

Package ‘LMDT’

October 24, 2026

Type Package

Title Local Manifold Dispersion Test for Heteroskedasticity

Version 0.1.0

Description Implements the Local Manifold Dispersion Test (LMDT) for detecting complex, non-linear, and high-dimensional heteroskedasticity in linear regression models. LMDT measures the spectral smoothness (Dirichlet energy) of robust log-dispersion residuals over a self-tuning k-nearest neighbors graph Laplacian, as formulated in Jabbar (2026). The test statistic is asymptotically distribution-free with respect to error kurtosis and accommodates high-dimensional covariates ($p > 50$) via Gavish-Donoho optimal hard thresholding and supervised metric learning.

License GPL (≥ 3)

URL <https://amjed-droid.github.io/LMDT/>,
<https://github.com/amjed-droid/LMDT>

BugReports <https://github.com/amjed-droid/LMDT/issues>

Depends R ($\geq 4.0.0$)

Imports stats

Suggests testthat ($\geq 3.0.0$), knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Author Ahmed Sattar Jabbar [aut, cre]

Maintainer Ahmed Sattar Jabbar <ahmed.state.me@gmail.com>

Repository CRAN

Date/Publication 2026-10-24 09:30:02 UTC

Contents

LMDT-package	2
lmdt_graph	2
lmdt_metric	3
lmdt_pca	4
lmdt_test	4
plot.lmdt	6
print.lmdt	7
Index	8

LMDT-package	<i>Local Manifold Dispersion Test for Heteroskedasticity</i>
--------------	--

Description

Implements the Local Manifold Dispersion Test (LMDT) for detecting complex, non-linear, and high-dimensional heteroskedasticity in linear regression models. LMDT measures the graph-spectral Dirichlet energy of robust log-dispersion residuals over a self-tuning k-nearest neighbors normalized graph Laplacian.

Details

The primary user function is [lmdt_test](#), which supports both formula and matrix interfaces.

Author(s)

Ahmed Sattar Jabbar <ahmed.state.me@gmail.com>

References

Jabbar, A. S. (2026). The Local Manifold Dispersion Test: A Nonparametric Graph-Spectral Approach for Detecting Complex and High-Dimensional Heteroskedasticity. *Working Paper*.

lmdt_graph	<i>Self-Tuning Manifold Graph Construction</i>
------------	--

Description

Constructs an undirected, self-tuning k-nearest neighbors affinity graph and its normalized Laplacian matrix from a covariate matrix.

Usage

`lmdt_graph(X, k = NULL)`

Arguments

`X` A numeric matrix or data frame of covariates (n x p).
`k` Number of nearest neighbors. Default is adaptive rule of thumb.

Value

A list containing matrices W, S, L, local scales sigma, and vertex degrees.

Examples

```
set.seed(42)
X <- matrix(rnorm(50 * 2), ncol = 2)
g <- lmdt_graph(X, k = 5)
dim(g$S)
```

lmdt_metric	<i>Supervised Metric Learning for Sparse Noise Dilution</i>
-------------	---

Description

Computes correlation-based feature weights to soft-threshold uninformative ambient noise coordinates.

Usage

```
lmdt_metric(X, u, lambda = 0.15)
```

Arguments

`X` A numeric matrix of covariates (n x p).
`u` A numeric vector of log-dispersion scores.
`lambda` Shrinkage prior in (0, 1). Default is 0.15.

Value

A rescaled covariate matrix with filtered noise dimensions.

Examples

```
set.seed(42)
X <- matrix(rnorm(50 * 10), nrow = 50, ncol = 10)
u <- rnorm(50)
X_rescaled <- lmdt_metric(X, u, lambda = 0.15)
dim(X_rescaled)
```

lmdt_pca

Gavish–Donoho Optimal Adaptive PCA Denoising

Description

Filters high-dimensional correlated features by hard-thresholding singular values at the Gavish–Donoho (2014) optimal threshold.

Usage

```
lmdt_pca(X)
```

Arguments

X A numeric matrix of covariates (n x p).

Value

A matrix of retained principal components.

Examples

```
set.seed(42)
X <- matrix(rnorm(100 * 20), nrow = 100, ncol = 20)
X_clean <- lmdt_pca(X)
ncol(X_clean)
```

lmdt_test

Local Manifold Dispersion Test for Heteroskedasticity

Description

Performs the Local Manifold Dispersion Test (LMDT) for regression heteroskedasticity.

Usage

```
lmdt_test(x, ...)

## S3 method for class 'formula'
lmdt_test(formula, data = list(), ...)

## S3 method for class 'lm'
lmdt_test(x, ...)

## Default S3 method:
lmdt_test(
```

```

x,
y,
k = NULL,
method = c("permutation", "asymptotic", "wild_bootstrap"),
alternative = c("one.sided", "two.sided"),
denoise = c("none", "pca", "metric"),
B = 499L,
c = 1e-04,
lambda = 0.15,
...
)

```

Arguments

x	A numeric design matrix of regressors, or a formula.
...	Further arguments passed to methods.
formula	A symbolic regression formula (e.g. $y \sim x_1 + x_2$).
data	An optional data frame containing model variables.
y	A numeric vector of regression residuals or response variable.
k	An integer number of nearest neighbors. Default uses adaptive rule of thumb.
method	P-value calibration: "permutation" (default), "asymptotic", or "wild_bootstrap".
alternative	Alternative hypothesis: "one.sided" (smooth manifold variance, default) or "two.sided".
denoise	High-dimensional pre-filtering: "none" (default), "pca", or "metric".
B	Number of bootstrap or permutation replications. Default is 499.
c	Positive stabilizer for log-dispersion transform. Default is 1e-4.
lambda	Shrinkage prior for metric learning. Default is 0.15.

Value

An object of class "lmdt" and "htest".

References

Jabbar, A. S. (2026). The Local Manifold Dispersion Test: A Nonparametric Graph-Spectral Approach for Detecting Complex and High-Dimensional Heteroskedasticity. *Working Paper*.

Examples

```

set.seed(42)
n <- 100
x1 <- rnorm(n)
x2 <- rnorm(n)
e <- rnorm(n, sd = exp(0.5 * x1))
y <- 1 + 2 * x1 + 3 * x2 + e
fit <- lm(y ~ x1 + x2)

```

```
# Using formula interface with asymptotic calibration
test_asymp <- lmdt_test(y ~ x1 + x2, method = "asymptotic")
print(test_asymp)

# Using Wild Bootstrap calibration
test_boot <- lmdt_test(y ~ x1 + x2, method = "wild_bootstrap", B = 199)
print(test_boot)
```

plot.lmdt

Plot Diagnostic Visualizations for LMDT

Description

Produces diagnostic plots displaying the null distribution of the test statistic (under bootstrap/permutation or asymptotic Gaussian) and the ordered log-dispersion profile.

Usage

```
## S3 method for class 'lmdt'
plot(x, ...)
```

Arguments

x	An object of class "lmdt".
...	Additional graphical parameters passed to plotting functions.

Value

Invisibly returns x.

Examples

```
set.seed(42)
fit <- lm(rnorm(50) ~ matrix(rnorm(100), 50, 2))
res <- lmdt_test(fit, method = "asymptotic")
plot(res)
```

print.lmdt*Methods for LMDT Test Results*

Description

Print and summary methods for objects of class "lmdt".

Usage

```
## S3 method for class 'lmdt'
print(x, digits = 4, ...)

## S3 method for class 'lmdt'
summary(object, ...)
```

Arguments

x	An object of class "lmdt".
digits	Number of significant digits.
object	An object of class "lmdt".
...	Additional arguments.

Value

Invisibly returns the input object.

Index

* **package**

LMDT-package, [2](#)

LMDT (LMDT-package), [2](#)

LMDT-package, [2](#)

lmdt_graph, [2](#)

lmdt_metric, [3](#)

lmdt_pca, [4](#)

lmdt_test, [2, 4](#)

plot.lmdt, [6](#)

print.lmdt, [7](#)

summary.lmdt (print.lmdt), [7](#)