Package 'dacc'

November 9, 2026

Title Detection and Attribution Analysis of Climate Change

Version 0.1.0

Date 2026-11-08

Description Detection and attribution analysis of climate change using errors-in-variables regression. Implements the polynomial shrinkage framework under variance heterogeneity proposed by Li and Li (2026) <doi:10.48550/arXiv.2608.12496> and the latent factor framework proposed by Li and Li (2026) <doi:10.48550/arXiv.2609.13488>. Estimates forcing scaling factors and variance inflation coefficients for forced simulations and control runs relative to observations. Selects polynomial covariance weights or bias-corrected spiked precision weights by minimizing estimated total asymptotic scaling factor variance, and provides asymptotic confidence intervals and residual diagnostics. Also retains earlier total least squares and estimating equation procedures (Li et al., 2025, <doi:10.1175/JCLI-D-24-0193.1>; Ma et al., 2023, <doi:10.1175/JCLI-D-22-0681.1>) for comparison and reproducibility, together with covariance shrinkage estimators.

Depends R (>= 3.5.0)

Imports MASS, stats, graphics, utils, Iso, pracma, janitor, magrittr, sp, CFtime, ncdf4

Suggests RMTstat

License GPL (>= 3)

Encoding UTF-8

URL https://github.com/LiYanStat/dacc

BugReports https://github.com/LiYanStat/dacc/issues

Repository CRAN

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Haoran Li [aut],
Yan Li [aut, cre],
Kun Chen [aut],
Jun Yan [aut]

Maintainer Yan Li <yan.4.li@uconn.edu>

Date/Publication 2026-11-09 00:10:19 UTC

Contents

Covest 2
fingerprint 3
fingerprintLatent 6
fingerprintLegacy 11
fingerprintPoly 14
fpPrep 18
globalDat 19
plot.fingerprint 20
simDat 21

Index **23**

`Covest` *Regularized estimators for covariance matrix.*

Description

This function estimate the covariance matrix under l2 loss and minimum variance loss, provide linear shrinkage estimator under l2 loss and nonlinear shrinkage estimator under minimum variance loss.

Usage

```
Covest(Z, method = c("mv", "l2"), bandwidth = NULL)
```

Arguments

`Z`	n*p matirx with sample size n and dimension p. Replicates for computing the covariance matrix, should be centered.
`method`	methods used for estimating the covariance matrix.
`bandwidth`	bandwidth for the "mv" estimator, default value are set to be list in (0.2, 0.5).

Value

regularized estimate of covariance matrix.

Author(s)

Yan Li

References

- Olivier Ledoit and Michael Wolf (2004), A well-conditioned estimator for large-dimensional covariance matrices, *Journal of multivariate analysis*, 88(2), 365–411.
- Olivier Ledoit and Michael Wolf (2017), Direct nonlinear shrinkage estimation of large-dimensional covariance matrices, *Working Paper No. 264, UZH*.
- Li et al (2023), Regularized fingerprinting in detection and attribution of climate change with weight matrix optimizing the efficiency in scaling factor estimation, *Ann. Appl. Stat.* 17(1), 225–239.

Examples

```
## randomly generate a n * p matrix where n = 50, p = 100
Z <- matrix(rnorm(50 * 100), nrow = 50, 100)
## linear shrinkage estimator under l2 loss
Cov.est <- Covest(Z, method = "l2")$output
## nonlinear shrinkage estimator under minimum variance loss
Cov.est <- Covest(Z, method = "mv", bandwidth = 0.35)$output
```

`fingerprint` *Detection and attribution of climate change*

Description

Common entry point for the detection and attribution methods in dacc. Select polynomial shrinkage under variance heterogeneity, latent-factor fingerprinting, estimating equations, or two-sample and bootstrap TLS procedures.

Usage

```
fingerprint(
    Xtilde,
    Y,
    mruns,
    ctlruns,
    method = c(
        "poly", "linear", "latent", "EE", "PBC", "TS", "TS.TempSt"
    ),
    ...
)
```

Arguments

`Xtilde`	Numeric matrix of forcing ensemble means, one forcing per column.
`Y`	Observation vector or matrix as required by the selected method.
`mruns`	Ensemble sizes, one per forcing.
`ctlruns`	Control runs, with one run per row.

`method` Estimation method: "poly" (default), "linear", "latent", "EE", "PBC", "TS", or "TS.TempSt".

`...` Arguments for the selected implementation. See fingerprintPoly, fingerprintLatent or fingerprintLegacy. In particular, supply ensembles when estimating variance inflation. For latent selection, use factor.method (PLY or KN) or override the count with n.factors.

Details

The `method` argument selects both the scaling-factor estimator and its uncertainty procedure. Method-specific arguments are passed through ....

`"poly"` Polynomial-shrinkage fingerprinting proposed by Li and Li (2026a) in *Adaptable Fingerprinting with Nonlinear Shrinkage for Climate Change Detection and Attribution under Variance Heterogeneity*.

A polynomial transformation of the control covariance supplies the covariance weight for weighted total least squares (TLS). Polynomial coefficients minimize the estimated total asymptotic variance of the scaling-factor estimator; `degree` defaults to 3. Variance inflation coefficients are estimated by default. Marginal confidence intervals use the estimated asymptotic covariance.

`"linear"` Linear-shrinkage fingerprinting obtained as the degree-one specialization of the method proposed by Li and Li (2026a) in *Adaptable Fingerprinting with Nonlinear Shrinkage for Climate Change Detection and Attribution under Variance Heterogeneity*.

A degree-one polynomial supplies the covariance weight for weighted TLS. Its coefficient minimizes the estimated total asymptotic scaling-factor variance; any supplied `degree` is overridden with 1. Variance inflation coefficients are estimated by default. Marginal confidence intervals use the same asymptotic covariance construction as `"poly"`. This weight-selection criterion differs from the covariance-loss criterion of Ledoit and Wolf (2004).

`"latent"` Latent-factor fingerprinting proposed by Li and Li (2026b) in *Fingerprint Analysis for Climate Change Detection and Attribution under a Latent Factor Model*.

A spiked covariance model represents latent variability plus isotropic noise. Bias-corrected spike strengths, noise variance and estimated eigenvector alignments determine a precision weight, selected by minimizing estimated total asymptotic scaling-factor variance. The factor count uses KN by default (Kritchman and Nadler, 2008), or PLY (Passemier et al., 2017), unless `n.factors` is supplied. Variance inflation coefficients are estimated by default. Scaling factors use precision-weighted TLS, and marginal confidence intervals use the latent model's estimated asymptotic covariance.

`"EE"` Estimating-equation fingerprinting developed by Ma et al. (2023) in *Optimal Fingerprinting with Estimating Equations* and refined by Li et al. (2025) in *Improved Optimal Fingerprinting Based on Estimating Equations Reaffirms Anthropogenic Effect on Global Warming*.

The estimating equations correct for measurement error in forcing ensemble means. A pooled spatial covariance supplies the working weight, with a block-diagonal temporal-independence structure for multiple time steps. Marginal confidence intervals use an estimating-equation variance estimate and a normal approximation. Options include variance calibration through `cal.a`, linear contrasts through `Proj`, missing-response handling, and block-bootstrap variance estimation.

"PBC" Parametric-bootstrap-calibrated fingerprinting proposed by Li et al. (2021) in *Uncertainty in Optimal Fingerprinting Is Underestimated*.

A regularized covariance weight is used for TLS, with linear shrinkage for cov.method = "l2" or nonlinear minimum-variance shrinkage for cov.method = "mv". Parametric replicates regenerate the response, forcing ensemble means and controls from the fitted Gaussian model and re-estimate the weight. Bootstrap and normal-approximation TLS intervals are returned together with standard-error calibration ratios in pbc.ratio; the ratios are supplied separately rather than applied to the interval endpoints.

"TS" Two-sample regularized fingerprinting described by Ribes et al. (2013) in *Application of Regularised Optimal Fingerprinting to Attribution. Part I: Method, Properties and Idealised Analysis*, using the TLS framework of Allen and Stott (2003).

One control sample estimates the covariance weight used for TLS; a second sample, ctlruns.bhvar, supplies the variance estimate. Marginal confidence intervals use a normal approximation. Nonparametric bootstrap intervals are also returned, based on resampled coordinates of the transformed regression. Li et al. (2021) assess the uncertainty properties of the two-sample procedure.

"TS.TempSt" Two-sample fingerprinting with a temporally independent working covariance, combining the framework described by Ribes et al. (2013) in *Application of Regularised Optimal Fingerprinting to Attribution. Part I: Method, Properties and Idealised Analysis* with covariance pooling used by Ma et al. (2023) and Li et al. (2025).

Control data are pooled across time to estimate a common spatial covariance, which is repeated in diagonal temporal blocks and used as the TLS covariance weight. A second control sample supplies the variance estimate. Marginal normal-approximation intervals and nonparametric bootstrap intervals follow the same construction as "TS".

For "poly", "linear" and "latent", estimate.inflation = FALSE fixes both variance inflation coefficients at one; otherwise individual forcing members must be supplied in ensembles. The returned weight is a covariance weight for polynomial/linear fits and a precision weight for latent fits. The quadratic TLS criterion uses the inverse of the former and the latter directly. These fits support the common standardized-residual plot() method. See the implementation help pages for the residual diagnostics and their calibration limitations.

The methods "EE", "PBC", "TS" and "TS.TempSt" are implemented by fingerprintLegacy with the selected method passed unchanged. ctlruns supplies ctlruns.sigma; ctlruns.bhvar defaults to the same sample for compatibility. Supply a separate second control sample for a two-sample analysis. Arguments S and T describe the spatial and temporal dimensions used by "EE" and "TS.TempSt". Set B to a positive number of bootstrap replicates for "PBC", "TS" and "TS.TempSt". The modern options degree, estimate.inflation and latent-factor selection controls do not apply to these four methods.

Value

The result returned by the selected implementation. Polynomial and latent fits inherit from class fingerprint and support plot(). The EE and TLS procedures retain their implementation-specific list structures.

References

Li, H. and Li, Y. (2026a). Adaptable Fingerprinting with Nonlinear Shrinkage for Climate Change Detection and Attribution under Variance Heterogeneity. arXiv preprint. https://arxiv.org/

abs/2608.12496.

Li, H. and Li, Y. (2026b). Fingerprint Analysis for Climate Change Detection and Attribution under a Latent Factor Model. arXiv preprint. https://arxiv.org/abs/2609.13488.

Ma, S., Wang, T., Yan, J. and Zhang, X. (2023). Optimal Fingerprinting with Estimating Equations. *Journal of Climate*, **36**, 7109–7122. doi:10.1175/JCLID220681.1.

Li, Y., Wang, T., Yan, J. and Zhang, X. (2025). Improved Optimal Fingerprinting Based on Estimating Equations Reaffirms Anthropogenic Effect on Global Warming. *Journal of Climate*, **38**, 1779–1790. doi:10.1175/JCLID240193.1.

Li, Y., Chen, K., Yan, J. and Zhang, X. (2021). Uncertainty in Optimal Fingerprinting Is Underestimated. *Environmental Research Letters*, **16**, 084043.

Allen, M. R. and Stott, P. A. (2003). Estimating Signal Amplitudes in Optimal Fingerprinting, Part I: Theory. *Climate Dynamics*, **21**, 477–491. doi:10.1007/s0038200303139.

Ribes, A., Planton, S. and Terray, L. (2013). Application of Regularised Optimal Fingerprinting to Attribution. Part I: Method, Properties and Idealised Analysis. *Climate Dynamics*, **41**, 2817–2836. doi:10.1007/s0038201317357.

Ledoit, O. and Wolf, M. (2004). A Well-Conditioned Estimator for Large-Dimensional Covariance Matrices. *Journal of Multivariate Analysis*, **88**, 365–411. doi:10.1016/S0047259X(03)000964.

Kritchman, S. and Nadler, B. (2008). Determining the Number of Components in a Factor Model from Limited Noisy Data. *Chemometrics and Intelligent Laboratory Systems*, **94**, 19–32.

Passemier, D., Li, Z. and Yao, J. (2017). On Estimation of the Noise Variance in High Dimensional Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society: Series B*, **79**, 51–67.

See Also

fingerprintPoly, fingerprintLatent, fingerprintLegacy, plot.fingerprint

Examples

```
set.seed(12)
N <- 60
X <- matrix(3 * sin(seq_len(N) / 6), ncol = 1)
Z <- matrix(rnorm(120 * N), nrow = 120)
Y <- as.vector(X + rnorm(N))
fit <- fingerprint(X, Y, mruns = 30, ctlruns = Z,
                   method = "linear", estimate.inflation = FALSE)
fit$ci
plot(fit)
```

`fingerprintLatent` *Detection and attribution under a latent factor model*

Description

Implements the latent-factor optimal fingerprinting framework proposed by Li and Li (2026). The joint errors-in-variables model links the observed climate response, simulated forcing responses, and preindustrial control runs through a common spiked covariance structure. Variance inflation coefficients account for differences in the magnitude of internal variability between observations and simulations. The procedure corrects empirical spike strengths and noise variance, estimates the inflation coefficients, and selects precision weights by minimizing the estimated total asymptotic variance of the forcing scaling-factor estimator.

Usage

```
fingerprintLatent(
    Xtilde,
    Y,
    mruns,
    ctlruns,
    n.factors = NULL,
    ensembles = NULL,
    estimate.inflation = TRUE,
    conf.level = 0.9,
    theta = NULL,
    inflation.grid = NULL,
    optim.control = list(maxit = 200),
    factor.method = c("KN", "PLY"),
    factor.level = 0.95,
    factor.delta = 0.05,
    max.factors = min(nrow(ctlruns) - 1L, floor(ncol(ctlruns) * 0.1))
)
```

Arguments

`Xtilde`	Numeric $N \times p$ matrix of forcing ensemble means.
`Y`	Numeric observation vector of length N.
`mruns`	Positive integer ensemble sizes n_i, one per forcing.
`ctlruns`	Numeric $m \times N$ matrix of known zero-mean controls. Singular sample covariance is allowed, including $m < N$.
`n.factors`	Number K of latent factors. With `NULL` (default), estimate it using `factor.method`. An explicit nonnegative integer smaller than $\min(N, m)$ overrides selection; zero gives an isotropic model.
`ensembles`	List of p individual forced ensembles, each `mruns[i]` by N. Required to estimate inflation coefficients; their means must agree with `Xtilde`.
`estimate.inflation`	Estimate both variance inflation coefficients (default `TRUE`), or fix both at one (`FALSE`).
`conf.level`	Marginal confidence level, default 0.9.
`theta`	Optional fixed vector of K spike weights, each at least -1. With `NULL`, select weights by constrained numerical optimization.

`inflation.grid`	Optional increasing, nonnegative candidate grid for the variance factor s_x^2, not the standard deviation multiplier s_x. With `NULL`, solve residual matching numerically after adaptive bracketing.
`optim.control`	List of control settings passed to `stats::optim` for weight selection.
`factor.method`	Factor-count selector: `"KN"` (default) or `"PLY"`. Used when `n.factors = NULL`.
`factor.level`	Tracy–Widom probability for KN selection, default 0.95. Separate from the scaling-factor confidence level `conf.level`.
`factor.delta`	Positive penalty exponent δ for PLY selection, default 0.05.
`max.factors`	Largest candidate count. Defaults to $\min(m-1, \lfloor 0.1N \rfloor)$ and must be a nonnegative integer smaller than $\min(m, N)$. Used only for automatic selection.

Details

Joint model for observations and simulations: Let $Y \in \mathbb{R}^N$ denote the observed climate response and let $X = (X_1, \ldots, X_p) \in \mathbb{R}^{N \times p}$ contain the unknown, deterministic mean responses to p external forcings. For forcing i, $\widetilde{X}_{ij}$ is the response in ensemble member j, and n_i is the ensemble size. The vectors $Z_1, \ldots, Z_m$ are preindustrial control runs. The joint model is

$$Y = X\beta + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \Sigma),$$

$$\widetilde{X}_{ij} = X_i + \eta_{ij}, \qquad \eta_{ij} \sim \mathcal{N}(0, s_x^2\Sigma), \quad j = 1, \ldots, n_i, \quad i = 1, \ldots, p,$$

$$Z_j \sim \mathcal{N}(0, s_z^2\Sigma), \qquad j = 1, \ldots, m.$$

The observation error ϵ, all forcing-simulation errors η_{ij}, and all control runs Z_j are mutually independent. The regression coefficients β are the forcing scaling factors used for detection and attribution. The positive variance inflation coefficients s_x^2 and s_z^2 are variance ratios for forced simulations and controls relative to observations; they need not exceed one. Setting both to one imposes equal internal variability across the three data sources.

The observation error has the latent-factor representation

$$\epsilon = \sum_{k=1}^{K} \sqrt{\ell_k} U_k f_k + \sigma e, \qquad f_k \overset{\text{iid}}{\sim} \mathcal{N}(0,1), \quad e \sim \mathcal{N}(0, I_N).$$

Here $f_1, \ldots, f_K$ are mutually independent latent factors, independent of the idiosyncratic noise vector e. The deterministic vectors $U_1, \ldots, U_K$ are orthonormal loading directions; $\sqrt{\ell_k}U_k$ is the loading vector for factor k, $\ell_k > 0$ is its strength, and $\sigma^2 > 0$ is the isotropic idiosyncratic noise variance. Consequently,

$$\Sigma = U \operatorname{diag}(\ell_1, \ldots, \ell_K) U^\top + \sigma^2 I_N, \qquad U = (U_1, \ldots, U_K).$$

The covariance eigenvalue in direction U_k is $\ell_k + \sigma^2$. Forcing-simulation errors and controls share these loading directions and relative factor strengths, with independent realizations of the latent factors and idiosyncratic noise, scaled by s_x and s_z, respectively.

Each unknown forcing response is estimated by its ensemble mean,

$$\overline{X}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \widetilde{X}_{ij}, \qquad \mathrm{E}(\overline{X}_i) = X_i, \qquad \mathrm{Cov}(\overline{X}_i) = \frac{s_x^2}{n_i}\Sigma.$$

The columns of `Xtilde` contain these means, `mruns` contains the ensemble sizes, and the rows of `ctlruns` contain the controls. Individual forcing ensembles are supplied in `ensembles` when estimating the variance inflation coefficients. Uncertainty in the ensemble means gives rise to the errors-in-variables formulation.

Scaling-factor estimation: Write $\overline{X} = (\overline{X}_1, \ldots, \overline{X}_p)$ and $D = \mathrm{diag}(1/n_1, \ldots, 1/n_p)$. For a positive semidefinite precision weight W and a given standard deviation multiplier $s = s_x$, the scaling factors minimize

$$\widehat{\beta}(s, W) = \arg\min_{\beta} \frac{(Y - \overline{X}\beta)^\top W (Y - \overline{X}\beta)}{1 + s^2 \beta^\top D \beta}.$$

The denominator accounts for uncertainty in the simulated ensemble means. The precision weight W enters the numerator directly, unlike the covariance weight returned by `fingerprintPoly`. The estimated variance inflation coefficients s_x^2 and s_z^2 are returned as `sigma.x2` and `sigma.z2`.

Spike correction and weight selection: Control covariance is $S = Z^\top Z/m$, without centering. Algorithm 1 corrects the control-scale spike strengths $\pi_k = s_z^2 \ell_k$ and noise variance $\sigma_z^2 = s_z^2 \sigma^2$. The second correction uses the updated noise variance in both the empirical excess eigenvalues and the spike equation. Empirical eigenvectors $\widetilde{U}$ are retained and their squared alignments with population eigenvectors are estimated from the corrected spikes. The precision weights take the form

$$W_\Theta = I_N + \widetilde{U}\,\mathrm{diag}(\theta_k)\widetilde{U}^\top, \qquad \theta_k \geq -1.$$

A weight of -1 removes that spike direction and is allowed. Equations (10)–(15), including $-\alpha\Psi_\Theta D$ in equation (14), provide the trace and design corrections used in Theorem 4.

When `n.factors = NULL`, the factor count is estimated from all N ordered eigenvalues of the control covariance, including zero eigenvalues when $m < N$. For these selectors, the sample size is $n = m$ control runs and the data dimension is $p = N$; it is not the number of forcing predictors. An explicit nonnegative `n.factors` bypasses automatic selection.

`factor.method = "PLY"` applies the Passemier et al. (2017) criterion. Let v_k be the Algorithm 1 bias-adjusted noise variance when assuming k spikes, and let M be `max.factors`. It selects the smallest minimizer of

$$v_k + k v_M \frac{p/n + 2\sqrt{p/n}}{p}\left(1 + \frac{n}{p^{1+\delta}}\right), \quad k = 0, \ldots, M,$$

where δ is `factor.delta`. Noise estimation for trial counts is separated from the eigenvector-alignment checks needed for inference.

`factor.method = "KN"` (default) applies the sequential Kritchman–Nadler (2008) Tracy–Widom threshold, using `RMTstat::qtw(factor.level, beta = 1)`. It tests successive eigenvalues with $d = p - k$, the finite-sample centering and scale in the supplied selector, and the adjusted noise estimate v_k. It returns the count before the first nonsignificant eigenvalue, or M if all candidates pass. Both methods allow zero factors. The default cap is $M = \min(m - 1, \lfloor 0.1N \rfloor)$; if $M = 0$, selection returns zero without requiring `RMTstat`. The KN option otherwise requires that optional package; explicit counts and PLY do not.

Selected factors at the detectability boundary have zero alignment, so the inferential variance formula is undefined there. An explicitly smaller `n.factors` is needed in that case; the selected count is not silently reduced. Zero factors gives an isotropic model.

When `theta = NULL`, local L-BFGS-B optimization minimizes $\operatorname{tr} \widehat{\Xi}_\Theta$, starting from the naive precision weights and identity weights. This finite numerical procedure does not guarantee a global optimum. Supply `theta` for a fixed-weight analysis.

Inflation coefficients and inference: With `estimate.inflation = TRUE`, individual forced runs estimate $\alpha = s_x^2/s_z^2$ from the ratio of within-ensemble variation to mean control variation. For each candidate weight, equations (16)–(17) estimate s_x by residual matching and then set $\widehat{s}_z = \widehat{s}_x/\sqrt{\alpha}$. Both coefficients are re-estimated as the weights change. The API consistently accepts and returns variance multipliers s_x^2 and s_z^2. With `estimate.inflation = FALSE`, both equal one and individual forced runs are unnecessary. Marginal confidence intervals condition on the selected factor count and use Theorem 4 with covariance $\widehat{\Xi}_\Theta/N$.

Residual standardization follows equation (21): $\widehat{s}_z\widehat{\epsilon}_i/\sqrt{S_{ii}}$. It omits the extra ensemble-mean variance multiplier used by `fingerprintPoly`; the resulting residuals may therefore have extra variability. Plot reference bands are pointwise.

With estimated inflation coefficients and exact residual matching, equations (16)–(17) force the residual-test numerator in Section 3.5 to zero. Consequently, `diagnostic.stat` is near zero and `p.value` near 0.5; they do not provide an independent goodness-of-fit check in this setting. An explicit finite `inflation.grid` only approximates this equality. With inflation coefficients fixed at one, the algebraic degeneracy does not apply. Missing values are unsupported.

Value

An object of classes `fingerprintLatent` and `fingerprint`, with a `plot()` method for standardized residuals. Components are:

`beta`	Named vector of p forcing scaling-inflation coefficient estimates.
`var`	Estimated $p \times p$ coefficient covariance $\widehat{\Xi}_\Theta/N$.
`sd`	Named vector of standard errors, `sqrt(diag(var))`.
`ci`	Marginal confidence intervals with columns `lower` and `upper`.
`sigma.x2`, `sigma.z2`	Estimated variance inflation coefficients s_x^2 and s_z^2, or one when `estimate.inflation = FALSE`.
`n.factors`	The selected or supplied integer factor count.
`theta`	Selected or supplied spike weights θ_k.
`spikes`	Bias-corrected control-scale strengths $\pi_k = s_z^2\ell_k$.
`noise.var`	Bias-corrected control noise variance $s_z^2\sigma^2$.
`weight`	$N \times N$ precision weight W_Θ, used directly in the quadratic loss.
`residuals`	Observed minus fitted response, $Y - \overline{X}\widehat{\beta}$.
`standardized.residuals`	Residuals standardized according to equation (21).
`diagnostic.stat`	Standardized residual consistency statistic from Section 3.5.
`p.value`	Upper-tail normal reference p-value; see the estimation limitation in Details.
`call`	The matched function call.

References

Li, H. and Li, Y. (2026). Fingerprint Analysis for Climate Change Detection and Attribution under a Latent Factor Model. arXiv preprint. https://arxiv.org/abs/2609.13488.

See Also

fingerprint, fingerprintPoly, plot.fingerprint

Examples

```
set.seed(2026)
N <- 80
n <- 30
m <- 160
u <- rep(1 / sqrt(N), N)
signal <- 3 * sin(seq_len(N) / 6)
Z <- matrix(rnorm(m * N), m) + outer(rnorm(m), sqrt(20) * u)
E <- matrix(rnorm(n * N), n) + outer(rnorm(n), sqrt(20) * u)
E <- sweep(E, 2, signal, "+")
Xtilde <- matrix(colMeans(E), ncol = 1)
Y <- signal + rnorm(N) + sqrt(20) * rnorm(1) * u
fit <- fingerprintLatent(Xtilde, Y, n, Z,
                         ensembles = list(E), factor.method = "PLY")
fit$beta
fit$ci
plot(fit)
```

---

fingerprintLegacy *Legacy fingerprinting methods for comparison and reproducibility*

Description

Provides the earlier detection and attribution procedures for reproducing previous analyses and comparing methods: EE, PBC, TS and TS.TempSt, with their original estimation and confidence interval procedures. This is the original fingerprint function under a new name; its arguments and defaults are retained. For new analyses, use fingerprint, which implements the manuscript method with polynomial shrinkage and optional estimation of variance inflation coefficients.

Usage

```
fingerprintLegacy(
  Xtilde,
  Y,
  mruns,
  ctlruns.sigma,
  ctlruns.bhvar,
  S,
  T,
```

```
  B = 0,
  Proj = diag(ncol(Xtilde)),
  method = c("EE", "PBC", "TS", "TS.TempSt"),
  cov.method = c("l2", "mv"),
  conf.level = 0.9,
  missing = FALSE,
  cal.a = TRUE,
  ridge = 0
)
```

Arguments

Xtilde	$n \times p$ matrix, signal pattern to be detected.
Y	$n \times 1$ matrix, length $S \times T$, observed climate variable.
mruns	number of ensembles to estimate the corresponding pattern. It is used as the scale of the covariance matrix for X_i.
ctlruns.sigma	$m \times n$ matrix, a group of m independent control runs for estimating covariance matrix, which is used in point estimation of the signal factors.
ctlruns.bhvar	$m \times n$ matrix, another group of m independent control runs for estimating the corresponding confidence interval of the signal factors, in EE or PBC approach should be same as ctlruns.sigma.
S	number of locations for the observed responses.
T	number of time steps for the observed responses.
B	number of replicates in bootstrap procedure, mainly for the PBC and TS methods, can be specified in "EE" method but not necessary. By default B = 0 as the default method is "EE".
Proj	The projection matrix for computing for scaling factors of other external forcings with the current input when using EE. For example, when ALL and NAT are used for modeling, specifying the Proj matrix to return the results for ANT and NAT.
method	for estimating the scaling factors and corresponding confidence interval
cov.method	method for estimation of covariance matrix in confidence interval estimation of PBC method. (only for PBC method).
conf.level	confidence level for confidence interval estimation.
missing	indicator for whether missing values present in Y.
cal.a	indicator for calculating the a value, otherwise use default value a = 1. (only for EE method)
ridge	shrinkage value for adjusting the method for missing observations if missing = TRUE. (only for EE method)

Value

a list of the fitted model including point estimate and interval estimate of coefficients and corresponding estimate of standard error.

Author(s)

Yan Li

References

- Gleser (1981), Estimation in a Multivariate "Errors in Variables" Regression Model: Large Sample Results, *Ann. Stat.* 9(1) 24–44.
- Golub and Laon (1980), An Analysis of the Total Least Squares Problem, *SIAM J. Numer. Anal*. 17(6) 883–893.
- Pesta (2012), Total least squares and bootstrapping with applications in calibration, *Statistics* 47(5), 966–991.
- Li et al (2021), Uncertainty in Optimal Fingerprinting is Underestimated, *Environ. Res. Lett.* 16(8) 084043.
- Sai et al (2023), Optimal Fingerprinting with Estimating Equations, *Journal of Climate* 36(20), 7109—7122.
- Li et al (2025), Improved Optimal Fingerprinting Based on Estimating Equations Reaffirms Anthropogenic Effect on Global Warming, *Journal of Climate*. 38(8), 1779—1790.

See Also

`fingerprint` for the recommended manuscript method.

Examples

```
## load the example dataset
data(simDat)
Cov <- simDat$Cov[[1]]
ANT <- simDat$X[, 1]
NAT <- simDat$X[, 2]

## generate the simulated data set
## generate regression observation
Y <- MASS::mvrnorm(n = 1, mu = ANT + NAT, Sigma = Cov)
## generate the forcing responses
mruns <- c(1, 1)
Xtilde <- cbind(MASS::mvrnorm(n = 1, mu = ANT, Sigma = Cov / mruns[1]),
              MASS::mvrnorm(n = 1, mu = NAT, Sigma = Cov / mruns[2]))
## control runs
ctlruns <- MASS::mvrnorm(100, mu = rep(0, nrow(Cov)), Sigma = Cov)
## ctlruns.sigma for the point estimation and ctlruns.bhvar for the interval estimation
ctlruns.sigma <- ctlruns.bhvar <- ctlruns
## number of locations
S <- 25
## number of year steps
T <- 10

## call the function to estimate the signal factors via EE
fingerprintLegacy(Xtilde, Y, mruns,
          ctlruns.sigma, ctlruns.bhvar,
```

```
S, T,
## B = 0, by default
method = "EE",
conf.level = 0.9,
cal.a = TRUE,
missing = FALSE, ridge = 0)
```

`fingerprintPoly` *Detection and attribution with polynomial shrinkage under variance heterogeneity*

Description

Implements the method proposed in work by Li and Li (2026) for detection and attribution (D&A) analysis of climate change using polynomial shrinkage under variance heterogeneity. The framework allows forced simulations and control runs to have different internal variability magnitudes from observations, while sharing a common covariance structure. It estimates forcing scaling factors, optionally estimates variance inflation coefficients, and provides asymptotic confidence intervals and approximate residual diagnostics.

Usage

```
fingerprintPoly(
    Xtilde,
    Y,
    mruns,
    ctlruns,
    ensembles = NULL,
    degree = 3L,
    estimate.inflation = TRUE,
    conf.level = 0.9,
    coefficient.grid = NULL,
    grid.size = 21L,
    psi1 = NULL,
    psi2 = NULL,
    psi3 = 1e-3,
    inflation.grid = NULL,
    refinements = 1L
)
```

Arguments

`Xtilde`	Numeric N by p matrix of ensemble means.
`Y`	Numeric observation vector of length N.
`mruns`	Positive integer ensemble sizes, one per forcing.
`ctlruns`	Numeric m by N matrix of zero-mean control runs. The number of runs m may be smaller than N; the sample covariance may be singular.

`ensembles`	List of p matrices, each n_i by N, containing individual forced runs. Required only when estimate.inflation is TRUE. Their means must agree with Xtilde and their row counts with mruns.
`degree`	Positive integer polynomial degree; default 3, or 1 for linear.
`estimate.inflation`	Estimate both variance inflation coefficients from data (default TRUE). FALSE fixes both at one and skips initialization, the inflation-coefficient search and refinement.
`conf.level`	Confidence level for marginal normal intervals.
`coefficient.grid`	Optional matrix with degree columns, each row containing ell_1,...,ell_k for f(x)=1+sum(ell_j*x^j).
`grid.size`	Odd integer number of points per coefficient (at least 5), default 21. The default grid spans the manuscript bounds with additional points near zero to resolve the bounded admissible region. At most 100000 combinations.
`psi1`	Upper spectral domain bound; defaults to 1.05*max(eigen(S)).
`psi2`	Upper bound on f(psi1); defaults to max(1.1,0.7*psi1).
`psi3`	Minimum polynomial root separation, default 0.001.
`inflation.grid`	Optional strictly increasing nonnegative grid for sigma_x^2. Otherwise an adaptive upper bound and a mixed linear/logarithmic grid is used.
`refinements`	Number of additional inflation/shrinkage updates, default 1.

Details

Model: Li and Li (2026) extend optimal fingerprinting to allow different magnitudes of internal variability in observations, forced simulations and control runs:

$$Y = X\beta + \epsilon, \qquad \epsilon \sim N(0, \Sigma),$$

$$\widetilde{X}_{ik} = X_i + \eta_{ik}, \qquad \eta_{ik} \sim N(0, \sigma_x^2 \Sigma), \quad k = 1, \ldots, n_i,$$

$$Z_j \sim N(0, \sigma_z^2 \Sigma), \quad j = 1, \ldots, m.$$

Here $Y \in \mathbb{R}^N$ is the observed climate response, $X = (X_1, \ldots, X_p)$ contains the true responses to p forcings, and β contains their scaling factors. The errors are mutually independent and share the covariance structure Σ. The coefficients σ_x^2 and σ_z^2 measure simulation variability relative to observations; setting both to one gives the conventional equal-variability model.

The inputs `Xtilde` and `mruns` specify the ensemble means $\widetilde{X}_i = n_i^{-1} \sum_k \widetilde{X}_{ik}$ and ensemble sizes n_i. Write $D = \mathrm{diag}(1/n_1, \ldots, 1/n_p)$.

Polynomial shrinkage and scaling factors: The control covariance $S = m^{-1} \sum_j Z_j Z_j^\top$. It is regularized using

$$W = f(S), \qquad f(t) = 1 + \sum_{j=1}^{k} \ell_j t^j.$$

Since $f(0) = 1$ and f is nondecreasing on the sample spectrum, W is positive definite even when S is singular. The degree k is controlled by `degree`: one gives linear shrinkage, while three is

recommended in Li and Li (2026). Candidate polynomials must be nondecreasing on $[0, \psi_1]$, satisfy $f(\psi_1) \leq \psi_2$, and have roots separated by at least ψ_3.

For a given W and $s = \sigma_x^2$, the scaling factors minimize the weighted total least squares criterion

$$\widehat{\beta}(s, W) = \arg\min_{\beta} \frac{(Y - \widetilde{X}\beta)^\top W^{-1}(Y - \widetilde{X}\beta)}{1 + s\beta^\top D\beta}.$$

The polynomial coefficients are selected by minimizing the estimated total asymptotic variance $\operatorname{tr}\widehat{\Xi}(f)$ over an admissible grid. The default uses 21 points per coefficient; `grid.size` and `coefficient.grid` control the search resolution. This is a finite-grid approximation to the optimization in Li and Li (2026).

Estimation of variance inflation coefficients and inference: The method estimates both variance inflation coefficients, σ_x^2 and σ_z^2, to account for differences in internal variability between observations, forced simulations and control runs. This is the default behavior (`estimate.inflation = TRUE`). Estimation uses the individual forced runs in `ensembles`, in addition to the ensemble means and control runs.

The within-ensemble variation estimates the marginal variability of the forced simulations:

$$V_X = \frac{\sum_{i=1}^{p}\sum_{k=1}^{n_i} \|\widetilde{X}_{ik} - \widetilde{X}_i\|^2}{N(\sum_i n_i - p)}.$$

For a given weight matrix W, the forced-simulation inflation coefficient is estimated by matching this variation to the residual variation implied by candidate values of σ_x^2:

$$\widehat{\sigma}_x^2 = \arg\min_{s \in \mathcal{G}} \left| \frac{s\|Y - \widetilde{X}\widehat{\beta}(s, W)\|^2}{N\{1 + s\widehat{\beta}(s, W)^\top D\widehat{\beta}(s, W)\}} - V_X \right|,$$

where $\mathcal{G}$ is the grid supplied through `inflation.grid` or constructed automatically. The control-run inflation coefficient is then estimated from the relative variation in controls and forced simulations:

$$\widehat{\sigma}_z^2 = \widehat{\sigma}_x^2 \frac{V_Z}{V_X}, \qquad V_Z = \operatorname{tr}(S)/N.$$

Initial estimates of both coefficients are obtained using normalized Ledoit–Wolf weights. These estimates are used to select the polynomial shrinkage rule. The coefficients are then re-estimated using the selected weight matrix, and the polynomial is reselected using the updated coefficients. The number of these additional updates is controlled by `refinements` (default one). The final estimates are returned as `sigma.x2` and `sigma.z2`.

As an optional restriction, `estimate.inflation = FALSE` fixes both coefficients at one, imposing equal variability across observations and simulations. This bypasses inflation estimation and refinement, so individual forced runs are unnecessary. Polynomial selection and scaling-factor inference remain part of the procedure.

The estimated inflation coefficients enter both the final scaling-factor estimate and its estimated covariance $\widehat{\Xi}(\widehat{f})/N$. Confidence intervals use the resulting asymptotic normal approximation. The residual consistency statistic and its approximate upper-tail normal p-value are returned as `diagnostic.stat` and `p.value`. Their reference calibration can be affected by estimation of the inflation coefficients and selection of the shrinkage rule. Missing observations are not supported.

Value

An object of classes `fingerprintPoly` and `fingerprint`, with a `plot()` method for standardized residuals. A list containing the following components, where N is the observation dimension and p is the number of forcings:

`beta` A named numeric vector of length p of estimated forcing scaling factors $\widehat{\beta}$.

`var` The $p \times p$ estimated covariance matrix of $\widehat{\beta}$, equal to $\widehat{\Xi}(\widehat{f})/N$.

`sd` A named numeric vector of standard errors, equal to `sqrt(diag(var))`.

`ci` A $p \times 2$ matrix of marginal confidence intervals at `conf.level`, with columns `lower` and `upper`.

`sigma.x2` The variance inflation coefficient for individual forced simulations relative to observations, estimated from data or fixed at one.

`sigma.z2` The variance inflation coefficient for control runs relative to observations, estimated from data or fixed at one.

`coefficients` A numeric vector $(1, \ell_1, \ldots, \ell_k)$ defining the selected polynomial $\widehat{f}(t) = 1 + \sum_{j=1}^{k} \ell_j t^j$. Coefficients are ordered by ascending power; the intercept is fixed at one.

`weight` The $N \times N$ regularized covariance weight matrix $W = \widehat{f}(S)$. Its inverse is used in the weighted TLS loss.

`residuals` A numeric vector of length N of fitted residuals $Y - \widetilde{X}\widehat{\beta}$.

`standardized.residuals` Residuals divided by their estimated marginal standard deviations, computed from $(1 + \widehat{\sigma}_x^2 \widehat{\beta}^\top D\widehat{\beta}) S_{ii}/\widehat{\sigma}_z^2$.

`diagnostic.stat` The standardized residual consistency statistic $\widehat{Q}/\sqrt{\widehat{V}}$, as defined in Li and Li (2026).

`p.value` The approximate upper-tail standard normal p-value for residual consistency. Factor estimation and shrinkage selection can affect its calibration.

`call` The matched function call.

References

Li, H. and Li, Y. (2026). Adaptable Fingerprinting with Nonlinear Shrinkage for Climate Change Detection and Attribution under Variance Heterogeneity. arXiv preprint. https://arxiv.org/abs/2608.12496.

See Also

`fingerprint`, `fingerprintLatent`; `plot.fingerprint` for residual plots; `fingerprintLegacy` for earlier methods and their confidence intervals.

Examples

```
set.seed(12)
N <- 60
signal <- 3 * sin(seq_len(N) / 6)
ensembles <- list(sweep(matrix(rnorm(30 * N), 30), 2, signal, "+"))
Xtilde <- matrix(colMeans(ensembles[[1]]), ncol = 1)
Z <- matrix(rnorm(120 * N), nrow = 120)
Y <- signal + rnorm(N)

fit <- fingerprintPoly(Xtilde, Y, mruns = 30, ctlruns = Z,
                   ensembles = ensembles)

# Observation index, with the default 95% normal reference band.
plot(fit)

# Use time coordinates and customize the appearance.
years <- 1961 + seq_len(N) - 1
plot(fit, index = years, xlab = "Year", col = "steelblue",
     band.col = "grey90")

# Change the pointwise reference level to 90%.
plot(fit, reference.level = 0.90)
```

`fpPrep` *Process netCDF4 Gridded Data into Format of the fingerprint() Function*

Description

This function detects the signal factors on the observed data via total least square linear regression model.

Usage

```
fpPrep(
  datafile,
  variable,
  region = "GL",
  target.year,
  average = 5,
  reference = c(1961, 1990),
  regridding = NULL
)
```

Arguments

`datafile`	path to the netCDF4 gridded datafile to be processed
`variable`	the climate variable to be extracted

region	the longitude and latitude boundary for selected region, should match the format of IPCC AR6 regions, the lon and lat of the vertices
target.year	vector of length 2, the starting and ending year of the selected time period for D&A analysis
average	number of years for average on each gridbox, default is 5-year average
reference	vector of length 2, the starting and ending year of reference time period for computing anomalies
regridding	whether the grid box should be regridded. Specify the size of the grid box, e.g., c(40, 30) for $40^\circ \times 30^\circ$ grid box. If no regridding, leave empty

Value

a dataset of the processed gridded climate variables for Y, Xtilde or control runs

Author(s)

Yan Li

`globalDat` *Sample Dataset Used in Numerical Studies of "Detection and Attribution Analysis of Temperature Changes with Estimating Equations".*

Description

A list of the observations and expected responses to different external forcings with name `Y`, `X`, `ctlruns`, `nruns.X` and `Xtilde` where

- `Y`: the $5^\circ \times 5^\circ$ gridded observations on global scale
- `X`: a data matrix of the expected responses to external forcing;
- `ctlruns`: replicates of control runs from pre-industrial simulations
- `nruns.X`: number of runs for the estimated responses to external forcings
- `Xtilde`: the selected estimated responses to external forcing ANT and NAT

Usage

```
data(globalDat)
```

Format

A data list with the observed and simulated data on global scale.

Examples

```
data(globalDat)
```

`plot.fingerprint` *Plot standardized fingerprinting residuals*

Description

Displays standardized residuals from polynomial or latent-factor fingerprinting against observation index or user-supplied coordinates, with a reference line at zero. Patterns can indicate departures from the model.

Usage

```
## S3 method for class 'fingerprint'
plot(
  x,
  index = NULL,
  xlab = if (is.null(index)) "Observation index" else "Coordinate",
  ylab = "Standardized residual",
  main = "Standardized residuals",
  pch = 19,
  reference.level = 0.95,
  band.col = "grey92",
  ylim = NULL,
  ...
)
```

Arguments

`x`	A fitted object returned by `fingerprint()`, `fingerprintPoly()`, or `fingerprintLatent()`.
`index`	Optional numeric vector of observation coordinates, such as time, of the same length as the residuals. Defaults to observation index.
`xlab, ylab`	Axis labels.
`main`	Plot title.
`pch`	Plotting symbol, default 19 (filled circles).
`reference.level`	Central standard normal reference probability, default 0.95. The band is pointwise, not simultaneous.
`band.col`	Fill color for the normal reference band.
`ylim`	Optional vertical axis limits; by default include the residuals and reference band.
`...`	Additional graphical arguments passed to `graphics::plot()`, such as `col`, `cex`, or `type`.

Details

Standardization uses estimated marginal variances; it does not remove dependence between observation coordinates. Residuals need not be independent even when the model is appropriate. The shaded band spans the standard normal quantiles at (1 - reference.level)/2 and (1 + reference.level)/2, approximately -1.96 and 1.96 by default. It is a pointwise reference interval for residuals, not a confidence interval for a fitted mean or a simultaneous envelope.

Value

The fitted object `x`, invisibly.

See Also

`fingerprint`

Examples

```
set.seed(12)
N <- 60
signal <- 3 * sin(seq_len(N) / 6)
ensembles <- list(sweep(matrix(rnorm(30 * N), 30), 2, signal, "+"))
Xtilde <- matrix(colMeans(ensembles[[1]]), ncol = 1)
Z <- matrix(rnorm(120 * N), nrow = 120)
Y <- signal + rnorm(N)

fit <- fingerprint(Xtilde, Y, mruns = 30, ctlruns = Z,
                   ensembles = ensembles)

# Observation index, with the default 95% normal reference band.
plot(fit)

# Use time coordinates and customize the appearance.
years <- 1961 + seq_len(N) - 1
plot(fit, index = years, xlab = "Year", col = "steelblue",
     band.col = "grey90")

# Change the pointwise reference level to 90%.
plot(fit, reference.level = 0.90)
```

---

simDat *Sample Dataset in Simulation Studies of "Regularized Fingerprinting in Detection and Attribution of Climate Change with Weight Matrix Optimizing the Efficiency in Scaling Factor Estimation".*

Description

A data list of designed covariance matrix and the expected responses to the two forcings ANT and NAT with name `Cov` and `X` where

- `Cov`: a list of the true covariance matrices;
- `X`: a data matrix of the expected responses to external forcing;

Usage

```
data(simDat)
```

Format

A data list with two separate data sets.

Examples

```
data(simDat)
```

Index

∗ **NetCDF**
    fpPrep, 18
∗ **datasets**
    globalDat, 19
    simDat, 21
∗ **equation**
    fingerprintLegacy, 11
∗ **estimating**
    fingerprintLegacy, 11
∗ **estimation**
    Covest, 2
∗ **fingerprinting**
    fingerprintLegacy, 11
    fpPrep, 18
∗ **regression**
    fingerprintLegacy, 11
∗ **tls**
    fingerprintLegacy, 11
∗ **variance**
    Covest, 2

Covest, 2

fingerprint, 3, *11*, *13*, *17*, *21*
fingerprintLatent, *6*, 6, *17*
fingerprintLegacy, *6*, 11, *17*
fingerprintPoly, *6*, *11*, 14
fpPrep, 18

globalDat, 19

plot.fingerprint, *6*, *11*, *17*, 20

simDat, 21