Package 'comets'

January 31, 2025

Type Package

Title Covariance Measure Tests for Conditional Independence

Version 0.1-1

Description Covariance measure tests for conditional independence testing against conditional covariance and nonlinear conditional mean alternatives. The package implements versions of the generalised covariance measure test (Shah and Peters, 2020, <doi:10.1214/19-aos1857>) and projected covariance measure test (Lundborg et al., 2023, <doi:10.1214/24-AOS2447>). The tram-GCM test, for censored responses, is implemented including the Cox model and survival forests (Kook et al., 2024, <doi:10.1080/01621459.2024.2395588>). Application examples to variable significance testing and modality selection can be found in Kook and Lundborg (2024, <doi:10.1093/bib/bbae475>).

Depends R (>= 4.2.0)

Imports ranger, glmnet, Formula, survival, coin

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

Suggests testthat (>= 3.0.0), ggplot2, tidyr, dplyr, xgboost

Config/testthat/edition 3

URL https://github.com/LucasKook/comets

BugReports https://github.com/LucasKook/comets/issues

NeedsCompilation no

Author Lucas Kook [aut, cre] (<https://orcid.org/0000-0002-7546-7356>), Anton Rask Lundborg [ctb]

Maintainer Lucas Kook <lucasheinrich.kook@gmail.com>

Repository CRAN

Date/Publication 2025-01-31 10:10:06 UTC

Contents

`comet` *Covariance measure tests with formula interface*

Description

Covariance measure tests with formula interface

Usage

```
comet(formula, data, test = c("gcm", "pcm", "wgcm"), ...)

comets(formula, data, test = c("gcm", "pcm", "wgcm"), ...)
```

Arguments

`formula`	Formula of the form `Y ~ X | Z` for testing Y independent of X given Z.
`data`	Data.frame containing the variables in `formula`.
`test`	Character string; `"gcm"`, `"pcm"`, or `"wgcm"`.
`...`	Additional arguments passed to `test`.

Details

Formula-based interface for the generalised and projected covariance measure tests.

Value

Object of class `"gcm"`, `"wgcm"` or `"pcm"` and `"htest"`. See `gcm` and `pcm` for details.

References

Kook, L. & Lundborg A. R. (2024). Algorithm-agnostic significance testing in supervised learning with multimodal data. Briefings in Bioinformatics, 25(6), 2024. doi:10.1093/bib/bbae475

Examples

```
tn <- 1e2
df <- data.frame(y = rnorm(tn), x1 = rnorm(tn), x2 = rnorm(tn), z = rnorm(tn))
comet(y ~ x1 + x2 | z, data = df, test = "gcm")
```

---

gcm *Generalised covariance measure test*

Description

Generalised covariance measure test

Usage

```
gcm(
  Y,
  X,
  Z,
  alternative = c("two.sided", "less", "greater"),
  reg_YonZ = "rf",
  reg_XonZ = "rf",
  args_YonZ = NULL,
  args_XonZ = NULL,
  type = c("quadratic", "max", "scalar"),
  B = 499L,
  coin = TRUE,
  cointrol = list(distribution = "asymptotic"),
  return_fitted_models = FALSE,
  multivariate = c("none", "YonZ", "XonZ", "both"),
  ...
)
```

Arguments

Y	Vector or matrix of response values.
X	Matrix or data.frame of covariates.
Z	Matrix or data.frame of covariates.
alternative	A character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". Only applies if type = "quadratic" and Y and X are one-dimensional.
reg_YonZ	Character string or function specifying the regression for Y on Z. See ?regressions for more detail.
reg_XonZ	Character string or function specifying the regression for X on Z. See ?regressions for more detail.

args_YonZ	A list of named arguments passed to reg_YonZ.
args_XonZ	A list of named arguments passed to reg_XonZ.
type	Type of test statistic, either "quadratic" (default) or "max". If "max" is specified, the p-value is computed based on a bootstrap approximation of the null distribution with B samples.
B	Number of bootstrap samples. Only applies if type = "max" is used.
coin	Logical; whether or not to use the coin package for computing the test statistic and p-value. The coin package computes variances with n - 1 degrees of freedom. The default is TRUE.
cointrol	List; further arguments passed to independence_test.
return_fitted_models	Logical; whether to return the fitted regressions (default is FALSE).
multivariate	Character; specifying which regression can handle multivariate outcomes ("none", "YonZ", "XonZ", or "both"). If "none", then the regression is run using each column in Y (or X) as the response.
...	Additional arguments passed to reg_YonZ.

Details

The generalised covariance measure test tests whether the conditional covariance of Y and X given Z is zero.

Value

Object of class 'gcm' and 'htest' with the following components:

statistic	The value of the test statistic.
p.value	The p-value for the hypothesis
parameter	In case X is multidimensional, this is the degrees of freedom used for the chi-squared test.
hypothesis	String specifying the null hypothesis.
null.value	String specifying the null hypothesis.
method	The string "Generalised covariance measure test".
data.name	A character string giving the name(s) of the data.
rY	Residuals for the Y on Z regression.
rX	Residuals for the X on Z regression.
models	List of fitted regressions if return_fitted_models is TRUE.

References

Rajen D. Shah, Jonas Peters "The hardness of conditional independence testing and the generalised covariance measure," The Annals of Statistics, 48(3), 1514-1538. doi:10.1214/19aos1857

Examples

```
n <- 1e2
X <- matrix(rnorm(2 * n), ncol = 2)
colnames(X) <- c("X1", "X2")
Z <- matrix(rnorm(2 * n), ncol = 2)
colnames(Z) <- c("Z1", "Z2")
Y <- X[, 2]^2 + Z[, 2] + rnorm(n)
(gcm1 <- gcm(Y, X, Z))
```

---

pcm *Projected covariance measure test for conditional mean independence*

Description

Projected covariance measure test for conditional mean independence

Usage

```
pcm(
  Y,
  X,
  Z,
  rep = 1,
  est_vhat = TRUE,
  reg_YonXZ = "rf",
  reg_YonZ = "rf",
  reg_YhatonZ = "rf",
  reg_VonXZ = "rf",
  reg_RonZ = "rf",
  args_YonXZ = NULL,
  args_YonZ = NULL,
  args_YhatonZ = NULL,
  args_VonXZ = NULL,
  args_RonZ = NULL,
  frac = 0.5,
  indices = NULL,
  coin = FALSE,
  cointrol = NULL,
  return_fitted_models = FALSE,
  ...
)
```

Arguments

Y	Vector of response values. Can be supplied as a numeric vector or a single column matrix.

X	Matrix or data.frame of covariates.
Z	Matrix or data.frame of covariates.
rep	Number of repetitions with which to repeat the PCM test
est_vhat	Logical; whether to estimate the variance functional
reg_YonXZ	Character string or function specifying the regression for Y on X and Z, default is "rf" for random forest. See ?regressions for more detail.
reg_YonZ	Character string or function specifying the regression for Y on Z, default is "rf" for random forest. See ?regressions for more detail.
reg_YhatonZ	Character string or function specifying the regression for the predicted values of reg_YonXZ on Z, default is "rf" for random forest. See ?regressions for more detail.
reg_VonXZ	Character string or function specifying the regression for estimating the conditional variance of Y given X and Z, default is "rf" for random forest. See ?regressions for more detail.
reg_RonZ	Character string or function specifying the regression for the estimated transformation of Y, X, and Z on Z, default is "rf" for random forest. See ?regressions for more detail.
args_YonXZ	A list of named arguments passed to reg_YonXZ.
args_YonZ	A list of named arguments passed to reg_YonZ.
args_YhatonZ	A list of named arguments passed to reg_YhatonZ.
args_VonXZ	A list of named arguments passed to reg_VonXZ.
args_RonZ	A list of named arguments passed to reg_RonZ.
frac	Relative size of train split.
indices	A numeric vector of indices specifying the observations used for estimating the estimating the direction (the other observations will be used for computing the final test statistic). Default is NULL and the indices will be generated randomly using frac. When using rep larger than 1, a list (of length rep) of indices can be supplied.
coin	Logical; whether or not to use the coin package for computing the test statistic and p-value. The coin package computes variances with n - 1 degrees of freedom. The default is TRUE.
cointrol	List; further arguments passed to independence_test.
return_fitted_models	Logical; whether to return the fitted regressions (default is FALSE).
...	Additional arguments currently ignored.

Details

The projected covariance measure test tests whether the conditional mean of Y given X and Z is independent of X.

Value

Object of class 'pcm' and 'htest' with the following components:

statistic	The value of the test statistic.
p.value	The p-value for the hypothesis
parameter	In case X is multidimensional, this is the degrees of freedom used for the chi-squared test.
hypothesis	Null hypothesis of conditional mean independence.
null.value	Null hypothesis of conditional mean independence.
method	The string "Projected covariance measure test".
data.name	A character string giving the name(s) of the data.
check.data	A data.frame containing the residuals for plotting.
models	List of fitted regressions if return_fitted_models is TRUE.

References

Lundborg, A. R., Kim, I., Shah, R. D., & Samworth, R. J. (2022). The Projected Covariance Measure for assumption-lean variable significance testing. arXiv preprint. doi:10.48550/arXiv.2211.02039

Examples

```
n <- 1e2
X <- matrix(rnorm(2 * n), ncol = 2)
colnames(X) <- c("X1", "X2")
Z <- matrix(rnorm(2 * n), ncol = 2)
colnames(Z) <- c("Z1", "Z2")
Y <- X[, 2]^2 + Z[, 2] + rnorm(n)
(pcm1 <- pcm(Y, X, Z))
```

---

plm_equiv_test *Equivalence test for the parameter in a partially linear model*

Description

Equivalence test for the parameter in a partially linear model

Usage

```
plm_equiv_test(Y, X, Z, from, to, scale = c("plm", "cov", "cor"), ...)
```

Arguments

Y	Vector or matrix of response values.
X	Matrix or data.frame of covariates.
Z	Matrix or data.frame of covariates.
from	Lower bound of the equivalence margin
to	Upper bound of the equivalence margin
scale	Scale on which to specify the equivalence margin. Default "plm" corresponds to the partially linear model parameter described in the details. "cov" corresponds to the conditional covariance and "cor" to conditional correlation which lies in $[-1, 1]$.
...	Further arguments passed to gcm

Details

The partially linear model postulates

$$Y = X\theta + g(Z) + \epsilon,$$

and the target of inference is theta. The target is closely related to the conditional covariance between Y and X given Z:

$$\theta = E[cov(X, Y|Z)]/E[Var(X|Z)].$$

The equivalence test (based on the GCM test) tests $H_0 : \theta \notin [\texttt{from}, \texttt{to}]$ versus $H_1 : \theta \in [\texttt{from}, \texttt{to}]$. Y, X (and theta) can only be one-dimensional. There are no restrictions on Z. The equivalence test can also be performed on the conditional covariance scale directly (using scale = "cov") or on the conditional correlation scale:

$$E[cov(X, Y|Z)]/\sqrt{E[Var(X|Z)]E[Var(Y|Z)]}$$

, using scale = "cor".

Value

Object of class 'gcm' and 'htest'

Examples

```
n <- 150
X <- rnorm(n)
Z <- matrix(rnorm(2 * n), ncol = 2)
colnames(Z) <- c("Z1", "Z2")
Y <- X^2 + Z[, 2] + rnorm(n)
plm_equiv_test(Y, X, Z, from = -1, to = 1)
```

plot.gcm *Plotting methods for COMETs*

Description

Plotting methods for COMETs

Usage

```
## S3 method for class 'gcm'
plot(x, plot = TRUE, ...)

## S3 method for class 'pcm'
plot(x, plot = TRUE, ...)

## S3 method for class 'wgcm'
plot(x, plot = TRUE, ...)
```

Arguments

x	Object of class 'gcm', 'pcm', or 'wgcm'.
plot	Logical; whether to print the plot (default: TRUE).
...	Currently ignored.

rf *Implemented regression methods*

Description

Implemented regression methods

Usage

```
rf(y, x, ...)

survforest(y, x, ...)

qrf(y, x, ...)

lrm(y, x, ...)

glrm(y, x, ...)

lasso(y, x, ...)
```

```
ridge(y, x, ...)

postlasso(y, x, ...)

cox(y, x, ...)

tuned_rf(
  y,
  x,
  max.depths = 1:5,
  mtrys = list(1, function(p) ceiling(sqrt(p)), identity),
  verbose = FALSE,
  ...
)

xgb(y, x, nrounds = 2, verbose = 0, ...)

tuned_xgb(
  y,
  x,
  etas = c(0.1, 0.5, 1),
  max_depths = 1:5,
  nfold = 5,
  nrounds = c(2, 10, 50),
  verbose = 0,
  metrics = list("rmse"),
  ...
)
```

Arguments

y	Vector (or matrix) of response values.
x	Design matrix of predictors.
...	Additional arguments passed to the underlying regression method. In case of "rf", "tuned_rf", "survforest" and "qrf", this is ranger. In case of "lasso" and "ridge", this is glmnet. In case of "cox", this is coxph. In case of "xgb" and "tuned_xgb" this is xgboost.
max.depths	Values for max.depth to tune out-of-bag. See ranger.
mtrys	for mtry to tune out-of-bag. See ranger.
verbose	See xgboost.
nrounds	See xgboost.
etas	Values for eta to cross-validate. See xgboost.
max_depths	Values for max_depth to cross-validate. See xgboost.
nfold	Number of folds for nfold-cross validation.
metrics	See xgboost.

Details

The implemented choices are "rf" for random forests as implemented in ranger, "lasso" for cross-validated Lasso regression (using the one-standard error rule), "ridge" for cross-validated ridge regression (using the one-standard error rule), "cox" for the Cox proportional hazards model as implemented in survival, "qrf" or "survforest" for quantile and survival random forests, respectively. The option "postlasso" option refers to a cross-validated LASSO (using the one-standard error rule) and subsequent OLS regression. The "lrm" option implements a standard linear regression model. The "xgb" and "tuned_xgb" options require the xgboost package.

The "tuned_rf" regression method tunes the mtry and max.depth parameters in ranger out-of-bag. The "tuned_xgb" regression method uses k-fold cross-validation to tune the nrounds, mtry and max_depth parameters in xgb.cv.

New regression methods can be implemented and supplied as well and need the following structure. The regression method "custom_reg" needs to take arguments y, x, ..., fit the model using y and x as matrices and return an object of a user-specified class, for instance, 'custom'. For the GCM test, implementing a residuals.custom method is sufficient, which should take arguments object, response = NULL, data = NULL, ... For the PCM test, a predict.custom method is necessary for out-of-sample prediction and computation of residuals.

---

rgcm *GCM test with pre-computed residuals*

Description

GCM test with pre-computed residuals

Usage

```
rgcm(
  rY,
  rX,
  alternative = "two.sided",
  type = c("quadratic", "max", "scalar"),
  ...
)
```

Arguments

rY	Vector or matrix of response values.
rX	Matrix or data.frame of covariates.
alternative	A character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". Only applies if type = "quadratic" and Y and X are one-dimensional.
type	Type of test statistic, either "quadratic" (default) or "max". If "max" is specified, the p-value is computed based on a bootstrap approximation of the null distribution with B samples.
...	Further arguments passed to independence_test().

Value

Object of class 'gcm' and 'htest' with the following components:

statistic	The value of the test statistic.
p.value	The p-value for the hypothesis
parameter	In case X is multidimensional, this is the degrees of freedom used for the chi-squared test.
hypothesis	String specifying the null hypothesis.
null.value	String specifying the null hypothesis.
method	The string "Generalised covariance measure test".
data.name	A character string giving the name(s) of the data.
rY	Residuals for the Y on Z regression.
rX	Residuals for the X on Z regression.

---

wgcm *Weighted Generalised covariance measure test*

Description

Weighted Generalised covariance measure test

Usage

```
wgcm(
  Y,
  X,
  Z,
  reg_YonZ = "rf",
  reg_XonZ = "rf",
  reg_wfun = "rf",
  args_YonZ = NULL,
  args_XonZ = NULL,
  args_wfun = NULL,
  frac = 0.5,
  B = 499L,
  coin = TRUE,
  cointrol = NULL,
  return_fitted_models = FALSE,
  multivariate = c("none", "YonZ", "XonZ", "both"),
  ...
)
```

Arguments

Y	Vector of response values. Can be supplied as a numeric vector or a single column matrix.
X	Matrix or data.frame of covariates.
Z	Matrix or data.frame of covariates.
reg_YonZ	Character string or function specifying the regression for Y on Z. See ?regressions for more detail.
reg_XonZ	Character string or function specifying the regression for X on Z. See ?regressions for more detail.
reg_wfun	Character string or function specifying the regression for estimating the weighting function. See ?regressions for more detail.
args_YonZ	A list of named arguments passed to reg_YonZ.
args_XonZ	A list of named arguments passed to reg_XonZ.
args_wfun	Additional arguments passed to reg_XonZ.
frac	Relative size of train split.
B	Number of bootstrap samples. Only applies if type = "max" is used.
coin	Logical; whether or not to use the coin package for computing the test statistic and p-value. The coin package computes variances with n - 1 degrees of freedom. The default is TRUE.
cointrol	List; further arguments passed to independence_test.
return_fitted_models	Logical; whether to return the fitted regressions (default is FALSE).
multivariate	Character; specifying which regression can handle multivariate outcomes ("none", "YonZ", "XonZ", or "both"). If "none", then the regression is run using each column in Y (or X) as the response.
...	Additional arguments currently ignored.

Details

The weighted generalised covariance measure test tests whether a weighted version of the conditional covariance of Y and X given Z is zero.

Value

Object of class 'wgcm' and 'htest' with the following components:

statistic	The value of the test statistic.
p.value	The p-value for the hypothesis
parameter	In case X is multidimensional, this is the degrees of freedom used for the chi-squared test.
hypothesis	String specifying the null hypothesis .
null.value	String specifying the null hypothesis.
method	The string "Generalised covariance measure test".

data.name	A character string giving the name(s) of the data.
rY	Residuals for the Y on Z regression.
rX	Weighted residuals for the X on Z regression.
W	Estimated weights.
models	List of fitted regressions if return_fitted_models is TRUE.

References

Scheidegger, C., Hörrmann, J., & Bühlmann, P. (2022). The weighted generalised covariance measure. Journal of Machine Learning Research, 23(273), 1-68.

Examples

```
n <- 100
X <- matrix(rnorm(2 * n), ncol = 2)
colnames(X) <- c("X1", "X2")
Z <- matrix(rnorm(2 * n), ncol = 2)
colnames(Z) <- c("Z1", "Z2")
Y <- X[, 2]^2 + Z[, 2] + rnorm(n)
(wgcm1 <- wgcm(Y, X, Z))
```

Index