number of primal-dual or proximal iterations.
initial_step	Initial SGLASSO proximal step; ignored by SCLIME.
max_backtracking	Maximum SGLASSO line-search reductions per update.
strict	If TRUE, fail on a spatial-median or solver certificate failure. If FALSE, return an explicitly invalid diagnostic fit with estimate = NULL.

Details

With method = "sglasso", the function solves

$$\min_{V \succ 0} \{tr(AV) - \log\det V + \lambda \|V\|_1\}.$$

Here $\|V\|_1$ is the paper's full elementwise norm, so the diagonal is penalized. A positive-definite backtracking proximal-gradient algorithm is used and the full subgradient KKT residual must not exceed solver_tol.

The SCLIME primal-dual iteration is a convergent Chambolle–Pock method; SGLASSO uses the standard convex proximal-gradient majorization inequality. An optimizer stopping because of solver_max_iter is not called a valid estimator. With strict = FALSE, a failed run returns estimate = NULL plus its last-iterate certificate; it never returns a matrix that looks like a valid paper estimator. Neither method uses a ridge, eigenvalue floor, pseudoinverse, constraint relaxation, or post-hoc KKT repair.

The constants in the paper's theoretical choices of λ_n are not observable tuning formulas. Therefore lambda must be supplied rather than silently replacing them by an undocumented default.

Value

An object of class `spatial_sign_precision_fit`. A successful fit contains `estimate`, the sample SSCM, $p\widehat{S}$, the spatial median, and complete feasibility/KKT diagnostics.

References

Lu, Z. and Feng, L. (2025). Robust sparse precision matrix estimation and its applications. arXiv:2503.03575. https://arxiv.org/abs/2503.03575.

Examples

```
x <- rbind(
  c(-2, 0, 1), c(-1, 1, 0), c(0, -1, 2), c(1, 0, -1),
  c(2, 1, 1), c(0, 2, -2), c(-1, -2, 0), c(1, -1, 1)
)
spatial_sign_precision(x, lambda = 0.4, method = "sglasso")
```

---

spatial_sign_precision_lda

Spatial-sign precision plug-in LDA

Description

Combines classwise robust locations with a certified SCLIME or SGLASSO inverse-shape fit from spatial_sign_precision(). Plain matrices, invalid fits, thresholded fits, and fits that have lost their original feasibility/KKT certificate are rejected. The equal-prior score is

$$(z - (\widetilde{\mu}_1 + \widetilde{\mu}_2)/2)^\top \widehat{\Omega}(\widetilde{\mu}_1 - \widetilde{\mu}_2).$$

Because the decision is equal-prior, the inverse-shape scale is irrelevant.

Usage

```
spatial_sign_precision_lda(
  x,
  y,
  precision_fit,
  locations = NULL,
  prior = c(0.5, 0.5),
  median_tol = 1e-08,
  median_max_iter = 500L,
  zero_tol = 0,
  strict = TRUE,
  tie = c("class1", "class2")
)
```

Arguments

x	Numeric training matrix with observations in rows.
y	Binary response. Its first factor level is class 1.
precision_fit	A valid, certified object returned by spatial_sign_precision() with method sclime or sglasso.
locations	NULL to compute classwise spatial medians, or a list of two supplied finite robust location vectors.
prior	Equal class probabilities; non-equal probabilities are rejected for this scale-free elliptical rule.
median_tol, median_max_iter, zero_tol	Spatial-median controls used only when locations is NULL.
strict	If TRUE, spatial-median nonconvergence is an error. If FALSE, it produces an explicitly invalid fit.
tie	Which class receives a score exactly equal to zero.

Value

An object of class hd_classifier_fit.

References

Lu, Z. and Feng, L. (2025). Robust sparse precision matrix estimation and its applications. doi:10.48550/arXiv.2503.03575.

Examples

```
x <- rbind(
  c(2, 1), c(1, 2), c(2, 2), c(3, 1),
  c(-2, -1), c(-1, -2), c(-2, -2), c(-3, -1)
)
y <- factor(rep(c("first", "second"), each = 4))
residual <- rbind(
  sweep(x[1:4, ], 2, colMeans(x[1:4, ]), "-"),
  sweep(x[5:8, ], 2, colMeans(x[5:8, ]), "-")
)
precision_fit <- spatial_sign_precision(
  residual, lambda = 0.5, method = "sglasso"
)
spatial_sign_precision_lda(x, y, precision_fit)
```

`spatial_sign_sphericity_test` *Spatial-sign sphericity test*

Description

This is the constant-score special case of `hallin_paindaveine_shape_test()`. It reports $Q_S = p\,\mathrm{tr}(\Omega - I_p/p)^2$ and calibrates $n(p+2)Q_S/2$ by chi-squared with $(p-1)(p+2)/2$ degrees of freedom.

Usage

```
spatial_sign_sphericity_test(
  x,
  center = NULL,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0
)
```

Arguments

`x`	Numeric matrix with observations in rows.
`center`	A supplied finite center vector. `NULL` estimates the spatial median.
`tol`, `max_iter`, `zero_tol`	Controls passed to `spatial_median()` when the center is estimated.

Value

An `hd_covariance_test` object.

References

Hallin, M. and Paindaveine, D. (2006). *Annals of Statistics*, 34, 2707–2756. doi:10.1214/009053606000000731.

Examples

```
set.seed(39)
x <- matrix(rnorm(30), nrow = 15, ncol = 2)
spatial_sign_sphericity_test(x, center = c(0, 0))
```

`srivastava_du_one_sample_test` *Srivastava–Du one-sample high-dimensional mean test*

Description

Tests $H_0 : \mu = \mu_0$ using the diagonal-standardised Srivastava–Du statistic. The alternative is unrestricted, while the non-negative quadratic evidence is calibrated in the upper tail of its asymptotic standard normal distribution.

Usage

```
srivastava_du_one_sample_test(x, mu = NULL)
```

Arguments

`x`	A numeric matrix or data frame with observations in rows and variables in columns. At least four observations are required.
`mu`	A numeric vector giving the null mean. The default is the zero vector.

Details

The implementation follows the finite-sample centring and scale correction in Srivastava and Du (2008). It computes $\mathrm{tr}(R^2)$ through the smaller of a primal $p \times p$ matrix and a dual $N \times N$ Gram matrix. Before centring, each column of `x` and the matching entry of `mu` are divided by one common column scale. This is an algebraically neutral change of units that protects the statistic at very large or small numerical scales; reported means, differences, and marginal variances are converted back to the input units. Every sample variance must be strictly positive; no ridge, absolute-value repair, or variance flooring is applied.

Value

An object of class `c("hd_location_test", "htest")`. In addition to the standardised statistic and upper-tail p-value, `components` contains `A`, `nu`, `trace.R2`, `c`, the null centre, the denominator variance, the sample mean, the mean difference, and the marginal sample variances.

References

Srivastava, M. S. and Du, M. (2008). A test for the mean vector with fewer observations than the dimension. *Journal of Multivariate Analysis*, 99, 386–402.

Examples

```
set.seed(21)
x <- matrix(rnorm(32), nrow = 8, ncol = 4)
srivastava_du_one_sample_test(x)
```

`srivastava_katayama_kano_two_sample_test`
Srivastava–Katayama–Kano two-sample mean test

Description

Tests equality of two high-dimensional mean vectors using the diagonal-standardised statistic of Srivastava, Katayama, and Kano (SKK). The two samples may have different covariance matrices. With unbiased sample covariance matrices S_1 and S_2, define

$$\widehat{D} = \operatorname{diag}(S_1)/n_1 + \operatorname{diag}(S_2)/n_2,$$

$$Q = (\bar{X}_1 - \bar{X}_2)^{\mathsf{T}} \widehat{D}^{-1} (\bar{X}_1 - \bar{X}_2), \qquad \widehat{q} = (Q - p)/\sqrt{p}.$$

Usage

```
srivastava_katayama_kano_two_sample_test(x, y)
```

Arguments

`x`, `y`	Numeric matrices or data frames with observations in rows and the same variables in columns. Each group must have at least two observations.

Details

This implementation uses the correction in the published corrigendum. If $A_k = \widehat{D}^{-1/2} S_k \widehat{D}^{-1/2}$, it computes

$$\widehat{F}_k = p^{-1} \left\{ \operatorname{tr}(A_k^2) - \operatorname{tr}(A_k)^2/(n_k - 1) \right\},$$

and

$$\widehat{G} = p^{-1} \operatorname{tr}(A_1 A_2).$$

Thus

$$\widehat{V}_q = 2\widehat{F}_1/n_1^2 + 2\widehat{F}_2/n_2^2 + 4\widehat{G}/(n_1 n_2).$$

The feasible finite-sample factor is calculated from the *sample* matrix

$$\widehat{R} = A_1/n_1 + A_2/n_2, \qquad \widehat{c} = 1 + \operatorname{tr}(\widehat{R}^2)/p^{3/2},$$

and the reported statistic is

$$Z_{\mathrm{SKK}} = \widehat{q}/\sqrt{\widehat{V}_q \widehat{c}}.$$

Trace functionals are formed through a $p \times p$ primal Gram matrix only when $p \le n_1 + n_2$; otherwise an $(n_1 + n_2) \times (n_1 + n_2)$ dual Gram matrix is used, so the high-dimensional path does not construct a $p \times p$ object. Before centring, both samples are divided by the same scale within each variable. This algebraically neutral change of units protects very large and small inputs while preserving componentwise diagonal-scale invariance. Every entry of $\widehat{D}$ and the final normalising variance must be strictly positive. No ridge, absolute-value repair, or variance flooring is used.

Value

An object of class `c("hd_location_test", "htest")`. Besides the upper-tail asymptotic normal p-value, `components` contains `Q.raw`, `q.scaled`, `F1`, `F2`, `G`, the variance estimate, the sample `trace.R.hat2` and `c.hat`, sample means, and marginal variance estimates. `diagnostics` records the primal/dual path, internal column scales, covariance denominators, and the absence of regularisation.

References

Srivastava, M. S., Katayama, S., and Kano, Y. (2013). A two sample test in high dimensional data. *Journal of Multivariate Analysis*, **114**, 349–358. doi:10.1016/j.jmva.2012.08.014.

Srivastava, M. S., Katayama, S., and Kano, Y. (2013). Corrigendum to "A two sample test in high dimensional data". *Journal of Multivariate Analysis*, **120**, 251. doi:10.1016/j.jmva.2013.04.016.

Examples

```
set.seed(2301)
x <- matrix(rnorm(60), 12, 5)
y <- matrix(rnorm(70, 0.2), 14, 5)
srivastava_katayama_kano_two_sample_test(x, y)
```

---

sscca *Primary spatial-sign sparse canonical correlation analysis*

Description

Fits the sparse CCA estimator proposed by Qian, Liu, and Feng. For paired rows, a joint spatial median and joint sample spatial-sign covariance matrix are computed, then the first sparse pair solves

$$\max_{w_x, w_y} \; w_x^T (pS_{xy}) w_y - \lambda_x \|w_x\|_1 - \lambda_y \|w_y\|_1$$

subject to $w_x^T (pS_{xx}) w_x \le 1$ and $w_y^T (pS_{yy}) w_y \le 1$, where $p = p_x + p_y$.

Usage

```
sscca(
  x,
  y,
  lambda_x,
  lambda_y,
  selection = c("fixed", "bic1", "bic2"),
  tol = 1e-07,
  max_iter = 500L,
  inner_tol = 1e-09,
  inner_max_iter = 5000L,
  median_tol = 1e-08,
```

```
  median_max_iter = 500L,
  zero_tol = 0,
  zero_action = c("zero", "error"),
  support_tol = 0,
  metric_tol = sqrt(.Machine$double.eps),
  strict = TRUE
)
```

Arguments

x, y	Paired finite numeric matrices, observations in rows.
lambda_x, lambda_y	Non-negative penalties. Scalars are required for fixed fitting; explicit sequences are allowed for BIC selection.
selection	One of "fixed", "bic1", or "bic2".
tol	Positive relative tolerance for the alternating directions.
max_iter	Maximum alternating iterations.
inner_tol	Positive KKT/update tolerance for each metric-lasso block.
inner_max_iter	Maximum coordinate-descent sweeps per candidate.
median_tol, median_max_iter	Controls passed to the joint spatial median fit.
zero_tol	Non-negative exact/near-zero spatial-sign tolerance.
zero_action	Use the package convention "zero" for $U(0) = 0$, or fail with "error" when a fitted residual is zero.
support_tol	Non-negative absolute threshold used only to count BIC degrees of freedom and report support. It does not alter coefficients.
metric_tol	Positive relative tolerance for PSD/rank diagnostics; no eigenvalue is floored or projected.
strict	If TRUE, method failures are errors. Otherwise an invalid hd_cca_fit is returned and no last iterate is presented as an estimate.

Details

Every conditional metric-lasso block is solved with its actual diagonal. This corrects the referenced mixedCCA coordinate update, which omits division by the metric diagonal and is exact only when that diagonal is one. With selection = "bic1" or "bic2", each alternating block chooses from the explicitly supplied lambda sequence using the two criteria printed in the primary paper. No lambda grid, ridge, pseudoinverse, or matrix repair is invented by this function.

Value

An sscca_fit and hd_cca_fit object containing the two canonical coefficient vectors, raw and sign scores, spatial-sign blocks, selected penalties, BIC paths, convergence and KKT certificates.

References

Qian, J., Liu, W., and Feng, L. (2025). High dimensional sparse canonical correlation analysis for elliptical symmetric distributions. arXiv:2504.13018.

Examples

```
t <- seq_len(12)
x <- cbind(x1 = sin(t), x2 = cos(t / 2))
y <- cbind(y1 = sin(t) + 0.2 * cos(t), y2 = cos(t / 2) - 0.1 * sin(t))
fit <- sscca(x, y, lambda_x = 0.01, lambda_y = 0.01)
fit$canonical.correlations
```

---

sscm *Spatial sign covariance matrix*

Description

Computes the average outer product of centered spatial signs. By default the sample is centered at its spatial median, matching Chapter 1 of the book.

Usage

```
sscm(x, center = "spatial", tol = 1e-08, max_iter = 500L, zero_tol = 0)
```

Arguments

x	Observations in rows.
center	Either a numeric center or one of "spatial", "mean", and "none".
tol, max_iter	Convergence controls used when estimating a spatial center.
zero_tol	Tolerance for zero residuals.

Value

A positive semidefinite matrix with center and n_zero attributes. Its trace is one when there are no zero residuals.

References

Visuri, S., Koivunen, V., and Oja, H. (2000). Sign and rank covariance matrices. *Journal of Statistical Planning and Inference*, 91, 557-575.

Examples

```
set.seed(1)
sscm(matrix(rnorm(60), 20, 3))
```

`sslda` *Spatial-sign direct sparse linear discriminant analysis*

Description

Computes classwise spatial medians and SSCMs, pools the SSCMs with their class sample sizes, and solves

$$\min_{\gamma} \|\gamma\|_1 : \|(p\widetilde{S} + ridgeI)\gamma - (\widetilde{\mu}_1 - \widetilde{\mu}_2)\|_\infty \leq \lambda.$$

The explicit ridge changes the constraint operator and is therefore stored as part of the method rather than treated as a numerical repair. This implementation follows the equal-prior SSLDA rule.

Usage

```
sslda(
  x,
  y,
  lambda,
  prior = c(0.5, 0.5),
  ridge = 0,
  median_tol = 1e-08,
  median_max_iter = 500L,
  zero_tol = 0,
  solver_tol = 1e-07,
  solver_max_iter = 100000L,
  strict = TRUE,
  tie = c("class1", "class2")
)
```

Arguments

`x`	Numeric training matrix with observations in rows.
`y`	Binary response. Its first factor level is class 1.
`lambda`	Positive Dantzig constraint radius; it is never selected implicitly.
`prior`	Equal class probabilities; non-equal probabilities are rejected because the SSLDA theory and score in the primary paper assume equal priors.
`ridge`	Explicit non-negative ridge added to the pooled covariance. Zero implements the unmodified paper operator.
`median_tol`, `median_max_iter`, `zero_tol`	Spatial-median and zero-sign controls.
`solver_tol`	Positive tolerance required by every solver certificate.
`solver_max_iter`	Positive maximum number of primal–dual iterations.
`strict`	If TRUE, a failed certificate is an error. If FALSE, an explicitly invalid, non-predictable fit is returned with a warning.
`tie`	Which class receives a score exactly equal to zero.

Value

An object of class hd_classifier_fit.

References

Zhuang, D. and Feng, L. (2025). Spatial sign based direct sparse linear discriminant analysis for high dimensional data. doi:10.48550/arXiv.2504.11117.

Examples

```
x <- rbind(
  c(2, 1), c(1, 2), c(2, 2), c(3, 1),
  c(-2, -1), c(-1, -2), c(-2, -2), c(-3, -1)
)
y <- factor(rep(c("first", "second"), each = 4))
sslda(x, y, lambda = 0.25, ridge = 0.1)
```

---

ssqda *Spatial-sign sparse quadratic discriminant analysis*

Description

Fits SSQDA with classwise spatial medians and covariance surrogates equal to a trace estimate times the spatial-sign covariance matrix. The ordered triple-U trace is evaluated through its exact $O(np)$ identity, $\sum_i \|X_i - \bar{X}\|^2/(n-1)$. Each class must contain at least three observations. The Dantzig programs, determinant certificate, equal-prior contract, and score orientation are the same as in SDAR.

Usage

```
ssqda(
  x,
  y,
  lambda_D = NULL,
  lambda_beta = NULL,
  parameter_grid = NULL,
  folds = 10L,
  median_tol = 1e-08,
  median_max_iter = 1000L,
  zero_tol = 0,
  solver_tol = 1e-07,
  feasibility_tol = 1e-07,
  solver_max_iter = 10000L,
  strict = TRUE
)
```

Arguments

x	Numeric training matrix with observations in rows.
y	Two-class response. Its first observed level is class 1.
lambda_D, lambda_beta	Explicit non-negative Dantzig bounds.
parameter_grid	Paired lambda_D and lambda_beta columns for deterministic stratified joint cross-validation.
folds	Number of folds used only with a parameter grid.
median_tol	Positive spatial-median equation tolerance.
median_max_iter	Positive spatial-median iteration limit.
zero_tol	Non-negative threshold for a zero spatial residual. No perturbation is made when such a residual occurs.
solver_tol	Positive optimization and certificate tolerance.
feasibility_tol	Positive tolerance for final constraint feasibility.
solver_max_iter	Positive optimization iteration limit.
strict	Whether numerical failure is an error; otherwise an invalid hd_classifier_fit is returned with a warning.

Value

An hd_classifier_fit; non-negative scores select class 1.

References

Feng, L. (2025). Spatial sign based sparse quadratic discriminant analysis for high-dimensional elliptical distributions. arXiv:2504.11187.

Examples

```
x <- rbind(
  c(-3, -0.5), c(-2.2, 1.1), c(-1.4, -1.3),
  c(-2.7, 1.8), c(-0.9, 0.4), c(-1.8, -2),
  c(2.8, 0.2), c(1.9, 1.7), c(1.2, -1.8),
  c(2.5, 2.2), c(0.7, -0.2), c(1.6, -2.4)
)
y <- factor(rep(c("left", "right"), each = 6))
fit <- ssqda(x, y, lambda_D = 1, lambda_beta = 1,
             solver_tol = 1e-6)
```

`tensor_spatial_sign_precision` *Sparse tensor-elliptical precision matrices from tensor spatial signs*

Description

Implements the Spatial-Sign Separate tensor lasso of Liu, Lu, Zhou, Feng, and Wang. Observations may be a list of identically dimensioned numeric arrays or one numeric array whose first dimension indexes observations. R's column-major vectorization is used: the first tensor mode varies fastest, and mode-k matricization follows Kolda–Bader ordering.

Usage

```
tensor_spatial_sign_precision(
  data,
  lambda,
  center = NULL,
  median_tol = 1e-08,
  median_max_iter = 500L,
  zero_tol = 0,
  solver_tol = 1e-07,
  solver_max_iter = 10000L,
  initial_step = 1,
  max_backtracking = 100L,
  keep_whitened = FALSE,
  strict = TRUE
)
```

Arguments

`data`	A list of same-dimension numeric tensors, or a numeric array with observations in its first dimension. A numeric matrix represents n observations of an order-one tensor.
`lambda`	One non-negative tuning value or one value per mode. The value is λ_k in the paper, not the rescaled glasso penalty.
`center`	`NULL` for the ordinary sample spatial median, or a finite numeric vector/array with the tensor mode dimensions.
`median_tol`, `median_max_iter`, `zero_tol`	Controls for the ordinary spatial median and tensor spatial signs. Exact zero residuals map to zero signs.
`solver_tol`	Positive tolerance required simultaneously for KKT and relative-update certificates.
`solver_max_iter`	Positive graphical-lasso iteration limit.
`initial_step`	Positive initial proximal-gradient step.

`max_backtracking`
: Positive line-search reduction limit per iteration.

`keep_whitened`
: If `TRUE`, retain the vectorized tensors after other-mode pilot whitening for each target mode. This can be large.

`strict`
: If `TRUE`, stop on any median, pilot, or solver failure. If `FALSE`, return an explicitly invalid fit with no estimate.

Details

The ordinary spatial median of the vectorized observations is used unless `center` is supplied. Write $U_{i,(l)}$ for the mode-l matricization of tensor sign U_i. The mode-l pilot is

$$\widetilde{\Omega}_l^{raw} = \left\{ \frac{p_l}{n} \sum_i U_{i,(l)} U_{i,(l)}^T \right\}^{-1}$$

exactly when $np_* > p_l^2(p_l - 1)/2$; otherwise it is I_{p_l}. Every pilot is Frobenius-normalized. Other modes are whitened with the symmetric positive-definite square roots of these normalized pilots, yielding

$$\widehat{S}_k = \frac{p_k}{n} \sum_i V_i^{(k)} V_i^{(k)T}.$$

Each raw precision matrix minimizes the primary-paper objective

$$\frac{1}{p_k}\{\mathrm{tr}(\widehat{S}_k\Omega) - \log\det(\Omega)\} + \lambda_k \sum_{a \neq b} |\Omega_{ab}|.$$

Thus the equivalent ordinary graphical-lasso penalty is $\rho_k = p_k\lambda_k$; diagonal entries are not penalized. A self-contained positive-definite proximal-gradient solver is used. A fit is valid only if its objective is non-increasing, every raw solution is SPD, the diagonal and off-diagonal KKT residuals do not exceed `solver_tol`, and the relative update meets `solver_tol`. The reported normalized estimates are raw solutions divided by their Frobenius norms; KKT certificates refer to the raw solutions.

No ridge, jitter, eigenvalue floor, pseudoinverse, or post-hoc repair is applied. A singular pilot on the paper's inverse branch is therefore a failure. With `strict = FALSE`, any such failure or any uncertified solver run returns `estimate = NULL` and `valid = FALSE`, together with available diagnostics; it never exposes a last iterate as a valid estimate.

Value

An object of class `tensor_spatial_sign_precision_fit`. A valid fit contains `estimate`, `raw.precision`, the center and tensor signs, mode dimensions, pilot source scatters and branches, pilot square roots, whitened mode scatters, paper and effective penalties, and complete SPD, descent, KKT, and relative-update certificates.

References

Liu, J., Lu, Z., Zhou, L., Feng, L., and Wang, Z. (2025). Tensor Elliptical Graphic Model. arXiv:2508.00333. https://arxiv.org/abs/2508.00333.

Examples

```
x <- array(c(
  -2, -1, 1, 2, -1, 1, 2, -2,
  1, -2, 2, -1, 2, 1, -1, -2,
  2, 1, -2, -1, -2, 2, 1, -1,
  -1, 2, -2, 1, 1, -2, 2, -1
), dim = c(8, 2, 2))
fit <- tensor_spatial_sign_precision(
  x, lambda = 1, center = array(0, c(2, 2))
)
fit$estimate
```

---

threshold_spatial_sign_precision

Threshold a spatial-sign precision estimate

Description

Applies the support-recovery rule of Lu and Feng (2025),

$$\widetilde{v}_{ij} = \widehat{v}_{ij} 1\{|\widehat{v}_{ij}| \geq \tau\}.$$

The comparison is non-strict and applies to all entries, including the diagonal, exactly as defined in the primary paper. The paper's theoretical threshold contains unknown constants, so `tau` is deliberately explicit.

Usage

```
threshold_spatial_sign_precision(fit, tau)
```

Arguments

`fit`	A valid object returned by `spatial_sign_precision()`.
`tau`	One finite non-negative threshold.

Value

A list containing the thresholded matrix, logical support, sign matrix, threshold, and source fit.

References

Lu, Z. and Feng, L. (2025). Robust sparse precision matrix estimation and its applications. arXiv:2503.03575.

Examples

```
x <- rbind(
  c(-2, 0), c(-1, 1), c(0, -1), c(1, 0), c(2, 1), c(0, 2)
)
fit <- spatial_sign_precision(x, 0.4, method = "sglasso")
threshold_spatial_sign_precision(fit, tau = 0.1)
```

`threshold_tensor_spatial_sign_precision` *Threshold fitted tensor spatial-sign precision matrices*

Description

Applies the book's corrected threshold rule $\widehat{\Omega}_{ab} I\{|\widehat{\Omega}_{ab}| \geq \tau_k\}$ to every off-diagonal entry of a valid tensor spatial-sign precision fit. Equality is retained and diagonal entries are always preserved. The arXiv display omits the absolute-value bars, but its proof explicitly treats both positive and negative edges; the absolute-value rule is therefore used here.

Usage

```
threshold_tensor_spatial_sign_precision(fit, tau)
```

Arguments

`fit`	A valid object returned by `tensor_spatial_sign_precision`.
`tau`	One non-negative threshold or one threshold per mode.

Value

A list with thresholded modewise precision estimates, thresholds, and the original unthresholded estimates.

References

Liu, J., Lu, Z., Zhou, L., Feng, L., and Wang, Z. (2025). Tensor Elliptical Graphic Model. arXiv:2508.00333. https://arxiv.org/abs/2508.00333.

Examples

```
x <- array(
  sin(seq_len(48)) + cos(seq_len(48) / 3), dim = c(12, 2, 2)
)
fit <- tensor_spatial_sign_precision(x, lambda = 1, center = c(0, 0, 0, 0))
threshold_tensor_spatial_sign_precision(fit, tau = 0.05)
```

`tinst_two_sample_test` *Huang–Liu–Zhou–Feng two-sample inverse norm sign test*

Description

Tests equality of two multivariate location vectors with the two-sample inverse norm sign test (tINST) of Huang, Liu, Zhou, and Feng (2023). The implementation uses the paper's feasible cross statistic, observation-wise leave-one-out diagonal estimates, and the inverse norm weight $\omega(r) = r^{-1}$. The manuscript's three group-index typographical errors are corrected: every leave-one-out sum uses the corresponding group size n_k, every diagonal update is an update of D_k, and both indices of the second within-group trace estimator run over group 2.

Usage

```
tinst_two_sample_test(
  x,
  y,
  alpha = 0.05,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

`x, y`	Numeric matrices or data frames with observations in rows and the same variables in columns. Each group must have at least three rows.
`alpha`	Significance level used for the returned upper-tail rejection decision.
`tol`	Positive convergence tolerance for every joint and weighted location iteration.
`max_iter`	Positive maximum number of updates for every individual fit.
`zero_tol`	Non-negative tolerance for detecting a zero diagonally standardized residual. The default detects exact zeros only.
`strict`	If `TRUE`, error when any required iterative fit does not become stable. If `FALSE`, warn and return the last iterates together with diagnostics. A large score residual alone does not trigger this contract.

Details

Every full-sample and leave-one-out diagonal fit starts at the sample mean and marginal sample variances. The inverse-norm weighted location fit starts at its leave-one-out sample mean. The source article says to iterate until convergence but does not prescribe a tolerance. Here `iteration.stable` means that the relative location update (and, for a joint fit, the log-diagonal update) is at most `tol`. The separately returned `score.residual` is diagnostic only and is not presented as a certificate that the estimating equation has been solved. This distinction matters for inverse-norm iteration because it can approach an included observation. The diagonal fixed-point equations identify only relative

coordinate scales; this implementation retains the common scale carried by the sample-variance initialization and applies no determinant or trace normalization.

No zero marginal variance is replaced, no ridge is added, and a non-positive or non-finite published variance estimate is an error. With `strict = TRUE` (the default), failure of any required iteration to become stable is also an error. Setting `strict = FALSE` returns the last iterates with a warning and complete per-fit stability and score diagnostics.

A common translation and common coordinatewise scaling are applied internally before iteration. This is algebraically neutral because tINST is invariant to common shifts and nonsingular diagonal changes of units, and it protects residual calculations from avoidable cancellation and overflow.

Value

An object of class `c("hd_location_test", "htest")`. Its `components` field contains the feasible statistic, nuisance estimates, ordered-pair trace estimates, variance terms, leave-one-out radii and directions, and internally scaled fit estimates. Its `diagnostics` field contains all iteration counts, update sizes, estimating-equation residuals, the rejection rule, and the explicit no-repair contract.

References

Huang, X., Liu, B., Zhou, Q., and Feng, L. (2023). A high-dimensional inverse norm sign test for two-sample location problems. *Canadian Journal of Statistics*, 51, 1004–1033. doi:10.1002/cjs.11731.

Examples

```
set.seed(2023)
x <- matrix(stats::rt(24, df = 5), 8, 3)
y <- matrix(stats::rt(30, df = 5), 10, 3)
tinst_two_sample_test(x, y)
```

`truncated_power_pca` *Truncated-power sparse principal components*

Description

Applies the truncated-power iteration to a supplied positive-semidefinite operator. At every iteration only the `sparsity` largest absolute entries of the matrix-vector product are retained; exact ties are resolved by the smallest coordinate index. Multiple components use the Mackey deflation $(I - vv')M(I - vv')$.

Usage

```
truncated_power_pca(
  operator,
  sparsity,
  components = 1L,
```

```
  initial = NULL,
  solver_tol = 1e-08,
  solver_max_iter = 1000L,
  symmetry_tol = sqrt(.Machine$double.eps),
  strict = TRUE,
  keep_operator = TRUE
)
```

Arguments

operator	Finite symmetric positive-semidefinite matrix.
sparsity	Required support size, scalar or one value per component.
components	Number of components.
initial	Optional p by components deterministic starting matrix.
solver_tol	Positive fixed-point tolerance.
solver_max_iter	Positive maximum number of iterations per component.
symmetry_tol	Positive relative symmetry/PSD certification tolerance.
strict	If TRUE, an uncertified solver stops; otherwise a warning and an invalid hd_pca_fit are returned.
keep_operator	Retain the supplied operator in the fit.

Details

The returned certificate reports unit-norm and support violations, the sign-invariant truncated fixed-point residual, objective monotonicity, and iteration count. Because an operator rather than observations is supplied, scores, center, and n are NULL, NULL, and NA.

Value

A truncated_power_pca_fit inheriting from hd_pca_fit.

References

Yuan, X.-T. and Zhang, T. (2013). Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14, 899–925.

Examples

```
truncated_power_pca(diag(c(4, 2, 1)), sparsity = 1)
```

`tyler_shape` *Tyler's shape estimator*

Description

Solves Tyler's fixed-point equation and applies the book's trace normalization, $\mathrm{tr}(V) = p$. The exact unregularized estimator requires more observations than variables and data in general position.

Usage

```
tyler_shape(
  x,
  center = "mean",
  initial = NULL,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  warn = TRUE
)
```

Arguments

`x`	Observations in rows.
`center`	Either a numeric center or one of `"mean"`, `"spatial"`, and `"none"`. The default sample mean preserves affine equivariance of the resulting Tyler shape. A spatial-median center is more resistant to magnitude outliers but is only orthogonally equivariant.
`initial`	Optional positive definite starting shape.
`tol`	Relative Frobenius convergence tolerance.
`max_iter`	Maximum number of fixed-point iterations.
`zero_tol`	Tolerance used to detect undefined zero residuals. The default detects exact zeros only.
`warn`	If `TRUE`, warn when the iteration does not converge.

Value

A trace-normalized shape matrix. Convergence diagnostics and the center are stored as attributes.

References

Tyler, D. E. (1987). A distribution-free M-estimator of multivariate scatter. *The Annals of Statistics*, 15, 234-251.

Examples

```
set.seed(4)
x <- matrix(rt(200, df = 3), 50, 4)
tyler_shape(x)
```

`wang_feng_dms_test` *Wang–Feng double-max-sum change-point test*

Description

Implements the feasible DMS procedure of Wang and Feng (2023). With Rice difference variances $\widehat{\sigma}_j^2$, the sparse component is either

$$M = \max_{k,j} |C_{0,j}(k)|$$

or the trimmed weighted version

$$M^{\dagger} = \max_{\lambda \le k \le n-\lambda, j} |C_{1/2,j}(k)|.$$

The dense component is, literally,

$$S = \sum_{k=1}^{n-1} \sum_{j=1}^{p} C_{1/2,j}(k)^2,$$

not the book's displayed trimmed sum of `gamma = 0` CUSUMs. It is centered by `(n + 2) * p` and studentized by the primary paper's leave-four and leave-three finite-difference estimators.

Usage

```
wang_feng_dms_test(
  x,
  gamma = 0,
  lambda = NULL,
  combination = c("fisher", "cauchy"),
  alpha = 0.05,
  strict = TRUE
)
```

Arguments

`x`	Numeric `n` by `p` data matrix.
`gamma`	Either `0` or `0.5` for the max component.
`lambda`	Boundary removal integer required when `gamma = 0.5`.
`combination`	Primary Fisher rule or a labelled Cauchy extension.
`alpha`	Test level.
`strict`	Failure contract; failed calculations return no estimate when `FALSE` and error when `TRUE`.

Details

The unweighted max pivot is `2 * M^2 - log(2 * p)`. The weighted pivot is `A(p * log(h)) * M - D(p * log(h))`, where `h = ((lambda / n)^(-1) - 1)^2`. These correct the nested-log and h errors in the book. `combination = "fisher"` is the primary DMS rule. The optional Cauchy rule is clearly labelled a software extension.

The leaveout diagonal is the paper's displayed `A_m = {2,...,n} \\ {i_1,...,i_m}` estimator. No bridging adjacent difference is silently inserted. A non-positive component variance or feasible DMS variance is a method failure; it is never repaired by an absolute value or floor.

Value

An `htest` object with max, sum, trace, variance, and localization components.

References

Wang, G. and Feng, L. (2023), JRSS B 85, 936–958; arXiv:2205.00709.

Examples

```
x <- rbind(matrix(-0.5, 6, 3), matrix(0.5, 6, 3))
x <- x + matrix(seq_len(length(x)) %% 5, nrow(x), ncol(x)) / 10
wang_feng_dms_test(x, gamma = 0)
```

---

wang_liu_feng_ma_serial_panel_test

Wang-Liu-Feng-Ma serial-panel Fisher independence test

Description

Implements the serial-correlation-aware panel procedure of Wang, Liu, Feng and Ma. The dense component is the signed-correlation sum

$$S_N = \sqrt{2/\{N(N-1)\}} \sum_{i<j} \hat{\rho}_{ij}$$

with the published leave-two-units-out variance estimator. The sparse component uses $L_N = \max_{i<j} \hat{\rho}_{ij}^2$ and the temporal effective dimension $\mathrm{tr}^2(\widetilde{\Sigma})/\|\widetilde{\Sigma}\|_F^2$. Fisher's statistic combines the two upper-tail p-values and is calibrated by χ_4^2.

Usage

```
wang_liu_feng_ma_serial_panel_test(
  panel,
  regressors = NULL,
  temporal_covariance = NULL,
  nu = 1.42,
  component = c("fisher", "max", "sum"),
  keep_matrices = FALSE
)
```

Arguments

`panel`	Numeric T by N residual or outcome matrix.
`regressors`	As in `feng_jiang_liu_xiong_panel_independence_test()`.
`temporal_covariance`	Optional supplied positive-definite T by T temporal covariance. If `NULL`, use the paper's thresholded sample estimator.
`nu`	Threshold constant, strictly greater than $\sqrt{2}$; used only by the internal temporal estimator. The default `1.42` is the value used in the primary paper's applications.
`component`	One of `"fisher"`, `"max"`, or `"sum"`.
`keep_matrices`	Whether to retain correlation and temporal-estimation matrices.

Details

The internal temporal estimator is implemented literally: the cross-unit sample covariance is hard-thresholded using the paper's $\hat{P}_N$ and $\nu > \sqrt{2}$. No positive-definite projection, ridge, or replacement of a non-positive $\hat{P}_N$ is applied. A scientifically justified temporal covariance estimate may instead be supplied explicitly.

Value

An object inheriting from `htest`.

References

Wang, H., Liu, B., Feng, L. and Ma, Y. (2026). Fisher's Combined Probability Test for Cross-Sectional Independence in Panel Data Models with Serial Correlation. *Statistica Sinica*, 36, 1-21. doi:10.5705/ss.202023.0348

Examples

```
panel <- matrix(c(-2, 1, 0, 2, -1, 3, 1, -2,
                  2, 1, -3, 1, 3, -1, 2, -2,
                  1, 3, -2, 1), 5, 4)
wang_liu_feng_ma_serial_panel_test(
  panel, temporal_covariance = diag(nrow(panel))
)
```

`wang_liu_feng_vector_independence_test` *Wang-Liu-Feng rank max-sum test for vector independence*

Description

Tests independence between two high-dimensional random vectors using the Spearman or Kendall procedures of Wang, Liu and Feng. For every cross-block coordinate pair, the function computes the exact primary rank correlation. It then forms the max statistic, the centered sum of squared correlations, and their Fisher combination. The max null variance is $1/(n-1)$ for Spearman and $2(2n+5)/\{9n(n-1)\}$ for Kendall. The sum variance is estimated by the paper's intrinsic permutation calibration; this is part of the callable method, not a replication simulation.

Usage

```
wang_liu_feng_vector_independence_test(
  x,
  y,
  measure = c("spearman", "kendall"),
  component = c("fisher", "max", "sum"),
  B = 199L,
  seed = NULL,
  keep_correlations = FALSE,
  keep_permutation = FALSE
)
```

Arguments

`x`	Numeric n by p matrix.
`y`	Numeric n by q matrix with the same rows as `x`.
`measure`	Either `"spearman"` or `"kendall"`.
`component`	One of `"fisher"`, `"max"`, or `"sum"`.
`B`	Number of intrinsic permutations used to estimate the sum variance; at least two.
`seed`	Optional non-negative integer. An explicit seed is localized and preserves the caller's R random-number state. `NULL` uses the caller's stream normally.
`keep_correlations`	Whether to retain the p by q matrix of pairwise rank correlations.
`keep_permutation`	Whether to retain the intrinsic permutation statistics and permutation index matrix.

Details

Exact ties are rejected because the primary finite-sample null moments and distribution-free calibration assume continuous margins. The chapter's general mutual-independence construction and the degenerate Hoeffding D, Blum-Kiefer-Rosenblatt R, and Bergsma-Dassios-Yanagimoto tau-star families remain review-only: this function does not invent missing executable variance/eigenspectrum contracts for them.

Value

An object inheriting from `htest`, with max, sum, and Fisher components.

References

Wang, H., Liu, B. and Feng, L. (2026). Testing Independence Between High-Dimensional Random Vectors Using Rank-Based Max-Sum Tests. *Scandinavian Journal of Statistics*, 53, 821-847. doi:10.1111/sjos.70063

Examples

```
x <- matrix(c(1, 4, 2, 6, 3, 5, 2, 6, 1, 5, 3, 4), 6, 2)
y <- matrix(c(6, 2, 5, 1, 4, 3), 6, 1)
wang_liu_feng_vector_independence_test(x, y, B = 19, seed = 7)
```

`wang_liu_feng_vector_u_independence_test`
Wang–Liu–Feng degenerate rank-U vector-independence test

Description

Implements the exact high-order Hoeffding D, Blum–Kiefer–Rosenblatt R, and Bergsma–Dassios–Yanagimoto tau-star examples in the primary paper. Each coordinate-pair U-statistic is computed by enumerating every sample subset and every kernel symmetrization. The maximum, centered sum of squares, intrinsic X-row permutation variance, primary extreme-value tail, and Fisher combination are then evaluated literally.

Usage

```
wang_liu_feng_vector_u_independence_test(
  x,
  y,
  measure = c("hoeffding_d", "bkr_r", "tau_star"),
  component = c("fisher", "max", "sum"),
  B = 199L,
  seed = NULL,
  max_kernel_evaluations = 50000000L,
  keep_estimates = FALSE,
  keep_permutation = FALSE
)
```

Arguments

`x`	Numeric n by p matrix.
`y`	Numeric n by q matrix with matching rows.
`measure`	One of `"hoeffding_d"`, `"bkr_r"`, or `"tau_star"`.
`component`	One of `"fisher"`, `"max"`, or `"sum"`.
`B`	Number of intrinsic X-row permutations; at least two.
`seed`	Optional non-negative integer. An explicit seed is localized and preserves the caller's random-number state.

`max_kernel_evaluations`
: Positive integer upper bound on all exact symmetrized-kernel terms, including the observed statistic and all permutations.

`keep_estimates`
: Whether to retain the p by q matrix of observed U-statistics.

`keep_permutation`
: Whether to retain permutation sum statistics and permutation indices.

Details

This is an auditable exact implementation, not a scalable algorithm. Its kernel-term workload is $(B+1)pq\binom{n}{m}m!$ for kernel order $m = 5, 6, 4$. Computation stops before drawing permutations when this exceeds `max_kernel_evaluations`. No incomplete-U approximation is substituted.

The primary tau-star example prints `(n-1)! 4! / n!`, which conflicts with the paper's generic U-statistic definition and its own null second moment. This implementation uses the coherent generic normalization $1/\binom{n}{4}$ and records that decision in diagnostics.

Value

An `htest` object with exact workload and calibration diagnostics.

References

Wang, H., Liu, B. and Feng, L. (2026). Testing Independence Between High-Dimensional Random Vectors Using Rank-Based Max-Sum Tests. Scandinavian Journal of Statistics 53, 821–847. doi:10.1111/sjos.70063

Examples

```
x <- cbind(c(1, 4, 2, 7, 3, 6, 5), c(7, 2, 5, 1, 6, 3, 4))
y <- cbind(c(2, 7, 4, 1, 6, 3, 5))
wang_liu_feng_vector_u_independence_test(
  x, y, measure = "tau_star", B = 7, seed = 11
)
```

`wang_peng_li_one_sample_test`
: *Wang–Peng–Li one-sample high-dimensional spatial-sign test*

Description

Tests $H_0 : \mu = \mu_0$ against an unrestricted location alternative using the raw spatial-sign statistic of Wang, Peng, and Li (2015). For $Z_i = U(X_i - \mu_0)$, where $U(v) = v/\|v\|$ for nonzero v and $U(0) = 0$, the unstandardised statistic is

$$T_n = \sum_{1 \le i < j \le n} Z_i^{\mathsf{T}} Z_j.$$

Usage

```
wang_peng_li_one_sample_test(x, mu = NULL)
```

Arguments

x	A numeric matrix or data frame with observations in rows and variables in columns. At least three observations are required.
mu	A finite numeric vector giving the null location. The default is the zero vector.

Details

The unknown $\mathrm{tr}(B^2)$ in $\mathrm{Var}(T_n) = n(n-1)\,\mathrm{tr}(B^2)/2$ is estimated by the feasible cross-validation estimator in equation (7) of the original paper:

$$\widehat{\mathrm{tr}(B^2)} = \frac{1}{n(n-1)} \sum_{j \neq k} \mathrm{tr}\{(Z_j - \bar{Z}_{(j,k)}) Z_j^{\mathsf{T}} (Z_k - \bar{Z}_{(j,k)}) Z_k^{\mathsf{T}}\},$$

where $\bar{Z}_{(j,k)}$ is the sign mean after deleting observations j and k. The implementation evaluates this literal leave-two-out formula. It does not substitute the paper's equation (8), whose shortcut uses $\|Z_i\|^2 = 1$; consequently the documented $U(0) = 0$ convention remains coherent even when an observation equals the null location exactly.

Residual directions are normalised after max-absolute scaling, and subtraction is prescaled only when direct finite subtraction overflows. Thus common nonzero rescaling of all residuals leaves the calculation unchanged even at extreme finite units. Exact null residuals contribute a zero sign and are counted in `diagnostics`. A non-finite or non-positive cross-validation variance estimate is a genuine degenerate calibration and raises an error; no absolute value, ridge, or numerical floor is used.

The p-value uses the upper tail of the WPL asymptotic standard normal law. Although the vector alternative is conventionally labelled `two.sided`, large positive quadratic evidence is the rejection direction. The calibration requires the high-dimensional trace and concentration conditions in Wang, Peng, and Li (2015); it is not an exact finite-sample test.

Value

An object of class `c("hd_location_test", "htest")`. Its raw components include `T.WPL`, the literal cross-validation numerator, `trace.B2.hat`, its estimated variance and standard error, and the sum and mean of the spatial signs. `diagnostics` records exact zero signs, overflow-safe subtraction fallbacks, and the no-repair variance policy.

References

Wang, L., Peng, B., and Li, R. (2015). A high-dimensional nonparametric multivariate test for mean vector. *Journal of the American Statistical Association*, **110**, 1658–1669. doi:10.1080/01621459.2014.988215.

Examples

```
set.seed(2601)
x <- matrix(stats::rt(240, df = 4), 20, 12)
wang_peng_li_one_sample_test(x)
```

`wang_xu_approx_randomization_test` *Wang–Xu approximate randomization test*

Description

Tests equality of two high-dimensional mean vectors under unrestricted and potentially unequal covariance matrices using the approximate randomization calibration of Wang and Xu (2022). Observations are rows and variables are columns.

Usage

```
wang_xu_approx_randomization_test(
  x,
  y,
  alpha = 0.05,
  calibration = c("auto", "exact", "monte_carlo"),
  B = 9999L,
  seed = NULL,
  workers = 1L,
  max_exact = 1048576L,
  keep_randomized = FALSE
)
```

Arguments

`x`, `y`	Numeric matrices or data frames containing the two independent samples. Both must have the same variables and at least four rows.
`alpha`	Test level strictly between zero and one.
`calibration`	Reference calculation: `"auto"`, `"exact"`, or `"monte_carlo"`.
`B`	Positive number of Monte Carlo sign draws. It is validated but otherwise ignored under exact calibration.
`seed`	`NULL`, or an integer-valued counter-generator seed in `[0, 2^32 - 1]`. Under Monte Carlo calibration, `NULL` draws and records one seed from R's RNG; an explicit seed makes the result reproducible without changing R's RNG state. Exact enumeration does not use a seed.
`workers`	Number of workers. The first implementation deliberately accepts only `1`; it never claims or silently performs parallel work.

`max_exact` Positive maximum number of global-sign-reduced patterns permitted for exact enumeration.

`keep_randomized`
If `TRUE`, retain every randomized statistic and sign pattern. This can require substantial memory.

Details

The observed statistic is the full-sample Chen–Qin statistic

$$T_{CQ} = \sum_{k=1}^{2} \frac{2}{n_k(n_k-1)} \sum_{i<j} X_{k,i}^{\mathsf{T}} X_{k,j} - \frac{2}{n_1 n_2} \sum_i \sum_j X_{1,i}^{\mathsf{T}} X_{2,j}.$$

The reference sample is not produced by permuting pooled group labels. Within each group the method forms adjacent half-differences

$$\widetilde{X}_{k,i} = (X_{k,2i} - X_{k,2i-1})/2, \qquad i = 1, \ldots, m_k, \quad m_k = \lfloor n_k/2 \rfloor,$$

and multiplies each half-difference by an independent Rademacher sign. If a group size is odd, its final row is deliberately unused by the reference distribution, exactly as in the paper. The observed statistic still uses every row.

With `calibration = "exact"`, all sign configurations are integrated exactly. Since simultaneous reversal of every sign leaves the statistic unchanged, the implementation evaluates one representative of each global-sign pair. The exact conditional tail is

$$2^{-(m_1+m_2)} \sum_e 1\{T_{CQ}(e) \geq T_{CQ}\},$$

with no plus-one correction. With `calibration = "monte_carlo"`, the paper's practical p-value is used literally:

$$\widehat{p} = \frac{1 + \sum_{b=1}^{B} 1\{T_{CQ}^{(b)} \geq T_{CQ}\}}{B+1}.$$

Thus ties are always counted in the upper tail. `calibration = "auto"` selects exact enumeration only when its reduced number of configurations does not exceed `max_exact`.

Exact enumeration removes Monte Carlo error from the conditional reference distribution; it does not make the overall Behrens–Fisher test finite-sample exact. Its level guarantee is asymptotic under the paper's moment and non-dominating-observation assumptions. Pairing follows the input row order, so arbitrary row reordering can change the finite-sample reference distribution.

Value

An object of class `c("hd_location_test", "htest")`. Raw components include the observed Chen–Qin decomposition, half-difference pseudo-samples, pairing maps, scaled quadratic kernel, tail count, and optional randomized values and signs. Diagnostics distinguish exhaustive reference calculation from finite-sample exactness and record the plus-one rule, seed, Monte Carlo resolution, discarded rows, numerical scaling, and degeneracy.

References

Wang, R. and Xu, W. (2022). An approximate randomization test for the high-dimensional two-sample Behrens–Fisher problem under arbitrary covariances. *Biometrika*, 109, 1117–1132. doi:10.1093/biomet/asac014.

Examples

```
x <- matrix(c(1, 0, 2, 1, 4, 1, 5, 3), 4, 2, byrow = TRUE)
y <- matrix(c(0, 2, 1, 4, 3, 5, 4, 7), 4, 2, byrow = TRUE)
wang_xu_approx_randomization_test(x, y, calibration = "exact")
```

`wang_yao_corrected_john_test`

Wang–Yao corrected John's test of sphericity

Description

For $U = p\,\mathrm{tr}(S^2)/\,\mathrm{tr}^2(S) - 1$, the known-zero-mean result is

$$nU - p \;\Rightarrow\; N(1+\beta, 4).$$

With an estimated mean, the primary extension instead centers nU by $np/(n-1)$. The test remains defined for $p \geq n$.

Usage

```
wang_yao_corrected_john_test(x, center = FALSE, beta = 0)
```

Arguments

`x`	Numeric matrix with observations in rows.
`center`	Whether the mean is estimated and removed.
`beta`	Finite fourth cumulant, or `NULL` for the empirical plug-in.

Value

An `hd_covariance_test` object.

References

Wang, Q. and Yao, J. (2013). *Electronic Journal of Statistics*, 7, 2164–2192. doi:10.1214/13-EJS842.

Examples

```
set.seed(41)
x <- matrix(rnorm(72), nrow = 24, ncol = 3)
wang_yao_corrected_john_test(x, center = TRUE)
```

`wang_yao_corrected_lrt` *Wang–Yao corrected likelihood-ratio test of sphericity*

Description

For effective covariance degrees of freedom m and $y = p/m < 1$, this function computes

$$\mathcal{L} = -\log|S| + p\log\{\mathrm{tr}(S)/p\}$$

and calibrates

$$\mathcal{L} + (p-m)\log(1-p/m) - p$$

by a normal distribution with mean $-\log(1-y)/2 + \beta y/2$ and variance $-2\log(1-y) - 2y$ for real data. `center = FALSE` is the known-zero-mean formula in the book. `center = TRUE` uses Wang and Yao's unknown-mean extension, replacing the spectral centering ratio by $p/(n-1)$.

Usage

```
wang_yao_corrected_lrt(x, center = FALSE, beta = 0)
```

Arguments

`x`	Numeric matrix with observations in rows.
`center`	Whether the mean is estimated and removed.
`beta`	Finite fourth cumulant, or `NULL` for the empirical plug-in.

Details

`beta = 0` gives the real Gaussian correction. A supplied value is the standardized fourth cumulant $EZ^4 - 3$; `beta = NULL` uses the scale-standardized empirical fourth moment under the spherical null.

Value

An `hd_covariance_test` object with all centering terms retained.

References

Wang, Q. and Yao, J. (2013). *Electronic Journal of Statistics*, 7, 2164–2192. doi:10.1214/13-EJS842.

Examples

```
set.seed(40)
x <- matrix(rnorm(72), nrow = 24, ncol = 3)
wang_yao_corrected_lrt(x, center = TRUE)
```

`wang_zhao_feng_wang_mutual_fund_fdr` *Robust mutual-fund selection with SS-BH or FSS-BH*

Description

Implements the one-sided procedures of Wang, Zhao, Feng and Wang for $H_{0i} : \alpha_i \leq 0$ against positive fund alpha. With observable factors, the primary observations are the *restricted* residualized returns $Z = M_F Y$; the projection contains no intercept. A simultaneous scaled spatial median gives $\widehat{\theta}$ and $\widehat{D}^{1/2} = \mathrm{diag}(\widehat{d}_i)$. If $\widehat{r}_t = \|\widehat{D}^{-1/2}(Z_t - \widehat{\theta})\|$, then

$$\widehat{\varsigma} = \frac{N\overline{r^{-1}}^2}{1 - 2(1 - \omega_T/T)\overline{r^{-1}}\bar{r} + (1 - \omega_T/T)\overline{r^2}\,\overline{r^{-1}}^2}$$

and $T_i^s = \sqrt{T\widehat{\varsigma}}\widehat{\theta}_i/\widehat{d}_i$. One-sided normal p-values are passed to the ordinary BH step-up rule.

Usage

```
wang_zhao_feng_wang_mutual_fund_fdr(
  returns = NULL,
  factors = NULL,
  restricted_residuals = NULL,
  omega_ratio = NULL,
  q = 0.05,
  adjustment = c("none", "supplied", "spatial_kendall"),
  latent_fit = NULL,
  n_factors = NULL,
  k_max = NULL,
  tol = 1e-07,
  max_iter = 500L,
  zero_tol = 0
)
```

Arguments

`returns`	`NULL` or a finite observation-by-fund matrix. Used when `restricted_residuals` is `NULL`.
`factors`	Optional observed factor vector or matrix. The projection is through the origin, as required by the primary restricted fit.
`restricted_residuals`	Optional already restricted observation-by-fund matrix. If factors are absent, `omega_ratio` is then mandatory.
`omega_ratio`	Optional $\omega_T/T \in (0, 1]$. It is inferred from factors for raw returns.
`q`	Target BH FDR level in `(0,1)`.
`adjustment`	One of `"none"`, `"supplied"`, or `"spatial_kendall"`.

`latent_fit`	For supplied adjustment, a list containing either `adjusted.residuals`, `common.component`, or compatible `loadings` and `scores`, all in the input return units.
`n_factors`	Explicit positive latent factor count.
`k_max`	Explicit upper bound for the paper's eigenvalue-ratio rule; used only when `n_factors` is `NULL`. Both arguments are used only with `adjustment = "spatial_kendall"`; irrelevant tuning arguments are rejected rather than silently ignored.
`tol`	Positive simultaneous scaled-median equation tolerance.
`max_iter`	Positive maximum update count.
`zero_tol`	Non-negative exact-zero radius tolerance. The default zero preserves every nonzero observation.

Details

The book draft instead starts from unrestricted, intercept-containing OLS residuals and replaces $\sqrt{\widehat{\varsigma}}$ by the inverse-radius mean. Both changes contradict the primary definition and erase the alpha signal. This implementation follows the article and returns the full radial correction. The published corrigendum changes only an affiliation.

For FSS-BH, `adjustment = "spatial_kendall"` implements the primary spatial Kendall matrix, $\widehat{\Gamma} = \sqrt{N}(\widehat{\xi}_1, \ldots, \widehat{\xi}_r)$, least-squares scores, and factor removal. Supply a fixed `n_factors`, or explicitly opt into the paper's eigenvalue-ratio selector by supplying `k_max`; there is deliberately no tuning default. Alternatively, `adjustment = "supplied"` accepts a validated nuisance fit. No simulation tuning is embedded.

Value

A `mutual_fund_fdr` object with fundwise statistics and p-values, BH decisions, the complete scaled-median fit, factor diagnostics, and all radial/projection quantities.

References

Wang, H., Zhao, P., Feng, L. and Wang, Z. (2025). Robust mutual fund selection with false discovery rate control. *Journal of Econometrics*, 252, 106121. doi:10.1016/j.jeconom.2025.106121. Preprint: https://arxiv.org/abs/2411.14016.

Wang, H., Zhao, P., Feng, L. and Wang, Z. (2026). Corrigendum. doi:10.1016/j.jeconom.2025.106162.

Examples

```
f <- cbind(market = seq(-1, 1, length.out = 14))
y <- cbind(f[, 1] + sin(1:14),
           0.15 - 0.3 * f[, 1] + cos(1:14),
           -0.1 + 0.2 * f[, 1] + sin(1:14 / 2))
wang_zhao_feng_wang_mutual_fund_fdr(y, f, q = 0.1)
```

`weighted_scaled_spatial_median` *Weighted scaled spatial median and diagonal HR scale*

Description

Fits the weighted diagonal Hettmansperger–Randles equations used by Yan, Zhao, and Feng. With $e_i = D^{-1/2}(X_i - \theta)$, $r_i = \|e_i\|$, and $U_i = U(e_i)$, the fitted pair solves

$$\sum_i r_i^m U_i = 0, \qquad p\ \mathrm{diag}\{n^{-1}\sum_i U_i U_i^{\mathsf{T}}\} = I_p.$$

The default $m = -1$ is the inverse-norm weighted estimator; the paper's theoretical family permits any finite $m \le 1$.

Usage

```
weighted_scaled_spatial_median(
  x,
  m = -1,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

`x`	A numeric matrix or data frame with observations in rows and at least two rows.
`m`	Finite radial power no greater than one. The default is `-1`.
`tol`	Positive relative iteration tolerance.
`max_iter`	Positive maximum number of iterations.
`zero_tol`	Non-negative threshold at which a standardized residual is treated as zero. The default detects exact zeros only.
`strict`	If `TRUE`, non-stabilization is an error. If `FALSE`, the final iterate is returned with a warning and full diagnostics.

Details

The implementation uses the dimensionally coherent iterations

$$\theta^+ = \theta + D^{1/2}\frac{\sum_i r_i^m U_i}{\sum_i r_i^{m-1}}, \qquad D^+ = pD^{1/2}\,\mathrm{diag}\{n^{-1}\sum_i U_i U_i^{\mathsf{T}}\}D^{1/2}.$$

These correct two incompatible $D^{-1/2}$ factors printed in the current arXiv source. The estimating equations, units, and Bahadur representation all require $D^{1/2}$ in those positions.

Value

A list of class `weighted_scaled_spatial_median` containing the fitted location, canonical diagonal scale, standardized radii and directions, radial weights, and convergence diagnostics.

References

Yan, G., Zhao, P., and Feng, L. (2025). Inverse norm weighted maxsum test for high dimensional location parameters. arXiv:2501.14168. https://arxiv.org/abs/2501.14168.

Examples

```
x <- matrix(c(-2, 1, 0, 3, -1, 2, 1, -3, 2, 0, 4, -2), ncol = 2)
weighted_scaled_spatial_median(x, m = 1, tol = 1e-6)
```

---

weighted_spatial_sign_alpha_oracle_test

Oracle weighted spatial-sign alpha statistic

Description

Evaluates the formula-complete weighted spatial-sign class in Chapter 4 from supplied nuisance-free components. For radial weight function K, the statistic is

$$Q_K = \frac{N}{h^T h} \sum_{s \neq t} h_s h_t K(r_s) K(r_t) U_s^T U_t,$$

divided by $\{2\hat{\psi}_{2,K}^2 \widehat{\mathrm{tr}(R^2)}\}^{1/2}$, where $\hat{\psi}_{2,K} = T^{-1} \sum_t K^2(r_t)$.

Usage

```
weighted_spatial_sign_alpha_oracle_test(
  directions,
  radii,
  h,
  trace_R2,
  K,
  data_name = "supplied oracle scores"
)
```

Arguments

`directions`	Numeric T by N matrix of unit spatial signs.
`radii`	Strictly positive finite length-T radial vector.
`h`	Finite length-T residualized-intercept vector.
`trace_R2`	Strictly positive supplied value of $\widehat{\mathrm{tr}(R^2)}$.

K	R function mapping the complete radial vector to a numeric vector of the same length.
data_name	Optional label used in the returned htest object.

Details

This interface deliberately says oracle: it does not estimate directions, radii, factor slopes, diagonal scale, or the trace term, and it does not claim primary-paper feasibility for an arbitrary user function. Use zhao_chen_zi_inst_alpha_test() for the primary inverse-norm endpoint.

Value

An upper-tail alpha-test object whose diagnostics explicitly record the oracle scope.

References

Zhao, P., Chen, D. and Zi, X. (2022). High-dimensional non-parametric tests for linear asset pricing models. Stat 11, e490. doi:10.1002/sta4.490

Examples

```
u <- rbind(c(1, 0), c(0, 1), c(-1, 0), c(0, -1))
weighted_spatial_sign_alpha_oracle_test(
  u, c(1, 2, 3, 4), rep(1, 4), trace_R2 = 3,
  K = function(r) 1 / r
)
```

---

white_noise_portmanteau_test

Classical Box–Pierce or Ljung–Box white-noise test

Description

Computes the univariate portmanteau statistic written in Chapter 4. This function is intentionally univariate; the high-dimensional procedures are provided by the other functions in this file.

Usage

```
white_noise_portmanteau_test(
  x,
  lag = 1L,
  type = c("Ljung-Box", "Box-Pierce"),
  center = TRUE
)
```

Arguments

x	Numeric vector, ordered in time.
lag	Positive truncation lag. It must be smaller than length(x).
type	Either "Ljung-Box" or "Box-Pierce".
center	Whether to subtract the sample mean before computing sample autocorrelations.

Value

An object inheriting from htest. The components field contains every lagged sample autocorrelation and its contribution.

References

Box, G. E. P. and Pierce, D. A. (1970). Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association*, 65, 1509–1526. Ljung, G. M. and Box, G. E. P. (1978). On a measure of lack of fit in time series models. *Biometrika*, 65, 297–303.

Examples

```
x <- c(0.2, -0.1, 0.3, 0.05, -0.2, 0.1, 0.4, -0.3)
white_noise_portmanteau_test(x, lag = 2)
```

---

xu_lin_wei_pan_aspu_test

Xu–Lin–Wei–Pan analytical adaptive sum-of-powers test

Description

Tests equality of two high-dimensional mean vectors with an analytical version of the adaptive sum-of-powers (aSPU) test of Xu, Lin, Wei, and Pan (2016). Observations are rows and variables are columns. Write the inverse-variance-standardised differences as

$$W_j = \frac{\bar{X}_{1j} - \bar{X}_{2j}}{\{\widehat{\sigma}_{1,jj}/n_1 + \widehat{\sigma}_{2,jj}/n_2\}^{1/2}},$$

with the common-covariance version replacing the denominator by the pooled marginal variance times $1/n_1+1/n_2$. With the default score_scale = "paper_raw", the primary paper's finite-power statistic is $L_\gamma = \sum_j (\bar{X}_{1j} - \bar{X}_{2j})^\gamma$; its Gaussian moments use the estimated covariance matrix of the mean difference. With score_scale = "book_studentized", the scale-invariant book variant is $T_\gamma = \sum_j W_j^\gamma$, whose moments use the correlation matrix of W. Both paths use $M = \max_j |W_j|$ for the infinite power.

Usage

```
xu_lin_wei_pan_aspu_test(
  x,
  y,
  powers = c(1:6, Inf),
  score_scale = c("paper_raw", "book_studentized"),
  correlation_source = c("common", "unequal", "supplied"),
  supplied_correlation = NULL,
  standard_errors = NULL,
  bandwidth = NULL,
  psd_adjust = c("error", "eigen_clip"),
  psd_tol = sqrt(.Machine$double.eps),
  miwa_steps = 128L
)
```

Arguments

x, y	Numeric matrices or data frames with observations in rows and the same variables in columns. Each group needs at least two rows.
powers	Distinct positive integer powers, optionally including `Inf`. The default is `c(1:6, Inf)`.
score_scale	Finite-power coordinate definition. `"paper_raw"` (the default) uses the primary paper's unstandardised sample-mean differences. `"book_studentized"` uses the scale-invariant inverse-variance-standardised coordinates in the accompanying book.
correlation_source	How to estimate the null correlation of the coordinate contrasts: pooled common covariance, unequal group covariances, or a supplied correlation matrix.
supplied_correlation	A finite $p \times p$ correlation matrix, required only for `correlation_source = "supplied"`.
standard_errors	Optional finite positive coordinate standard errors for the supplied-correlation path. If `NULL`, unequal-covariance sample standard errors are used.
bandwidth	Optional fixed hard-band width. For a common covariance it is one integer in $0, \ldots, p-1$; for unequal covariances it may be one recycled width or two group-specific widths. `NULL` leaves estimates unbanded. It is unavailable for a supplied correlation.
psd_adjust	Either `"error"` or `"eigen_clip"`. The latter explicitly clips eigenvalues below `psd_tol` and reports the adjustment.
psd_tol	Positive eigenvalue tolerance used by `psd_adjust`.
miwa_steps	Positive integer grid size for deterministic Miwa integration. Miwa supports at most 20 selected powers per parity family.

Details

The two finite-power paths are deliberately explicit because they are not generally numerically equivalent: the primary-paper path is sensitive to coordinate units, whereas the book path is invariant to positive diagonal rescaling. For numerical stability, raw scores and their covariance are divided by one common reported scale before the C++ moment calculation; reported finite-power moments are mapped back to the original units, while standardized statistics and p-values are unaffected by that normalization.

Every finite positive integer γ is allowed. In particular, the original method's default `1:6` includes odd powers; a restriction to even powers in the accompanying book draft is a transcription error. Under the Gaussian working limit, the standardized odd-power statistics use a joint two-sided normal tail and the standardized even-power statistics use a joint upper normal tail. The maximum statistic uses

$$G = M^2 - 2\log(p) + \log\{\log(p)\}, \qquad P(G \le g) = \exp\{-\pi^{-1/2}\exp(-g/2)\}.$$

The odd, even, and maximum groups are asymptotically independent. If m non-empty groups were requested, the final p-value is $1 - \{1 - \min(P_O, P_E, P_\infty)\}^m$. Thus a subset of power families is combined with its actual group count rather than always using exponent 3.

Finite-power Gaussian moments are evaluated exactly from Isserlis pairings in C++, and multivariate normal rectangles are evaluated deterministically with `mvtnorm::Miwa`. This function deliberately implements no permutation or parametric-bootstrap calibration. `bandwidth` applies the paper's hard covariance band, setting entries with $|j - k|$ larger than the supplied width to zero. A banded estimate need not be positive semidefinite: `psd_adjust = "error"` rejects it, while `"eigen_clip"` performs and reports an explicit eigenvalue clipping repair. No silent ridge, absolute-value variance repair, or variance floor is used.

Value

An object of class `c("hd_location_test", "htest")`. `components` retains coordinate contrasts, all SPU statistics and individual p-values, Gaussian means/covariances/correlations, group statistics and p-values, and the maximum-statistic calibration. `diagnostics` records covariance, banding, PSD, numerical-integration, and group-combination choices.

References

Xu, G., Lin, L., Wei, P., and Pan, W. (2016). An adaptive two-sample test for high-dimensional means. *Biometrika*, **103**, 609–624. doi:10.1093/biomet/asw029

Examples

```
set.seed(83)
x <- matrix(rnorm(80), 20, 4)
y <- matrix(rnorm(88, 0.15), 22, 4)
xu_lin_wei_pan_aspu_test(x, y)
```

`yan_zhao_feng_weighted_max_test` *Yan–Zhao–Feng weighted max test*

Description

Tests $H_0 : \theta = \mu_0$ with the weighted max statistic of Yan, Zhao, and Feng. The full-sample weighted estimator is obtained with `weighted_scaled_spatial_median()` estimating equations. Define $\widehat{\zeta}_k = n^{-1} \sum_i \widehat{r}_i^k$. The uncentred and centred statistics are

$$M_m = n\|\widehat{D}^{-1/2}(\widehat{\theta} - \mu_0)\|_\infty^2 p(1 - n^{-1/2})\widehat{\zeta}_{m-1}^2/\widehat{\zeta}_{2m},$$

$$T_{MAX}^{(m)} = M_m - 2\log p + \log\log p.$$

Its feasible null cdf is $F(t) = \exp\{-\pi^{-1/2}\exp(-t/2)\}$. The returned upper-tail p-value is evaluated with stable exponential tails.

Usage

```
yan_zhao_feng_weighted_max_test(
  x,
  mu = NULL,
  m = -1,
  alpha = 0.05,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

`x`	A numeric matrix or data frame with observations in rows and at least two rows.
`mu`	A finite null-location vector. The default is zero.
`m`	Finite radial power no greater than one. The default is `-1`.
`alpha`	Significance level for the reported upper-tail rejection rule.
`tol`	Positive relative iteration tolerance.
`max_iter`	Positive maximum number of iterations.
`zero_tol`	Non-negative threshold at which a standardized residual is treated as zero. The default detects exact zeros only.
`strict`	If `TRUE`, non-stabilization is an error. If `FALSE`, the final iterate is returned with a warning and full diagnostics.

Value

An object of class `c("hd_location_test", "htest")`. Raw components include both max statistics, radial moments and their logs, the complete fitted estimator, and convergence/no-repair diagnostics.

References

Yan, G., Zhao, P., and Feng, L. (2025). Inverse norm weighted maxsum test for high dimensional location parameters. arXiv:2501.14168. https://arxiv.org/abs/2501.14168.

Examples

```
set.seed(2501)
x <- matrix(stats::rnorm(80), 10, 8)
yan_zhao_feng_weighted_max_test(x, m = 1, tol = 1e-6)
```

---

yan_zhao_feng_weighted_maxsum_test

Yan–Zhao–Feng weighted max-sum test

Description

Combines the weighted max test with the general weighted sum statistic

$$T_{SUM}^{(m)} = \frac{2}{n(n-1)} \sum_{i<j} r_{ij,i}^{m} r_{ij,j}^{m} U_{ij,i}^{\mathsf{T}} U_{ij,j}.$$

Here D_{ij} is the *unweighted* diagonal HR scale fitted after deleting observations i, j; endpoints are centred at the null μ_0, not at the fitted leave-two-out location.

Usage

```
yan_zhao_feng_weighted_maxsum_test(
  x,
  mu = NULL,
  m = -1,
  alpha = 0.05,
  tol = 1e-08,
  max_iter = 500L,
  zero_tol = 0,
  strict = TRUE
)
```

Arguments

x	A numeric matrix or data frame with observations in rows and at least two rows.
mu	A finite null-location vector. The default is zero.
m	Finite radial power no greater than one. The default is -1.
alpha	Significance level for the reported upper-tail rejection rule.
tol	Positive relative iteration tolerance.
max_iter	Positive maximum number of iterations.
zero_tol	Non-negative threshold at which a standardized residual is treated as zero. The default detects exact zeros only.
strict	If TRUE, non-stabilization is an error. If FALSE, the final iterate is returned with a warning and full diagnostics.

Details

The operational sum calibration is the direct feasible estimator

$$\widehat{\sigma}_m^2 = 2n^{-4} \sum_{i \neq j} r_{ij,i}^{2m} r_{ij,j}^{2m} \{(U_{ij,i} - \widetilde{\mu}_{ij})^\top U_{ij,j}\} \{(U_{ij,j} - \widetilde{\mu}_{ij})^\top U_{ij,i}\},$$

where $\widetilde{\mu}_{ij} = (n-2)^{-1} \sum_{k \neq i,j} U_{ij,k}$ is the unweighted, null-centred leave-two sign mean established in Section S.3 of the official Feng–Liu–Ma supplement. The sum and max upper-tail p-values are combined with equal Cauchy weights. At $m = -1$, every sum component reduces to `inst_one_sample_test()`.

Value

An object of class `c("hd_location_test", "htest")`. Its `components` field includes max, sum, direct variance, Cauchy, every leave-two kernel, fitted scales and locations, and radial quantities. `diagnostics` records all full/leave-out updates, residuals, zero counts, asymptotic scope, and the no-repair contract.

References

Yan, G., Zhao, P., and Feng, L. (2025). Inverse norm weighted maxsum test for high dimensional location parameters. arXiv:2501.14168. https://arxiv.org/abs/2501.14168.

Feng, L., Liu, B., and Ma, Y. (2020). Supplementary material for *An inverse norm sign test of location parameter for high-dimensional data*. doi:10.6084/m9.figshare.11914095.v2.

Examples

```
set.seed(2502)
x <- matrix(stats::rnorm(80), 10, 8)

yan_zhao_feng_weighted_maxsum_test(x, m = 0, tol = 1e-6)
```

`zhang_feng_radial_directional_test` *Zhang–Feng radial–directional test of an elliptical model*

Description

Tests the defining radial–directional implication of an elliptical model after a supplied or explicitly fitted affine standardisation. If $Y_i = \widehat{\Sigma}^{-1/2}(X_i - \widehat{\mu})$, $L_i = \log \|Y_i\|$, and $U_i = Y_i/\|Y_i\|$, the coordinate scores are the ordinary empirical correlations $\widehat{\gamma}_j$ between L_i and U_{ij}. The paper's statistics are

$$T_{\mathrm{sum}} = n \sum_j \widehat{\gamma}_j^2, \qquad T_{\max} = n \max_j \widehat{\gamma}_j^2 - 2\log p + \log\log p.$$

Their analytic p-values use $(T_{\mathrm{sum}} - p)/\sqrt{2p} \Rightarrow N(0,1)$ and $F_G(t) = \exp\{-\pi^{-1/2}\exp(-t/2)\}$. The adaptive p-value is the equal-weight Cauchy combination of the two marginal p-values.

Usage

```
zhang_feng_radial_directional_test(
  x,
  fit = NULL,
  location = NULL,
  shape = NULL,
  pilot_precision = NULL,
  bandwidth = NULL,
  component = c("combined", "sum", "max"),
  calibration = c("analytic", "radial_bootstrap"),
  B = 999L,
  seed = NULL,
  keep_bootstrap = FALSE,
  alpha = 0.05,
  tol = 1e-08,
  max_iter = 1000L,
  median_tol = 1e-08,
  median_max_iter = 1000L,
  zero_tol = 0,
  radius_zero_tol = 0,
  strict = TRUE
)
```

Arguments

`x`	Numeric $n \times p$ matrix with observations in rows. At least three observations and two variables are required.
`fit`	Optional valid object returned by `high_dimensional_hr()`.
`location, shape`	Optional jointly supplied location vector and symmetric positive-definite shape matrix. Shape scale is immaterial.

`pilot_precision`
: Optional explicit pilot precision passed to `high_dimensional_hr()` when neither `fit` nor `location`/`shape` is supplied.

`bandwidth`
: Optional hard-banding width for an internally fitted HR estimator. `NULL` delegates the dimension-aware paper default to `high_dimensional_hr()`.

`component`
: Which result is the main `htest`: adaptive Cauchy `"combined"`, dense `"sum"`, or sparse `"max"`. All three are always returned under `components`.

`calibration`
: Either the asymptotic `"analytic"` calibration or the paper's intrinsic `"radial_bootstrap"` mean–variance correction.

`B`
: Number of intrinsic bootstrap replicates; at least two when used.

`seed`
: Optional non-negative integer. An explicit seed is localized and does not change the caller's R RNG state. `NULL` uses the current stream normally.

`keep_bootstrap`
: Whether to retain the two bootstrap-statistic vectors.

`alpha`
: Test level strictly between zero and one.

`tol`, `max_iter`, `median_tol`, `median_max_iter`, `zero_tol`, `strict`
: Controls passed to an internally requested `high_dimensional_hr()` fit.

`radius_zero_tol`
: Non-negative relative fitted-radius threshold below one. A positive value rejects a minimum radius no larger than this fraction of the maximum radius. The relative definition preserves the arbitrary common scale of `shape`; zero rejects only exact coincidences. No ridge, eigenvalue floor, projection, pseudoinverse, or zero-radius perturbation is used.

Details

There are three deliberately exclusive standardisation routes. Supply a valid `high_dimensional_hr()` object through `fit`; supply both `location` and a symmetric positive-definite `shape`; or supply `pilot_precision` and let this function call `high_dimensional_hr()` with no hidden tuning. The latter route delegates `bandwidth`, convergence controls, and the strict no-repair contract to that estimator. The primary paper's numerical section adds a ridge and an incompletely specified positive-definite projection. Those simulation choices are not package defaults and are not reproduced here.

With `calibration = "radial_bootstrap"`, the function implements Section 2.4 of the primary paper: fitted radii are resampled with replacement and paired independently with uniform directions on the sphere. Bootstrap means and standard deviations correct the normal and Gumbel arguments exactly as displayed in the paper. This is an intrinsic calibration of the test, not a reproduction of the paper's simulation study. A degenerate resample or non-positive bootstrap standard deviation is reported without deletion, redrawing, or flooring.

The procedure tests the coordinatewise radial–directional correlation restrictions targeted by the paper. It is not an omnibus finite-sample test against every non-elliptical distribution. With a consistently transformed supplied shape, translation, global scale, and signed coordinate permutations are exact finite-sample invariances. A general affine transform induces an orthogonal rotation after standardisation, but the paper's coordinatewise empirical variance normalisation means that neither component is exactly rotation invariant in a finite sample. The null calibration is asymptotically rotation-compatible under uniform directions. The max diagnostic and an internally banded HR fit are also explicitly basis/order-sensitive.

Value

An object of class `c("radial_directional_test", "htest")` with the selected main p-value, all three analytic or bootstrap-calibrated component results, coordinate correlations and their largest coordinate, fitted log radii and directions, the complete standardisation source, and numerical diagnostics.

References

Zhang, H. and Feng, L. (2026). *High-Dimensional Tests for Elliptical Models via Radial–Directional Dependence*. arXiv:2605.03592. https://arxiv.org/abs/2605.03592

Examples

```
x <- rbind(
  c(2, 0), c(-2, 0), c(0, 1), c(0, -1),
  c(1, 2), c(-1, -2), c(2, -1), c(-2, 1)
)
zhang_feng_radial_directional_test(
  x, location = c(0, 0), shape = diag(2)
)
```

---

zhang_feng_rank_tests *Zhang–Feng adaptive marginal-rank location tests*

Description

Implements the one- and two-sample rank tests studied by Zhang and Feng (2024): a maximum of standardized marginal rank scores, the squared-rank sum statistic inherited from Ouyang et al. (2022), and their equal-weight Cauchy combination. Observations are rows and the ordered coordinates are columns.

Usage

```
zhang_feng_rank_one_sample_test(
  x,
  mu = NULL,
  component = c("max", "sum", "combined"),
  tau_sq = NULL,
  tau_method = NULL,
  lag = NULL
)

zhang_feng_rank_two_sample_test(
  x,
  y,
  component = c("max", "sum", "combined"),
```

```
  tau_sq = NULL,
  tau_method = NULL,
  lag = NULL
)
```

Arguments

x	A numeric matrix or data frame with observations in rows.
mu	For the one-sample test, a finite null location vector with one entry per column of x. NULL uses zero.
component	Which published statistic to return: "max" (the default), "sum", or "combined".
tau_sq	An optional, externally supplied positive long-run variance for the standardized squared-rank score sequence. It is required for "sum" and "combined" unless the explicit Ouyang–Parzen route is selected. It is not used for "max".
tau_method	NULL, "supplied", or "ouyang_parzen". For the latter, lag must be supplied explicitly. There is no default bandwidth.
lag	For tau_method = "ouyang_parzen", the integer lag-window size L in 1, ..., p. Lags 1 through $L-1$ are used.
y	For the two-sample test, a second numeric matrix or data frame with observations in rows and the same variables as x.

Details

For one sample, let U_j be the sum of the ranks of $|X_{ij} - \mu_j|$ having positive residuals. For two samples, U_j is the usual Mann–Whitney statistic obtained from the pooled ranks of the first sample. If e_j and v_j denote the exact untied-null mean and variance of U_j, the maximum component is

$$T_{\max} = \max_j (U_j - e_j)^2 / v_j - 2\log(p) + \log\log(p).$$

Its limiting distribution has cdf $G(y) = \exp\{-\pi^{-1/2}\exp(-y/2)\}$.

Write $M_j = (U_j - e_j)^2$, with exact untied-null mean $E_0(M_j)$ and variance $\mathrm{var}_0(M_j)$. The sum component is

$$T_{\mathrm{sum}} = \frac{\sqrt{p}\{p^{-1}\sum_j M_j - E_0(M_j)\}}{\sqrt{\mathrm{var}_0(M_j)\tau^2}},$$

and uses an upper-tail standard-normal calibration. The long-run variance τ^2 is never chosen silently. Supply a positive tau_sq, or set tau_method = "ouyang_parzen" and explicitly supply the lag-window size lag. The latter computes

$$\widehat{\tau}^2 = 1 + 2\sum_{k=1}^{L-1} w(k/L)\widehat{\gamma}(k), \qquad \widehat{\gamma}(k) = \frac{1}{p-k}\sum_{j=1}^{p-k}(R_j - \bar{R})(R_{j+k} - \bar{R}),$$

where $R_j = \{M_j - E_0(M_j)\}/\sqrt{\mathrm{var}_0(M_j)}$ and w is the Parzen kernel. lag is L, so lags 1 through $L-1$ are included. No automatic bandwidth rule, positivity floor, ridge, or other repair is applied.

With component p-values $p_{\max}$ and p_{sum}, the combined test uses

$$p_{\mathrm{C}} = 1 - F_{\mathrm{C}}\left\{\tfrac{1}{2}\cot(\pi p_{\max}) + \tfrac{1}{2}\cot(\pi p_{\mathrm{sum}})\right\},$$

evaluated with stable log-tail arithmetic. The paper relies on asymptotic independence of the maximum and sum components.

Exact zeros or tied absolute residuals are rejected by the one-sample function, and exact pooled ties are rejected by the two-sample function. This is deliberate: the displayed finite-sample moments and asymptotic calibrations are for continuous marginal distributions and Zhang and Feng do not specify a tie correction. The one-sample null additionally assumes coordinatewise symmetry about `mu`. The two-sample construction assumes a common pure-shift model (the two populations otherwise have the same continuous marginal distributions). Both sum calibrations assume that the coordinates, in their supplied order, form a sufficiently weakly dependent (strong-mixing) sequence. Consequently, an Ouyang–Parzen result may change if columns are permuted; column order is part of that estimator's contract.

The primary paper contains only the squared-rank sum, maximum, and Cauchy combination above. It does not define the general rank L_q family or a minimum-p adaptive test attributed to it in the accompanying book draft; those procedures are therefore not manufactured here.

Value

An object of classes `hd_location_test` and `htest`. Beyond the standard fields it retains all marginal rank scores, exact null moments, both component p-values when available, long-run-variance inputs and autocovariance diagnostics, and the stable Cauchy calculation. For the combined test, the finite primary `statistic` is the Cauchy angle $\arctan(T_{\mathrm{C}})/\pi$; the formal (possibly infinite) T_{C} is retained in `raw.statistic` and `components`.

References

Zhang, J. and Feng, L. (2024). Adaptive rank-based tests for high dimensional mean problems. *Statistics & Probability Letters*, **214**, 110226. doi:10.1016/j.spl.2024.110226.

Ouyang, M., Xie, Y., and Wang, W. (2022). A new test of high-dimensional mean vector with applications to gene set testing. *Computational Statistics & Data Analysis*, **171**, 107495. doi:10.1016/j.csda.2022.107495.

Examples

```
x <- matrix(c(
  -4, -1,  2,  5,
  -2,  3, -5,  1,
   1, -4,  3, -6,
   3,  5, -1,  2,
   6, -2,  7, -3
), nrow = 5, byrow = TRUE)
zhang_feng_rank_one_sample_test(x)
zhang_feng_rank_one_sample_test(x, component = "sum", tau_sq = 1)

y <- x + matrix(rep(c(0.4, -0.2, 0.3, 0.1), each = nrow(x)), ncol = 4)
zhang_feng_rank_two_sample_test(x, y)
```

zhang_zhou_guo_tests *Zhang–Zhou–Guo normal-reference mean tests*

Description

Implements the feasible one-sample normal-reference test of Zhang, Zhou, and Guo (2022), plus paired and same-unit linear-hypothesis wrappers. Observations are rows and variables are columns. For null-centred rows Z_i, their mean $\bar{Z}$, and their unbiased sample covariance S, the statistic is

$$T = n\|\bar{Z}\|^2 - \mathrm{tr}(S) = \frac{2}{n-1}\sum_{i<j} Z_i^{\mathsf{T}} Z_j.$$

This centring is essential. The quantity $n\|\bar{Z}\|^2$ by itself is only the motivating Gaussian oracle quadratic form; it is not the paper's feasible test statistic. This corrects the conflation in the corresponding short passage of the book draft.

Usage

```
zhang_zhou_guo_one_sample_test(x, mu = NULL, alpha = 0.05)

zhang_zhou_guo_paired_test(x, y, delta = NULL, alpha = 0.05)

zhang_zhou_guo_linear_hypothesis_test(x, L, rhs = NULL, alpha = 0.05)
```

Arguments

x	A numeric matrix or data frame with observations in rows. At least four rows are required.
mu	A finite null mean vector with one value per column of x. NULL uses the zero vector.
alpha	A finite significance level strictly between zero and one.
y	For the paired wrapper, a numeric matrix or data frame with exactly the same dimensions as x; row i must be paired with row i.
delta	A finite null paired mean-difference vector. NULL uses zero.
L	A finite numeric contrast matrix with one column per column of x. A numeric vector is treated as a one-row matrix. Rows define the transformed coordinates.
rhs	A finite vector with one value per row of L, giving the right- hand side of $L\mu = \mathrm{rhs}$. NULL uses zero.

Details

Write $v = n - 1$, $t_r = \mathrm{tr}(S^r)$, and $a_r = \mathrm{tr}(\Sigma^r)$. The exact Gaussian/Wishart unbiased trace estimators used here are

$$\widehat{a}_1 = t_1, \qquad \widehat{a}_2 = \frac{v^2}{(v-1)(v+2)}\left(t_2 - \frac{t_1^2}{v}\right),$$

$$\widehat{a}_3 = \frac{v^4}{(v-1)(v+4)(v^2-4)}\left(t_3 - \frac{3t_1t_2}{v} + \frac{2t_1^3}{v^2}\right).$$

In particular, the denominator is $v+4$, as in Appendix (A.33) of the primary paper, not $v+3$.

Under the Gaussian normal reference, the first three cumulants of the centred statistic are

$$\kappa_1 = 0, \qquad \kappa_2 = \frac{2n}{n-1}a_2, \qquad \kappa_3 = \frac{8n(n-2)}{(n-1)^2}a_3.$$

Matching these to $\beta_0 + \beta_1\chi^2_d$ gives

$$\widehat{\beta}_0 = -\frac{n}{n-2}\frac{\widehat{a}_2^2}{\widehat{a}_3}, \qquad \widehat{\beta}_1 = \frac{n-2}{n-1}\frac{\widehat{a}_3}{\widehat{a}_2}, \qquad \widehat{d} = \frac{n(n-1)}{(n-2)^2}\frac{\widehat{a}_2^3}{\widehat{a}_3^2}.$$

The reported `statistic` is the chi-square argument $(T - \widehat{\beta}_0)/\widehat{\beta}_1$; the centred T is retained in `raw.statistic` and `components`. The p-value is the upper tail of $\chi^2_{\widehat{d}}$.

The normal-reference approximation deliberately does not estimate the additional non-Gaussian third-cumulant term involving $\Upsilon = \mathrm{E}\{(Z_1^\mathsf{T}Z_2)^3\}$. Consequently it is not claimed to be a finite-sample exact calibration outside Gaussian data. At least four rows are needed. Non-positive $\widehat{a}_2$ or $\widehat{a}_3$ is a calibration failure: no ridge, absolute value, floor, or degrees-of-freedom clamp is applied.

`zhang_zhou_guo_paired_test()` applies the one-sample method to the paired rows $X_i - Y_i - \delta$; the two matrices must therefore describe the same observational units in the same row order. It is not an independent two-sample test. `zhang_zhou_guo_linear_hypothesis_test()` tests $H_0 : L\mu = \mathrm{rhs}$ by the same-unit row transform $Z_i = LX_i - \mathrm{rhs}$. It is not an independent-group MANOVA or a between-group general linear hypothesis procedure. The Euclidean metric after transformation is intentional, so changing `L` by a non-orthogonal row transformation generally changes the test.

The test is invariant to row permutations, orthogonal coordinate changes, and a common nonzero scalar change of units (with the null transformed in the same way). It is not invariant to arbitrary coordinatewise rescaling.

A common safe scale is removed before the compiled calculation. The chi-square argument, degrees of freedom, and p-value are invariant to that scale. Input-unit versions of T and the traces are returned when they are representable; otherwise their scaled versions and log scale remain available. The compiled kernel chooses a primal covariance calculation when the transformed dimension does not exceed n, and an observation- level dual Gram calculation otherwise.

Value

An object of class `c("hd_location_test", "htest")`. Besides the standard fields, `components` contains the centred statistic, motivating oracle quantity, raw and unbiased trace quantities, all three cumulants, matched chi-square parameters, critical values, and input-scale representability flags. `diagnostics` records the precise formulas, primal/dual route, wrapper transformation, normal-reference limitation, safe scaling, and no-repair contract.

References

Zhang, J.-T., Zhou, B., and Guo, J. (2022). Testing high-dimensional mean vector with applications. *Statistical Papers*, **63**, 1105–1137. doi:10.1007/s00362021012702.

Examples

```
x <- rbind(
  c(-1.2, 0.4, 1.1), c(0.3, -0.7, 0.5), c(1.4, 0.8, -0.2),
  c(-0.6, 1.5, 0.9), c(0.9, -1.1, 1.3), c(1.7, 0.2, -0.8)
)
zhang_zhou_guo_one_sample_test(x)

y <- x + rbind(
  c(0.1, -0.2, 0.3), c(-0.2, 0.1, -0.1), c(0.3, 0.2, -0.2),
  c(-0.1, -0.3, 0.2), c(0.2, 0.3, 0.1), c(-0.3, -0.1, -0.3)
)
zhang_zhou_guo_paired_test(x, y)
zhang_zhou_guo_linear_hypothesis_test(x, rbind(c(1, -1, 0)))
```

---

zhang_zhu_zhang_two_sample_test

Zhang–Zhu–Zhang normal-reference scale-invariant two-sample test

---

Description

Tests equality of two high-dimensional mean vectors when the covariance matrices may differ. Observations are rows and variables are columns. If S_1 and S_2 are the unbiased sample covariance matrices, let

$$\widehat{\Omega}_n = \frac{n_2}{n} S_1 + \frac{n_1}{n} S_2, \qquad \widehat{D}_n = \operatorname{diag}(\widehat{\Omega}_n).$$

Notice the crossed sample-size weights. The scale-invariant statistic is

$$T_{n,p} = \frac{n_1 n_2}{np} (\bar{X}_1 - \bar{X}_2)^{\mathsf{T}} \widehat{D}_n^{-1} (\bar{X}_1 - \bar{X}_2).$$

Usage

```
zhang_zhu_zhang_two_sample_test(
  x,
  y,
  alpha = 0.05,
  df_adjustment = c("paper", "none")
)
```

Arguments

x, y	Numeric matrices or data frames with observations in rows and the same variables in columns. Each sample must contain at least three observations.
alpha	Significance level for the upper-tail rejection rule.
df_adjustment	Either "paper" for the published conditional finite-sample degrees-of-freedom adjustment or "none" for the unadjusted Welch–Satterthwaite degrees of freedom.

Details

This is the scale-invariant test of Zhang, Zhu, and Zhang (2023). It is not the raw-L_2 normal-reference test based on $\|\bar{X}_1 - \bar{X}_2\|^2$, and it is not the later normal-reference F-type test. Its feasible reference distribution is the one-parameter Welch–Satterthwaite approximation χ^2_d/d.

Put $\widehat{R}_i = \widehat{D}_n^{-1/2} S_i \widehat{D}_n^{-1/2}$. The group-specific bias-corrected squared traces are

$$\widehat{q}_i = \frac{(n_i-1)^2}{(n_i-2)(n_i+1)} \left\{ \operatorname{tr}(\widehat{R}_i^2) - \frac{\operatorname{tr}^2(\widehat{R}_i)}{n_i - 1} \right\}.$$

The combined estimate and unadjusted degrees of freedom are

$$\widehat{q} = \frac{n_2^2}{n^2}\widehat{q}_1 + \frac{n_1^2}{n^2}\widehat{q}_2 + \frac{2n_1 n_2}{n^2} \operatorname{tr}(\widehat{R}_1 \widehat{R}_2), \qquad \widehat{d} = \frac{p^2}{\widehat{q}}.$$

These corrections apply to the trace estimate; the statistic $T_{n,p}$ itself is not bias-subtracted.

With `df_adjustment = "paper"`, the empirical finite-sample rule in the paper is also used. Define

$$c_{n,p} = 1 + \operatorname{tr}(\widehat{R}_n^2)/p^{3/2}, \qquad \widehat{R}_n = \frac{n_2}{n}\widehat{R}_1 + \frac{n_1}{n}\widehat{R}_2.$$

If $c_{n,p} \le 1.2$, the reference degrees of freedom are $\widehat{d}/c_{n,p}$; otherwise they remain $\widehat{d}$. The raw squared trace in this rule is deliberately not replaced by the bias-corrected $\widehat{q}$. `df_adjustment = "none"` always uses $\widehat{d}$. Both calibrations are returned regardless of the selected option.

The Gaussian oracle mixture has cumulants 1, $2p^{-2}\operatorname{tr}(R_n^2)$, and $8p^{-3}\operatorname{tr}(R_n^3)$. The third cumulant and the theoretical quantity $d^* = \operatorname{tr}^3(R_n^2)/\operatorname{tr}^2(R_n^3)$ are not estimated by the paper's feasible calibration and therefore are not substituted here.

The C++ kernel translates and rescales each coordinate by a common rule across both groups before computing moments. It uses a streamed primal covariance identity when $p \le n_1 + n_2$ and an observation-level dual Gram identity otherwise; neither route constructs a persistent $p \times p$ matrix. No ridge, pseudoinverse, absolute-value repair, trace floor, or degrees-of-freedom clamp is used. A nonpositive feasible trace estimate is an explicit calibration failure.

Value

An object of class `c("hd_location_test", "htest")`. The statistic is $T_{n,p}$ and the p-value is the upper tail of the selected χ^2_d/d reference. `components` contains both adjusted and unadjusted degrees of freedom, p-values, critical values and rejection decisions; raw and corrected trace components; diagonal standardisation; and coordinate contributions. `diagnostics` records the crossed weights, trace formulas, primal/dual route, empirical threshold decision, and no-repair contract. `log.p.value` in each calibration remains available when the ordinary tail underflows.

References

Zhang, L., Zhu, T., and Zhang, J.-T. (2023). Two-sample Behrens–Fisher problems for high-dimensional data: a normal reference scale-invariant test. *Journal of Applied Statistics*, **50**, 456–476. doi:10.1080/02664763.2020.1834516.

Examples

```
x <- matrix(c(
  0.2, -0.4, 1.1, 0.7,
  1.0,  0.3, 0.2, -0.8,
 -0.6,  1.2, 0.5, 0.1,
  0.4, -0.7, 1.3, 0.9
), ncol = 4, byrow = TRUE)
y <- matrix(c(
 -0.1, 0.5,  0.4, 1.0,
  0.8, 1.1, -0.5, 0.2,
 -0.7, 0.2,  1.0, 0.6,
  0.3, 0.9,  0.1, -0.4,
  1.1, -0.3, 0.8, 0.5
), ncol = 4, byrow = TRUE)
zhang_zhu_zhang_two_sample_test(x, y)
```

---

zhao_chen_wang_spatial_sign_white_noise_test

Zhao–Chen–Wang spatial-sign white-noise test

Description

Implements the spatial-sign sum statistic and feasible variance in Zhao, Chen and Wang. The primary null variance is `(H / 2) * tr(Omega^2)^2`.

Usage

```
zhao_chen_wang_spatial_sign_white_noise_test(
  x,
  lag = 1L,
  center = c("none", "mean"),
  zero_action = c("error", "keep"),
  keep_signs = FALSE
)
```

Arguments

`x`	Numeric matrix with time points in rows.
`lag`	Positive lag truncation level, no larger than `n - 2`.
`center`	`"none"` is the centered-at-zero primary definition; `"mean"` is an explicit sample-mean preprocessing extension.
`zero_action`	`"error"` rejects a zero residual direction. `"keep"` uses the paper's convention `U(0) = 0` and records the count.
`keep_signs`	Whether to retain the fitted spatial-sign matrix.

Value

An `htest` object with the raw statistic, trace estimate, variance, lag contributions, and zero-direction diagnostics.

References

Zhao, P., Chen, D. and Wang, Z. Spatial-sign-based high-dimensional white noises test. doi:10.1080/24754269.2024.2363715.

Examples

```
t <- seq_len(18)
x <- cbind(sin(t), cos(t / 2), sin(t / 3 + 0.2))
zhao_chen_wang_spatial_sign_white_noise_test(x, lag = 2)
```

---

zhao_chen_zi_inst_alpha_test

Zhao–Chen–Zi inverse-norm sign alpha test

Description

Implements the primary inverse-norm endpoint of the weighted spatial-sign alpha class. The spatial directions and diagonal scale come from restricted factor residuals $Y_t - \hat{B} f_t$. Following the displayed Chapter 4 formula, the inverse radial weights use unrestricted OLS residuals $Y_t - \hat{\alpha} - \hat{B} f_t$ and the same diagonal. The statistic sets $K(r) = r^{-1}$ in the weighted quadratic form and uses the audited leave-two-out estimate of $\mathrm{tr}(R^2)$.

Usage

```
zhao_chen_zi_inst_alpha_test(
  returns,
  factors = NULL,
  tol = 1e-08,
  max_iter = 1000L,
  zero_tol = 0,
  keep_scores = FALSE
)
```

Arguments

`returns`	Finite observation-by-asset numeric matrix or data frame.
`factors`	NULL, a finite length-T numeric vector, or a finite observation-by-factor numeric matrix or data frame.
`tol`	Positive diagonal fixed-point tolerance.
`max_iter`	Positive integer update limit.
`zero_tol`	Non-negative standardized zero-radius threshold; default zero uses the literal spatial-sign convention.
`keep_scores`	Whether to retain the fitted directions, radii, weights, and residual matrices.

Details

A zero unrestricted radius makes inverse weighting undefined and is an error. No radius floor, weight cap, absolute-value variance repair, ridge, or generalized inverse is applied. Arbitrary supplied weights belong to the explicitly named oracle interface `weighted_spatial_sign_alpha_oracle_test()`.

Value

An upper-tail alpha-test object with literal weighted components and nuisance-fit diagnostics.

References

Zhao, P., Chen, D. and Zi, X. (2022). High-dimensional non-parametric tests for linear asset pricing models. Stat 11, e490. doi:10.1002/sta4.490

Examples

```
f <- cbind(seq(-1, 1, length.out = 12))
y <- outer(seq_len(12), 1:3, function(i, j) sin(i + j / 3))
zhao_chen_zi_inst_alpha_test(y, f)
```

---

zhao_conditional_spatial_sign_sum_test

Zhao robust spatial-sign sum test for conditional alpha

Description

Implements the spatial-sign sum test using a restricted null sieve `fit` and the separate unrestricted, uncentered-alpha-basis `trace_fit` required by the primary feasible trace. With signs U_t from the restricted residuals and $h = M_Z 1_T$, its centered numerator is

$$(h'h)^{-1} h'UU'h - 1.$$

If $\widetilde{U}_t$ denotes the signs from `trace_fit`, then

$$\widehat{\mathrm{tr}(\Sigma_u^2)} = \frac{\sum_{t \neq s} h_t^2 h_s^2 (\widetilde{U}_t' \widetilde{U}_s)^2}{(h'h)(h'h - 1)}$$

and the statistic divides the centered numerator by the square root of this estimate.

Usage

```
zhao_conditional_spatial_sign_sum_test(fit, trace_fit)
```

Arguments

`fit`	Restricted centered-basis fit from `conditional_alpha_sieve_fit()`.
`trace_fit`	Fit to the same returns using the corresponding design from `conditional_alpha_sieve_design()` with `center_alpha = FALSE`.

Details

The book calls this a leave-two-out estimator but does not give the required unrestricted residual construction, and it inserts an additional factor of two under the square root. The primary Zhao formula, reproduced explicitly in Zhao and Wang (2026), has no such factor. Exact zero rows use the paper's definition $U(0) = 0$; a non-positive final trace still fails.

Value

A `conditional_alpha_test/htest` object with the normal upper-tail calibration, both sign matrices, and the exact trace numerator and denominator.

References

Zhao, P. (2023). Robust high-dimensional alpha test for conditional time-varying factor models. *Statistics*, 57, 444–457. doi:10.1080/02331888.2023.2180003.

Zhao, P. and Wang, H. (2026). Robust spatial-sign-based testing of high-dimensional alpha in conditional factor models. https://arxiv.org/abs/2604.12252.

Examples

```
tt <- seq(0, 1, length.out = 18)
b <- cbind(tt, tt^2)
y <- cbind(sin(1:18), cos(1:18), sin(1:18 / 2))
null_fit <- conditional_alpha_sieve_fit(
  y, conditional_alpha_sieve_design(b)
)
trace_fit <- conditional_alpha_sieve_fit(
  y, conditional_alpha_sieve_design(b, center_alpha = FALSE)
)
zhao_conditional_spatial_sign_sum_test(null_fit, trace_fit)
```

---

zhao_feng_strongcorr_sign_test

Zhao–Feng strong-correlation spatial-sign wild-bootstrap test

---

Description

Tests a one-sample location null with the wild-bootstrap calibration of Zhao and Feng. The observed raw statistic is

$$S_n = \sum_{1 \le i < j \le n} U(X_i - \mu_0)^{\mathsf{T}} U(X_j - \mu_0),$$

where $U(0) = 0$. Bootstrap signs are deliberately fitted differently: if $\widehat{\mu}$ is the ordinary Euclidean sample spatial median and $\widehat{U}_i = U(X_i - \widehat{\mu})$, then a replicate is

$$S^* = \sum_{i<j} e_i e_j \widehat{U}_i^{\mathsf{T}} \widehat{U}_j.$$

The multipliers are either independent Rademacher variables or independent standard Gaussian variables. The spatial median is fitted once and is not refitted inside the bootstrap.

Usage

```
zhao_feng_strongcorr_sign_test(
  x,
  mu = NULL,
  alpha = 0.05,
  multiplier = c("rademacher", "gaussian"),
  B = 9999L,
  seed = NULL,
  keep_bootstrap = FALSE,
  tol = 1e-08,
  max_iter = 1000L,
  strict = TRUE
)
```

Arguments

x	A finite numeric matrix or data frame with observations in rows. At least two observations and one variable are required.
mu	A finite null-location vector. The default is the zero vector.
alpha	A finite test level strictly between zero and one.
multiplier	Either "rademacher" or "gaussian".
B	A positive integer number of wild-bootstrap replicates.
seed	NULL or an integer from zero through R's integer limit. An explicit seed makes the bootstrap reproducible without changing the caller's RNG state. With NULL, one seed is first drawn from R's RNG and recorded, after which the replicate generation is isolated locally.
keep_bootstrap	Whether to retain all raw and pair-normalised bootstrap statistics.
tol	A finite positive tolerance passed to spatial_median().
max_iter	A positive integer iteration limit for the spatial median.
strict	Whether failure of the spatial-median iteration is an error (TRUE) or a warning followed by use of its last finite iterate (FALSE).

Details

The paper writes both statistics divided by $\sqrt{\tau}\sqrt{\binom{n}{2}}$, where $\tau = \mathrm{tr}(\Sigma_U^2)$, but explicitly notes that τ need not be estimated because this common factor cancels from the bootstrap comparison. Accordingly, the test uses raw pair sums and also reports their $\sqrt{\binom{n}{2}}$-normalised, tau-free versions. It does not invent a plug-in estimate of τ.

The paper prescribes the empirical $(1-\alpha)$ quantile and rejects when the observed statistic is strictly greater. For reproducibility this implementation defines that quantile as the type-1 inverse empirical cdf: order statistic $\lceil (1-\alpha)B \rceil$. The ordinary p.value uses the separately labelled finite-Monte-Carlo plus-one convention

$$\{1 + \#(S_b^* \geq S_n)\}/(B+1),$$

with ties counted in the upper tail. Both rejection decisions are returned because they can differ at finite B.

The pair-sum identity used computationally is $\{\|\sum_i U_i\|^2 - \sum_i \|U_i\|^2\}/2$. Thus an observation exactly equal to a centre remains the literal zero sign; the implementation does not silently replace the diagonal term by n, add jitter, or repair a degenerate bootstrap distribution. A degenerate distribution is returned with an explicit diagnostic.

The test is invariant to a common translation (when `mu` is translated), orthogonal transformations, and a common nonzero scalar transformation. It is not coordinatewise scale invariant. `strict = TRUE` makes failure of the ordinary spatial-median iteration an error. With `strict = FALSE`, the last finite iterate is used with a warning and its diagnostics are retained. No ridge, floor, perturbation, or pseudoinverse is applied.

Value

An object of class `c("hd_location_test", "htest")`. Its `p.value` is the auxiliary plus-one Monte Carlo upper-tail probability. `components` contains the raw and tau-free statistics, the paper type-1 critical rule, both decisions, null-centred and fitted signs, the fitted ordinary spatial median, and optional bootstrap draws. `diagnostics` records convergence, zero residuals, overflow-safe residual subtraction, Monte Carlo conventions, and the no-repair contract.

References

Zhao, P. and Feng, L. (2026). Note on High Dimensional Spatial-Sign Test for One Sample Problem. arXiv:2601.08736. doi:10.48550/arXiv.2601.08736.

Examples

```
x <- rbind(c(-1.0, 0.4), c(0.2, -0.8), c(1.3, 0.6),
           c(-0.4, 1.1), c(0.8, -0.2))
zhao_feng_strongcorr_sign_test(x, B = 99, seed = 2601)
```

`zhao_feng_wang_wang_robust_alpha_test`

Zhao–Feng–Wang–Wang robust maximum and adaptive alpha test

Description

Fits the simultaneous scaled spatial median to the restricted factor residuals and computes the primary robust maximum statistic. Let r denote the fitted standardized radii and $\eta = T^{-1}1'P_F1$. The implemented nuisance constant is

$$\widehat{\zeta} = \frac{N\{\overline{r^{-1}}\}^2}{1 - 2\eta\overline{r^{-1}}\,\bar{r} + \eta\overline{r^{-2}}\,\overline{r^2}}.$$

The centered maximum is

$$T\widehat{\zeta}\|\widehat{D}^{-1/2}\widehat{\theta}\|_\infty^2 - 2\log N + \log\log N,$$

calibrated by the upper tail of $\exp\{-\pi^{-1/2}\exp(-x/2)\}$.

Usage

```
zhao_feng_wang_wang_robust_alpha_test(
  returns,
  factors = NULL,
  component = c("max", "combined"),
  bias = c("wild_bootstrap", "none", "supplied"),
  delta_q = NULL,
  bootstrap_reps = 100L,
  seed = NULL,
  keep_bootstrap = FALSE,
  tol = 1e-07,
  max_iter = 500L,
  zero_tol = 0
)
```

Arguments

returns	Finite observation-by-asset numeric matrix or data frame.
factors	NULL, a finite length-T numeric vector, or a finite observation-by-factor numeric matrix or data frame.
component	"max" for the robust maximum alone or "combined" for the primary truncated-Cauchy max/sum procedure.
bias	Bias calibration: primary "wild_bootstrap", literal zero "none" for formula auditing, or a user "supplied" value.
delta_q	Finite supplied bias when bias is "supplied"; otherwise must be NULL.
bootstrap_reps	Positive number of Rademacher replicates. The primary recommendation is 100.
seed	NULL to use and advance the current random stream, or a non-negative integer for an isolated deterministic bootstrap.
keep_bootstrap	Whether to retain all bootstrap Q values.
tol	Positive diagonal fixed-point tolerance.
max_iter	Positive integer update limit.
zero_tol	Non-negative standardized zero-radius threshold; default zero uses the literal spatial-sign convention.

Details

The book draft reverses the squared inverse-radius moment in zeta and replaces the primary final coefficient eta by eta squared. Both changes alter the statistic; this implementation uses the primary formula and returns every empirical radial moment. For component "combined", the robust maximum p-value and `liu_feng_ma_spatial_sign_alpha_test()` p-value are combined by the paper's truncated Cauchy rule: only component p-values below one half contribute. This is distinct from the untruncated generic Cauchy construction introduced elsewhere in the book.

Value

An upper-tail alpha-test object containing the complete scaled spatial-median fit, radial moments, corrected zeta, maximum calibration, and, for the combined procedure, the full LFM component.

References

Zhao, P., Feng, L., Wang, Z., and Wang, X. Robust high-dimensional alpha testing for linear factor pricing models. Oxford Bulletin of Economics and Statistics (2026). doi:10.1111/obes.70080. Preprint: https://arxiv.org/abs/2408.06612.

Examples

```
f <- cbind(seq(-1, 1, length.out = 12))
y <- outer(seq_len(12), 1:3, function(i, j) cos(i + j / 3))
zhao_feng_wang_wang_robust_alpha_test(y, f)
```

`zhao_wang_conditional_spatial_sign_test`
Zhao–Wang robust conditional maximum and adaptive alpha test

Description

Implements the CSM maximum of Zhao and Wang (2026) on a restricted conditional sieve fit. A simultaneous scaled spatial median gives $\widehat{\theta}$, $\widehat{D}$, and standardized radii. With $\omega_T = h'h$, the primary nuisance estimate is

$$\widehat{\zeta} = \frac{N\overline{r^{-1}}^2}{1 - 2(1 - \omega_T/T)\overline{r^{-1}}\bar{r} + (1 - \omega_T/T)\overline{r^2}\,\overline{r^{-1}}^2},$$

and the centered statistic is

$$T\widehat{\zeta}\|\widehat{D}^{-1/2}\widehat{\theta}\|_\infty^2 - 2\log N + \log\log N.$$

Usage

```
zhao_wang_conditional_spatial_sign_test(
  fit,
  trace_fit = NULL,
  component = c("max", "sum", "combined"),
  tol = 1e-07,
  max_iter = 500L,
  zero_tol = 0
)
```

Arguments

`fit`	Restricted centered-basis conditional sieve fit.
`trace_fit`	For `"sum"` or `"combined"`, the corresponding uncentered-alpha-basis fit required by CSS.
`component`	One of `"max"`, `"sum"`, or `"combined"`.
`tol`	Positive simultaneous scaled-median equation tolerance.
`max_iter`	Positive maximum update count.
`zero_tol`	Non-negative exact-zero radius tolerance.

Details

For `component = "combined"`, the CSM p-value and the exact Zhao CSS component from `zhao_conditional_spatial_sig` are combined by the paper's *truncated* Cauchy rule: a component contributes only when its p-value is below one half. This current primary source resolves the book's placeholder `ZhaoWang2024ConditionalMaxAlpha` citation. The book's CSM radial formula is otherwise consistent, while its CSS square-root factor is not.

Value

A `conditional_alpha_test/htest` object containing the full CSM fit, radial correction, CSS component, and truncated-Cauchy diagnostics.

References

Zhao, P. and Wang, H. (2026). Robust spatial-sign-based testing of high-dimensional alpha in conditional factor models. https://arxiv.org/abs/2604.12252.

Examples

```
tt <- seq(0, 1, length.out = 18)
b <- cbind(tt, tt^2)
y <- cbind(sin(1:18), cos(1:18), sin(1:18 / 2))
fit <- conditional_alpha_sieve_fit(
  y, conditional_alpha_sieve_design(b)
)
zhao_wang_conditional_spatial_sign_test(fit)
```

---

zhao_yang_zhang_feng_wang_adaptive_sphericity_test

Adaptive dense–sparse elliptical sphericity test

Description

Combines the feasible spatial-sign sum p-value (`p.SS`) with the spatial-sign max p-value (`p.SM`) using the truncated Cauchy rule of Zhao et al. (2026). The spatial median and fitted signs are computed once. The returned `Cauchy.score` is the inner non-negative score,

$$\frac{1}{2}\tan\{\pi(1/2 - p_{SS})\}I(p_{SS} < 1/2) + \frac{1}{2}\tan\{\pi(1/2 - p_{SM})\}I(p_{SM} < 1/2),$$

while `p.value` is the distinct final quantity $1 - F_C(\text{Cauchy.score})$. If neither component p-value is below one half, the score is zero and the combined p-value is exactly one half.

Usage

```
zhao_yang_zhang_feng_wang_adaptive_sphericity_test(
  x,
  alpha = 0.05,
  tol = 1e-08,
  max_iter = 1000L,
  strict = TRUE,
  keep_sscm = FALSE
)
```

Arguments

`x`	A finite numeric matrix or data frame with observations in rows; at least two rows and two columns are required.
`alpha`	A finite test level strictly between zero and one.
`tol`, `max_iter`	Convergence controls passed to `spatial_median()`.
`strict`	If `TRUE`, spatial-median nonconvergence is an error. If `FALSE`, the last finite iterate is used with a warning.
`keep_sscm`	Whether to retain the full fitted spatial-sign covariance matrix. Its diagonal and maximizing entries are always returned.

Details

This function uses the translation-invariant residual inverse-moment estimate of the spatial-sign sum bias specified by the adaptive-test primary source. An exact fitted zero residual makes that feasible bias undefined and causes an explicit error. No floor, deletion, ridge, or perturbation is applied.

Value

An object of class `c("hd_sphericity_test", "htest")`. The scalar `statistic` is the Cauchy score and `p.value` is the final combined probability. `components$sum` and `components$max` retain both complete component calibrations.

References

Zhao, P., Yang, F., Zhang, X., Feng, L., and Wang, Z. (2026). Adaptive tests for high-dimensional sphericity under different distribution types. *Journal of Multivariate Analysis*, 214, 105634. doi:10.1016/j.jmva.2026.105634.

Examples

```
x <- rbind(c(-2, 0), c(-1, 1), c(0, -2), c(1, 2), c(3, -1), c(2, 1))
zhao_yang_zhang_feng_wang_adaptive_sphericity_test(x)
```

`zhao_yang_zhang_feng_wang_sign_max_test`
Zhao–Yang–Zhang–Feng–Wang spatial-sign max sphericity test

Description

Computes the sparse-alternative max statistic from the fitted spatial-sign covariance matrix $\widehat{\Omega} = (\widehat{\psi}_{ij})$. The diagonal and off-diagonal entries are standardized with their distinct null variances, maximized, and centered by $-2\log\{p(p+1)/2\} + \log\log\{p(p+1)/2\}$. The upper-tail limiting cdf is $G(t) = \exp\{-\pi^{-1/2}\exp(-t/2)\}$.

Usage

```
zhao_yang_zhang_feng_wang_sign_max_test(
  x,
  alpha = 0.05,
  tol = 1e-08,
  max_iter = 1000L,
  strict = TRUE,
  keep_sscm = FALSE
)
```

Arguments

`x`	A finite numeric matrix or data frame with observations in rows; at least two rows and two columns are required.
`alpha`	A finite test level strictly between zero and one.
`tol`, `max_iter`	Convergence controls passed to `spatial_median()`.
`strict`	If `TRUE`, spatial-median nonconvergence is an error. If `FALSE`, the last finite iterate is used with a warning.
`keep_sscm`	Whether to retain the full fitted spatial-sign covariance matrix. Its diagonal and maximizing entries are always returned.

Details

Exact zero residuals retain `U(0) = 0`, so the empirical SSCM can have trace below one; the count and trace are returned. `keep_sscm = FALSE` avoids allocating a $p \times p$ return matrix while still computing the exact maximum. No covariance regularization or numerical repair is performed.

Value

An object of class `c("hd_sphericity_test", "htest")` with the Gumbel statistic, upper-tail p-value, fitted signs, SSCM diagnostics, and exact maximizing entry.

References

Zhao, P., Yang, F., Zhang, X., Feng, L., and Wang, Z. (2026). Adaptive tests for high-dimensional sphericity under different distribution types. *Journal of Multivariate Analysis*, 214, 105634. doi:10.1016/j.jmva.2026.105634.

Examples

```
x <- rbind(c(-2, 0), c(-1, 1), c(0, -2), c(1, 2), c(3, -1), c(2, 1))
zhao_yang_zhang_feng_wang_sign_max_test(x)
```

`zou_peng_feng_wang_sphericity_test`

Zou–Peng–Feng–Wang spatial-sign sphericity test

Description

Tests sphericity of an elliptical distribution with the bias-corrected spatial-sign sum statistic of Zou et al. (2014). Rows of `x` are observations. The raw statistic is

$$\widetilde{Q} = \frac{p}{n(n-1)} \sum_{i \neq j} (\widehat{U}_i^{\mathsf{T}} \widehat{U}_j)^2 - 1,$$

where $\widehat{U}_i = U(X_i - \widehat{\theta})$ and $\widehat{\theta}$ is the spatial median. The standardized statistic is $(\widetilde{Q} - p\widehat{\delta})/\sigma_0$, with $\sigma_0^2 = 4(p-1)/\{n(n-1)(p+2)\}$.

Usage

```
zou_peng_feng_wang_sphericity_test(
  x,
  alpha = 0.05,
  bias_estimator = c("residual", "second_order", "normal_limit"),
  tol = 1e-08,
  max_iter = 1000L,
  strict = TRUE
)
```

Arguments

`x`	A finite numeric matrix or data frame with observations in rows; at least two rows and two columns are required.
`alpha`	A finite test level strictly between zero and one.
`bias_estimator`	Feasible bias calibration: `"residual"`, the literal 2014 `"second_order"` prescription, or `"normal_limit"`.
`tol`, `max_iter`	Convergence controls passed to `spatial_median()`.
`strict`	If `TRUE`, spatial-median nonconvergence is an error. If `FALSE`, the last finite iterate is used with a warning.

Details

`bias_estimator = "residual"` uses the translation-invariant inverse residual moments adopted for the feasible sign-sum component in Zhao et al. (2026), and is the practical default. `"second_order"` reproduces the origin-dependent corrected-radius prescription in Zou et al. (2014), after that paper sets the unknown population centre to zero. It is retained for formula auditing and is not silently presented as translation invariant. `"normal_limit"` uses their large-dimensional shortcut $\widehat{\delta} = n^{-2} + 2n^{-3}$. Exact zero residuals make inverse moments undefined and are rejected unless the normal-limit shortcut is requested. No ridge, deletion, absolute value, or numerical floor is used.

Value

An object of class `c("hd_sphericity_test", "htest")`. Raw components include the fitted centre and signs, radii, inverse-moment ratios, feasible bias, null variance, and convergence diagnostics.

References

Zou, C., Peng, L., Feng, L., and Wang, Z. (2014). Multivariate-sign-based high-dimensional tests for sphericity. *Biometrika*, 101, 229–236. doi:10.1093/biomet/ast040.

Examples

```
x <- rbind(c(-2, 0), c(-1, 1), c(0, -2), c(1, 2), c(3, -1), c(2, 1))
zou_peng_feng_wang_sphericity_test(x)
```

Index

acg_loglik, 6

bai_saranadasa_two_sample_test, 6
basic_shape, 7
basic_shape(), *9*
basics_shape, 9
bickel_levina_covariance_threshold, 10
book_gaussian_alpha_cauchy_test, 11

cai_liu_adaptive_covariance_threshold, 13
cai_liu_xia_two_sample_test, 14
cca_bartlett_test, 16
chen_qin_two_sample_test, 17
chen_song_feng_rank_white_noise_test, 18
chen_zhang_zhong_covariance_test, 19
cheng_sscm_equality_test, 20
chime_clustering, 21
classical_cca, 23
classical_cusum_test, 24
classical_lda_classifier, 26
classical_pca, 27
classical_qda_classifier, 28
classical_spatial_tests, 29
clime_precision, 31
composite_t2_two_sample_test, 32
conditional_alpha_sieve_design, 34
conditional_alpha_sieve_design(), *35*, *114*, *233*
conditional_alpha_sieve_fit, 35
conditional_alpha_sieve_fit(), *114, 115*, *233*
conditional_factor_wald_test, 36

dsda_classifier, 37

ec2_covariance, 38
elliptical_factor_number, 40
elliptical_factor_precision, 41
elliptical_oracle_classifier, 42
elliptical_regularized_hotelling_cauchy_test, 43
elliptical_regularized_hotelling_test, 45
erht_change_point_test, 46
erht_wbs, 48

fair_classifier, 50
fantope_pca, 52
feng_jiang_liu_xiong_panel_independence_test, 53
feng_jiang_liu_xiong_panel_independence_test(), *202*
feng_lan_liu_ma_alpha_max_test, 54
feng_liu_ma_white_noise_test, 55
feng_liu_rank_sphericity_test, 56
feng_spatial_rank_proportionality_test, 57
feng_sun_one_sample_test, 59
feng_sun_one_sample_test(), *175*
feng_wang_pdq_two_sample_test, 60
feng_zhang_liu_spatial_rank_test, 63
feng_zou_wang_two_sample_sign_test, 65
feng_zou_wang_zhu_two_sample_test, 67
fisher_sun_gallagher_sphericity_test, 68

gaussian_alpha_combination_test, 69
gaussian_alpha_combination_test(), *12*
gaussian_covariance_lrt, 70
gaussian_graphical_lasso, 71
gaussian_lda_oracle, 72
gaussian_mixture_em, 73
gaussian_qda_oracle, 75
gaussian_wilks_independence_test, 76
generalized_sign_pca, 77
generic_weighted_hr_location, 79
gqda_classifier, 81
grs_alpha_test, 82

hallin_paindaveine_shape_test, 83
hd_classifier_fit, 84
high_dimensional_hr, 85
high_dimensional_hr(), *91*, *222*, *223*
hotelling_one_sample_test (hotelling_tests), 88
hotelling_tests, 88
hotelling_two_sample_test (hotelling_tests), 88
hr_estimator, 89
hr_gqda, 90
hr_gqda(), *91*
hr_qda (hr_gqda), 90

if_pca, 92
independence_classifier, 94
inst_one_sample_test, 95

jiang_da_qda, 97
john_sphericity_test, 98

k_spatial_median, 99
kendall_factor_number, 100
kendall_pca, 101

li_chen_covariance_test, 102
li_shao_sparse_qda, 103
li_wang_zou_two_sample_sign_test, 105
linear_pool_covariance, 107
liu_feng_ma_spatial_sign_alpha_test, 109
liu_feng_ma_spatial_sign_alpha_test(), *237*
lloyd_kmeans, 111
lpd_classifier, 112

ma_feng_wang_bao_conditional_alpha_test, 114
ma_lan_su_tsai_conditional_alpha_sum_test, 115
ma_lan_su_tsai_conditional_alpha_sum_test(), *114*
mauchly_sphericity_test, 116

nagao_identity_test, 117
normalize_shape, 118

ollila_raninen_shrinkage_covariance, 118
ollila_raninen_shrinkage_covariance(), *108*
oracle_weighted_sign_sum_test, 120

park_ayyala_one_sample_test, 121
pesaran_cd_test, 123
pesaran_yamagata_alpha_test, 124
pmd_sparse_cca, 125
pmd_sparse_pca, 126
poet_covariance, 128
poet_tme, 129
predict.hd_classifier_fit, 130
predict.semc_fit, 131

regularized_spatial_sign_covariance, 132
relliptical, 133
relliptical(), *138*
robust_factor_subspace, 134
robust_gqda, 135
rothman_levina_zhu_covariance_threshold, 136
rspherical, 138
rts_factor, 138

scaled_spatial_median, 140
scaled_spatial_median(), *175*
sdar_qda, 141
semc_fit, 142
semc_select_k_gap, 147
shao_threshold_lda, 149
sign_whitened_sparse_cca, 150
sm_sscm, 152
sparse_k_spatial_median, 153
sparse_kmeans, 155
sparse_kmeans_select_s, 156
sparse_kmedian, 158
sparse_plugin_lda, 160
sparse_plugin_qda, 161
sparse_sm_select_k, 162
sparse_sm_select_tau, 164
sparse_spatial_sign_pca, 166
spatial_kendall, 168
spatial_kendall(), *101*
spatial_median, 169
spatial_median(), *119*, *235*, *240–242*
spatial_rank, 170
spatial_rank_covariance, 170

spatial_rank_test
(classical_spatial_tests), 29
spatial_sign, 171
spatial_sign_change_point_test, 172
spatial_sign_max_test, 174
spatial_sign_max_test(), *175*
spatial_sign_maxsum_test, 175
spatial_sign_pca, 177
spatial_sign_pca(), *166*
spatial_sign_poet, 178
spatial_sign_precision, 179
spatial_sign_precision(), *86*
spatial_sign_precision_lda, 181
spatial_sign_sphericity_test, 183
spatial_sign_test
(classical_spatial_tests), 29
spatial_signed_rank_test
(classical_spatial_tests), 29
srivastava_du_one_sample_test, 184
srivastava_katayama_kano_two_sample_test,
185
sscca, 186
sscca(), *151*
sscm, 188
sscm(), *177*
sslda, 189
ssqda, 190

tensor_spatial_sign_precision, 192, *195*
threshold_spatial_sign_precision, 194
threshold_tensor_spatial_sign_precision,
195
tinst_two_sample_test, 196
truncated_power_pca, 197
tyler_shape, 199

wang_feng_dms_test, 200
wang_liu_feng_ma_serial_panel_test,
201
wang_liu_feng_vector_independence_test,
202
wang_liu_feng_vector_u_independence_test,
204
wang_peng_li_one_sample_test, 205
wang_xu_approx_randomization_test, 207
wang_yao_corrected_john_test, 209
wang_yao_corrected_lrt, 210
wang_zhao_feng_wang_mutual_fund_fdr,
211
weighted_scaled_spatial_median, 213
weighted_spatial_sign_alpha_oracle_test,
214
white_noise_portmanteau_test, 215

xu_lin_wei_pan_aspu_test, 216

yan_zhao_feng_weighted_max_test, 219
yan_zhao_feng_weighted_maxsum_test,
220

zhang_feng_radial_directional_test,
222
zhang_feng_rank_one_sample_test
(zhang_feng_rank_tests), 224
zhang_feng_rank_tests, 224
zhang_feng_rank_two_sample_test
(zhang_feng_rank_tests), 224
zhang_zhou_guo_linear_hypothesis_test
(zhang_zhou_guo_tests), 227
zhang_zhou_guo_one_sample_test
(zhang_zhou_guo_tests), 227
zhang_zhou_guo_paired_test
(zhang_zhou_guo_tests), 227
zhang_zhou_guo_tests, 227
zhang_zhu_zhang_two_sample_test, 229
zhao_chen_wang_spatial_sign_white_noise_test,
231
zhao_chen_zi_inst_alpha_test, 232
zhao_conditional_spatial_sign_sum_test,
233
zhao_conditional_spatial_sign_sum_test(),
239
zhao_feng_strongcorr_sign_test, 234
zhao_feng_wang_wang_robust_alpha_test,
236
zhao_wang_conditional_spatial_sign_test,
238
zhao_yang_zhang_feng_wang_adaptive_sphericity_test,
239
zhao_yang_zhang_feng_wang_sign_max_test,
241
zou_peng_feng_wang_sphericity_test,
242