Description

Verifies whether an argument matches formal requirements.

Usage

```r
.check(
  x,
  type,
  dim = NULL,
  miss = FALSE,
  min = NULL,
  max = NULL,
  values = NULL,
  inf = FALSE,
  null = FALSE
)
```

Arguments

- `x`: argument
- `type`: character "string","scalar","vector","matrix"
- `dim`: vector/matrix dimensionality: integer scalar/vector
- `miss`: accept missing values: logical
- `min`: lower limit: numeric
- `max`: upper limit: numeric
- `values`: only accept specific values: vector
- `inf`: accept infinite (Inf or -Inf) values: logical
- `null`: accept NULL: logical

Examples

```r
cornet:::.check(0.5,type="scalar",min=0,max=1)
```
.equal  

Description

Verifies whether two or more arguments are identical.

Usage

.equal(..., na.rm = FALSE)

Arguments

...  scalars, vectors, or matrices of equal dimensions
na.rm  remove missing values: logical

Examples

cornet::.equal(1,1,1)

.simulate  

Description

Simulates data for unit tests

Usage

.simulate(n, p, cor = 0, prob = 0.1, sd = 1, exp = 1, frac = 1)

Arguments

n  sample size: positive integer
p  covariate space: positive integer
cor  correlation coefficient: numeric between 0 and 1
prob  effect proportion: numeric between 0 and 1
sd  standard deviation: positive numeric
exp  exponent: positive numeric
frac  class proportion: numeric between 0 and 1
Details

For simulating correlated features ($\text{cor} > 0$), this function requires the R package MASS (see `mvrnorm`).

Value

Returns invisible list with elements y and X.

Examples

data <- cornet:::.simulate(n=10,p=20)
names(data)

Description

Compares models for a continuous response with a cut-off value.

Usage

`.test(y, cutoff, X, alpha = 1, type.measure = "deviance")`

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>continuous outcome: vector of length n</td>
</tr>
<tr>
<td>cutoff</td>
<td>cut-off point for dichotomising outcome into classes: meaningful value between min(y) and max(y)</td>
</tr>
<tr>
<td>X</td>
<td>features: numeric matrix with n rows (samples) and p columns (variables)</td>
</tr>
<tr>
<td>alpha</td>
<td>elastic net mixing parameter: numeric between 0 (ridge) and 1 (lasso)</td>
</tr>
<tr>
<td>type.measure</td>
<td>loss function for binary classification: character &quot;deviance&quot;, &quot;mse&quot;, &quot;mae&quot;, or &quot;class&quot; (see <code>cv.glmnet</code>)</td>
</tr>
</tbody>
</table>

Details

Splits samples into 80 percent for training and 20 percent for testing, calculates squared deviance residuals of logistic and combined regression, conducts the paired one-sided Wilcoxon signed rank test, and returns the $p$-value. For the multi-split test, use the median $p$-value from 50 single-split tests (van de Wiel 2009).

Examples

```r
n <- 100; p <- 200
y <- rnorm(n)
X <- matrix(rnorm(n*p),nrow=n,ncol=p)
cornet:::.test(y=y,cutoff=0,X=X)
```
**coef.cornet**

Extract estimated coefficients

**Description**

Extracts estimated coefficients from linear and logistic regression, under the penalty parameter that minimises the cross-validated loss.

**Usage**

```r
## S3 method for class 'cornet'
coef(object, ...)
```

**Arguments**

- `object`: cornet object
- `...`: further arguments (not applicable)

**Value**

This function returns a matrix with \( n \) rows and two columns, where \( n \) is the sample size. It includes the estimated coefficients from linear regression (1st column: "beta") and logistic regression (2nd column: "gamma").

**Examples**

```r
n <- 100; p <- 200
y <- rnorm(n)
X <- matrix(rnorm(n*p),nrow=n,ncol=p)
net <- cornet(y=y,cutoff=0,X=X)
coef(net)
```

**cornet**

Combined regression

**Description**

Implements lasso and ridge regression for dichotomised outcomes. Such outcomes are not naturally but artificially binary. They indicate whether an underlying measurement is greater than a threshold.
Usage

cornet(
  y,
  cutoff,
  X,
  alpha = 1,
  npi = 101,
  pi = NULL,
  nsigma = 99,
  sigma = NULL,
  nfolds = 10,
  foldid = NULL,
  type.measure = "deviance",
  ...
)

Arguments

y               continuous outcome: vector of length n
cutoff          cut-off point for dichotomising outcome into classes: meaningful value between min(y) and max(y)
X               features: numeric matrix with n rows (samples) and p columns (variables)
alpha           elastic net mixing parameter: numeric between 0 (ridge) and 1 (lasso)
npi             number of pi values (weighting)
pi              pi sequence: vector of increasing values in the unit interval; or NULL (default sequence)
nsigma          number of sigma values (scaling)
sigma           sigma sequence: vector of increasing positive values; or NULL (default sequence)
nfolds          number of folds: integer between 3 and n
foldid          fold identifiers: vector with entries between 1 and nfolds; or NULL (balance)
type.measure    loss function for binary classification: character "deviance", "mse", "mae", or "class" (see cv.glmnet)
...              further arguments passed to glmnet

Details

The argument family is unavailable, because this function fits a gaussian model for the numeric response, and a binomial model for the binary response.

Linear regression uses the loss function "deviance" (or "mse"), but the loss is incomparable between linear and logistic regression.

The loss function "auc" is unavailable for internal cross-validation. If at all, use "auc" for external cross-validation only.
Value

Returns an object of class cornet, a list with multiple slots:

- **gaussian**: fitted linear model, class glmnet
- **binomial**: fitted logistic model, class glmnet
- **sigma**: scaling parameters sigma, vector of length nsigma
- **pi**: weighting parameters pi, vector of length npi
- **cvm**: evaluation loss, matrix with nsigma rows and npi columns
- **sigma.min**: optimal scaling parameter, positive scalar
- **pi.min**: optimal weighting parameter, scalar in unit interval
- **cutoff**: threshold for dichotomisation

References


See Also

Methods for objects of class cornet include coef and predict.

Examples

```r
n <- 100; p <- 200
y <- rnorm(n)
X <- matrix(rnorm(n*p),nrow=n,ncol=p)
net <- cornet(y=y,cutoff=0,X=X)
net
```

---

**cv.cornet**

*Performance measurement*

Description

Compares models for a continuous response with a cut-off value.

Usage

```r
cv.cornet(
  y,
  cutoff,
  X,
  alpha = 1,
  nfolds.ext = 5,
  nfolds.int = 10,
)```
foldid.ext = NULL,
foldid.int = NULL,
type.measure = "deviance",
rf = FALSE,
xgboost = FALSE,
...
}

Arguments

y    continuous outcome: vector of length n
cutoff cut-off point for dichotomising outcome into classes: meaningful value between min(y) and max(y)
X    features: numeric matrix with n rows (samples) and p columns (variables)
alpha elastic net mixing parameter: numeric between 0 (ridge) and 1 (lasso)
nfolds.ext number of external folds
nfolds.int internal fold identifiers: vector of length n with entries between 1 and nfolds.int; or NULL
foldid.ext external fold identifiers: vector of length n with entries between 1 and nfolds.ext; or NULL
foldid.int number of internal folds
type.measure loss function for binary classification: character "deviance", "mse", "mae", or "class" (see cv.glmnet)
rf    comparison with random forest: logical
xgboost comparison with extreme gradient boosting: logical
... further arguments passed to cornet or glmnet

Details

Computes the cross-validated loss of logistic and combined regression.

Examples

```r
## Not run: n <- 100; p <- 200
y <- rnorm(n)
X <- matrix(rnorm(n*p), nrow=n, ncol=p)
start <- Sys.time()
loss <- cv.cornet(y=y, cutoff=0, X=X)
end <- Sys.time()
end - start

loss
## End(Not run)
```
**plot.cornet**  
*Plot loss matrix*

**Description**
Plots the loss for different combinations of scaling (sigma) and weighting (pi) parameters.

**Usage**
```r
## S3 method for class 'cornet'
plot(x, ...)
```

**Arguments**
- `x` *cornet* object
- `...` further arguments (not applicable)

**Value**
This function plots the evaluation loss (cvm). Whereas the matrix has sigma in the rows, and pi in the columns, the plot has sigma on the x-axis, and pi on the y-axis. For all combinations of sigma and pi, the colour indicates the loss. If the R package `RColorBrewer` is installed, blue represents low. Otherwise, red represents low. White always represents high.

**Examples**
```r
n <- 100; p <- 200
y <- rnorm(n)
X <- matrix(rnorm(n*p),nrow=n,ncol=p)
net <- cornet(y=y,cutoff=0,X=X)
plot(net)
```

**predict.cornet**  
*Predict binary outcome*

**Description**
Predicts the binary outcome with linear, logistic, and combined regression.

**Usage**
```r
## S3 method for class 'cornet'
predict(object, newx, type = "probability", ...)
```
Arguments

- **object**: cornet object
- **newx**: covariates: numeric matrix with \( n \) rows (samples) and \( p \) columns (variables)
- **type**: "probability", "odds", "log-odds"
- ... further arguments (not applicable)

Details

For linear regression, this function tentatively transforms the predicted values to predicted probabilities, using a Gaussian distribution with a fixed mean (threshold) and a fixed variance (estimated variance of the numeric outcome).

Examples

```r
n <- 100; p <- 200
y <- rnorm(n)
X <- matrix(rnorm(n*p), nrow=n, ncol=p)
net <- cornet(y=y, cutoff=0, X=X)
predict(net, newx=X)
```

Description

Prints summary of cornet object.

Usage

```r
## S3 method for class 'cornet'
print(x, ...)
```

Arguments

- **x**: cornet object
- ... further arguments (not applicable)

Value

Returns sample size \( n \), number of covariates \( p \), information on dichotomisation, tuned scaling parameter (sigma), tuned weighting parameter (pi), and corresponding loss.
Examples

n <- 100; p <- 200
y <- rnorm(n)
X <- matrix(rnorm(n*p),nrow=n,ncol=p)
net <- cornet(y=y,cutoff=0,X=X)
print(net)
Index

.check, 2
.equal, 3
.simulate, 3
.test, 4

coeff, 7
coeff.cornet, 5
cornet, 5, 5, 8–10
cornet-package (cornet), 5
cv.cornet, 7
cv.glmnet, 4, 6, 8

glmnet, 6, 8

mvrnorm, 4

plot.cornet, 9
predict, 7
predict.cornet, 9
print.cornet, 10