RcppRidge Package Documentation

May 26, 2021

`RcppRidge-package` *A short title line describing what the package does*

Description

A more detailed description of what the package does. A length of about one to five lines is recommended.

Details

This section should provide a more detailed overview of how to use the package, including the most important functions.

Author(s)

Your Name, email optional.

Maintainer: Your Name <your@email.com>

References

This optional section can contain literature or other references for background information.

See Also

Optional links to other man pages

Examples

```
  ## Not run:
     ## Optional simple examples of the most important functions
     ## These can be in \dontrun{} and \donttest{} blocks.

## End(Not run)
```

fit_rr *Fit a single ridge regression model*

Description

Fit a single ridge regression model

Usage

```
fit_rr(X, y, lambda)
```

Arguments

X	Data matrix
y	Column matrix of responses
lambda	Numeric hyperparameter controlling the strength of the L2 penalisation (non-negative)

Value

Vector of penalised regression coefficients

get_ocv *Fast calculate leave one out cross validation error (OCV)*

Description

Fast calculate OCV given a singular value decomposition (SVD) decomposition of data matrix X

Usage

```
get_ocv(X, y, lambda, U, s)
```

Arguments

X	Data matrix
y	Column matrix of responses
lambda	Numeric hyperparameter controlling the strength of the L2 penalisation (non-negative)
U	Matrix U from SVD of X = UDV
s	Elements of diagonal matrix D from SVD of X = UDV

Value

Numeric OCV

get_ocv_once — *Calculate leave one out cross validation error (OCV) for a single regression model*

Description

Calculate leave one out cross validation error (OCV) for a single regression model

Usage

```
get_ocv_once(X, y, lambda)
```

Arguments

X	Data matrix
y	Column matrix of responses
lambda	Numeric hyperparameter controlling the strength of the L2 penalisation (non-negative)

Value

Numeric OCV

k_means — *Our implementation of the k-means algorithm*

Description

Given an data matrix x, samples are clustered into a given number of groups.

Usage

```
k_means(x, centers = 5)
```

Arguments

x	numeric matrix of data, where rows are samples
centers	the number of groups

Value

vector of integers indicating the group allocations

`optim_rr` *Find the optimal regularisation parameter through optimised leave one out cross validation*

Description

Find the optimal regularisation parameter through optimised leave one out cross validation

Usage

```
optim_rr(X, y, lams)
```

Arguments

`X`	Data matrix
`y`	Column matrix of responses
`lams`	Vector of regularisation parameters to test

Value

Vector of OCVs

`par_reg` *Fit a ridge regression model to multiple groups in parallel*

Description

Fit a ridge regression model to multiple groups in parallel

Usage

```
par_reg(X, y, lams, idx)
```

Arguments

`X`	Data matrix
`y`	Column matrix of responses
`lams`	Vector of regularisation parameters to test
`idx`	Vector of sample groups

Value

List with two objects lambdas A vector of the optimal value of lambda for each group betas A matrix where columns are the fitted regression coefficients for each group

pca *Our implementation of principal component analysis (PCA)*

Description

Given an data matrix x, a linear projection is applied to maximise sample variation. The first two prinicpal components are returned.

Usage

```
pca(x, sigma = 1.5)
```

Arguments

x	numeric matrix of data, where rows are samples
sigma	

Value

dataframe of sample projections onto PC1 and PC2

predict_groups *Predict new samples using the results from par_reg*

Description

Predict new samples using the results from par_reg

Usage

```
predict_groups(X, betas, idx)
```

Arguments

X	Data matrix of test samples
betas	Matrix of regression coefficients
idx	Vector of sample groups, corresponding to the columns of betas (e.g. idx=c(1, 3) means betas[,1] will be used to predict X[1,], and betas[,3] will be used to predict X[2,])

Value

Vector of fitted values

predict_rr *Predict new sample responses using a tuned regression model*

Description

Predict new sample responses using a tuned regression model

Usage

```
predict_rr(X, beta)
```

Arguments

X	Data matrix of test samples
beta	Vector of regression coefficients

Value

Vector of fitted values

rcpp_hello_world *Simple function using Rcpp*

Description

Simple function using Rcpp

Usage

```
rcpp_hello_world()
```

Examples

```
## Not run:
rcpp_hello_world()

## End(Not run)
```

rmvn_omp *Sample from a multivariate Gaussian*

Description

Sample from a multivariate Gaussian

Usage

```
rmvn_omp(n, mu, sigma)
```

Arguments

n	Integer number of samples to draw
mu	Vector of means
sigma	Covariance matrix

Value

Matrix of MVN samples (n rows)

spectralClustering *Our implementation of spectral clustering*

Description

Given a data matrix x, samples are clustered into k groups using a spectral (eigen-) decomposition of the graph Laplacian. Uses the implementation of kmeans from this package ‘k_means‘.

Usage

```
spectralClustering(x, c = 1, k = 10)
```

Arguments

x	numeric matrix of data, where rows are samples
c	
k	the number of groups

Value

vector of groups

Index