Package 'kernreg'

October 11, 2026

Type Package

Title Nadaraya-Watson Kernel Regression

Version 1.1

Maintainer Michail Tsagris <mtsagris@uoc.gr>

Depends R (>= 4.0)

Imports Rcpp, RcppParallel, Rfast, Rfast2

LinkingTo Rcpp, RcppParallel

Encoding UTF-8

SystemRequirements GNU make

Description Fast implementation of Nadaraya-Watson kernel regression for either univariate or multivariate responses, with one or more bandwidths. K-fold cross-validation is also performed.

License GPL (>= 2)

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Michail Tsagris [aut, cre],
Christos Adam [aut] (ORCID: <https://orcid.org/0009-0003-3244-7034>)

Repository CRAN

Date/Publication 2026-10-11 16:50:22 UTC

Contents

cross-validation for the kernel regression with Euclidean response data 2
Kernel regression with a numerical response vector or matrix 3

Index **5**

cross-validation for the kernel regression with Euclidean response data

cross-validation for the kernel regression with Euclidean response data

Description

cross-validation for the kernel regression with Euclidean response data.

Usage

```
kernreg.tune(y, x, h = seq(0.1, 1, length = 10), type = "gauss",
nfolds = 10, folds = NULL, seed = NULL, graph = FALSE, ncores = 1)
```

Arguments

y	A matrix or a vector with the Euclidean response.
x	A matrix with the available predictor variables.
h	A vector with the bandwidth value(s) h to consider.
type	The type of kernel to use, "gauss" or "laplace".
nfolds	The number of folds. Set to 10 by default.
folds	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
seed	You can specify your own seed number here or leave it NULL.
graph	If graph is TRUE (default value) a plot will appear.
ncores	The number of cores to use. Default value is 1.

Details

A k-fold cross-validation for the kernel regression with a euclidean response is performed.

Value

A list including:

mspe	The mean squared prediction error (MSPE) for each fold and value of h.
h	The optimal h that leads to the minimum MSPE.
performance	The minimum MSPE.
runtime	The runtime of the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Wand M. P. and Jones M. C. (1994). Kernel smoothing. CRC press.

See Also

kern_reg

Examples

```
y <- iris[, 1]
x <- iris[, 2:4]
mod <- kernreg.tune(y, x, h = c(0.1, 0.2, 0.3) )
```

`Kernel regression with a numerical response vector or matrix`
Kernel regression with a numerical response vector or matrix

Description

Kernel regression (Nadaraya-Watson estimator) with a numerical response vector or matrix.

Usage

```
kern_reg(xnew, y, x, h = as.numeric( c(0.1, 0.2, 0.3, 0.4, 0.5,
0.6, 0.7, 0.8, 0.9, 1.0 )), type = "gauss", ncores = 1L)
```

Arguments

xnew	A matrix with the new predictor variables whose compositions are to be predicted.
y	A numerical vector or a matrix with the response value.
x	A matrix with the available predictor variables.
h	The bandwidth value(s) to consider.
type	The type of kernel to use, "gauss" or "laplace".
ncores	The number of cores to use. If greater than 1, parallel computing will take place. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down the process. The default is 1, meaning that code is executed serially.

Details

The Nadaraya-Watson estimator regression is applied.

Value

The fitted values. If a single bandwidth is considered then this is a vector or a matrix, depeding on the nature of the response. If multiple bandwidth values are considered then this is a matrix, if the response is a vector, or a list, if the response is a matrix.

Author(s)

Michail Tsagris and Christos Adam.

References

Wand M. P. and Jones M. C. (1994). Kernel smoothing. CRC press.

See Also

kernreg.tune

Examples

```
y <- iris[, 1]
x <- iris[, 2:4]
est <- kern_reg(x, y, x, h = c(0.1, 0.2) )
```

Index

cross-validation for the kernel
regression with Euclidean
response data, 2

kern_reg, *3*
kern_reg *(*Kernel regression with a
numerical response vector or
matrix*)*, 3
Kernel regression with a numerical
response vector or matrix, 3
kernreg.tune, *4*
kernreg.tune *(*cross-validation for the
kernel regression with
Euclidean response data*)*, 2