Package 'kernda'

November 19, 2026

Type Package

Title Kernel Discriminant Analysis

Version 1.1

Date 2026-11-19

Author Michail Tsagris [aut, cre],
Nikolaos Kontemeniotis [aut]

Maintainer Michail Tsagris <mtsagris@uoc.gr>

Depends R (>= 4.0)

Imports Compositional, Rfast, Rcpp, kernreg

LinkingTo Rcpp

Description Functions to perform discriminant analysis using kernel density estimation. For the case of binary classification the package relies on the Nadaraya-Watson estimator. References: Wand M.P. and Jones M.C. (1995). ``Kernel smoothing". CRC Press. <ISBN: 0412552701>.

License GPL (>= 2)

RoxygenNote 7.3.3

NeedsCompilation yes

Repository CRAN

Date/Publication 2026-11-19 07:40:02 UTC

Contents

kernda-package 2
cv.kdeda 2
cv.lrkde 3
cv.nbkde 5
kde_da 6
lr.kde 7
mlr.kde 9
nb.kde 10

Index **12**

kernda-package *Kernel Discriminant Analysis*

Description

Functions to perform discriminant analysis using kernel density estimation. For the case of binary classification the package relies on the Nadaraya-Watson estimator. References: Wand M.P. and Jones M.C. (1995). Kernel smoothing. CRC Press.

Details

Package:	kernda
Type:	Package
Version:	1.1
Date:	2026-11-19

Maintainers

Michail Tsagris <mtsagris@uoc.gr>.

Author(s)

Michail Tsagris <mtsagris@uoc.gr> and Nikolaos Kontemeniotis <kontemeniotisn@gmail.com>

cv.kdeda *Cross-validation for the kernel discriminant analysis*

Description

Cross-validation for the kernel discriminant analysis.

Usage

```
cv.kdeda(groups, x, h = seq(0.1, 1, length = 10), nfolds = 10, folds = NULL, ncores = 1)
```

Arguments

groups	A variable indicating the groupings. Any labels are allowed, but not missing values.
x	A numerical vector with the data. Missing values are not allowed.
h	A vector with the bandwidth parameter values of the Gaussian kernel.
nfolds	How many folds to create? This is ignored if folds is supplied.
folds	Do you already have a list with the folds? If not, leave this NULL.
ncores	The number of cores to use.

Details

A k-fold cross-validation of kde_da is performed, with stratified folds, and the proportion of correct classification is computed for each bandwidth value. The bandwidth is on the scale of x, so the default values may need to be changed for data on a different scale. If many bandwidth values have the same performance, the first of them is returned.

Value

A list including:

cv	A numerical matrix with the proportion of correct classification for each fold and bandwidth value.
perf	The average proportion of correct classification for each bandwidth value.
hopt	The optimal bandwidth value.

Author(s)

Michail Tsagris and Nikolaos Kontemeniotis.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Nikolaos Kontemeniotis <kontemeniotisn@gmail.com>.

References

Wand M.P. and Jones M.C. (1995). Kernel smoothing. CRC Press.

Hastie T., Tibshirani R. and Friedman J. (2009). The elements of statistical learning.

See Also

kde_da, lr.kde

Examples

```
x <- iris[, 4]
groups <- as.numeric( iris[, 5] )
a <- cv.kdeda(groups, x)
```

---

cv.lrkde *Cross-validation for the kernel regression with a binary response*

Description

Cross-validation for the kernel regression with a binary response.

Usage

```
cv.lrkde(groups, x, h = seq(0.1, 1, length = 10), nfolds = 10,
folds = NULL, graph = FALSE, ncores = 1)
```

Arguments

groups	A vector with the binary response (0s and 1s).
x	A numeric matrix, data.frame or vector with the available predictor variables. Missing values are not allowed.
h	A vector with the bandwidth value(s) h to consider.
nfolds	The number of folds. Set to 10 by default. This is ignored if folds is supplied.
folds	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
graph	If graph is TRUE a plot will appear. The default value is FALSE.
ncores	The number of cores to use. Default value is 1.

Details

A k-fold cross-validation for the kernel regression with a binary response is performed and the AUC is computed. The predictor variables are standardised by the function `kern_reg` of the package **kernreg**, using the training sample, so the bandwidth is in standard deviation units. A test fold containing one class only has no AUC and it is ignored when averaging.

Value

A list including:

cv	A numerical matrix with the AUC for each fold and bandwidth value.
perf	The average AUC for each bandwidth value.
hopt	The optimal bandwidth value.

Author(s)

Michail Tsagris.

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

References

Wand M.P. and Jones M.C. (1995). Kernel smoothing. CRC Press.

See Also

lr.kde

Examples

```
groups <- as.numeric( iris[1:100, 5] ) - 1
x <- iris[1:100, 2:4]
cv.lrkde(groups, x)
```

cv.nbkde *Cross-validation for the naive Bayes kernel discriminant analysis*

Description

Cross-validation for the naive Bayes kernel discriminant analysis.

Usage

```
cv.nbkde(groups, x, nfolds = 10, folds = NULL, ncores = 1L)
```

Arguments

groups	A variable indicating the groupings. Any labels are allowed, but not missing values.
x	A numerical vector with the data. Missing values are not allowed.
nfolds	How many folds to create? This is ignored if folds is supplied.
folds	Do you already have a list with the folds? If not, leave this NULL.
ncores	The number of cores to use.

Details

A k-fold cross-validation of kde_da is performed, with stratified folds, and the proportion of correct classification is computed for each bandwidth value. The bandwidth is on the scale of x, so the default values may need to be changed for data on a different scale. If many bandwidth values have the same performance, the first of them is returned.

Value

A list including:

cv	A numerical matrix with the proportion of correct classification for each fold and bandwidth value.
perf	The average proportion of correct classification for each bandwidth value.

Author(s)

Michail Tsagris.

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

References

Wand M.P. and Jones M.C. (1995). Kernel smoothing. CRC Press.

Hastie T., Tibshirani R. and Friedman J. (2009). The elements of statistical learning.

See Also

nb.kde, kde_da

Examples

```
x <- iris[, 1:4]
groups <- as.numeric( iris[, 5] )
a <- cv.nbkde(groups, x)
```

---

kde_da *Kernel discriminant analysis*

Description

Kernel discriminant analysis for a univariate predictor.

Usage

```
kde_da(xnew, groups, x, h = NULL, ncores = 1L, logged = FALSE)
```

Arguments

xnew	A numeric vector with the new values whose group is to be predicted. Missing values are allowed.
groups	A vector with the groups (labels) of the data, given as positive integers (1, 2, ...) or as a factor. Zeros, negative values and missing values are not allowed.
x	A numeric vector with continuous data, of the same length as groups. Missing values are not allowed.
h	A vector with one or more positive values for the bandwidth parameter of the Gaussian kernel. Each value is used for all groups. If h is NULL (the default), each group gets its own bandwidth, computed with bw.nrd. Each group must then have at least 2 observations and a non-zero bandwidth.
ncores	The number of cores to use. It has an effect only if the package was compiled with OpenMP support.
logged	Should the logarithm of the kernel density estimates be returned? The default is FALSE. This argument does not affect the predicted groups.

Details

The density of each group is estimated with a Gaussian kernel, using either the same bandwidth for all groups or, if h is NULL, the bandwidth given by bw.nrd for each group, and each new value is allocated to the group with the highest estimated posterior probability, using the sample proportions as prior probabilities. The bandwidth is on the scale of x, since x is not standardised. For values of xnew very far from the data the estimated densities may underflow to 0, but the predicted groups are still computed correctly. With logged = TRUE the log-densities are computed without this underflow.

Value

A list including:

class	An integer matrix with the predicted groups. It has one row for each value of xnew and one column for each value of h, or a single column if h is NULL. Missing values of xnew receive NA.
kde	A list with one matrix for each value of h, even if h is a single number. The names of the list are the values of h. If h is NULL the list has one matrix, named "bw.nrd". Each matrix has one row for each value of xnew and one column for each group, and it contains the kernel density estimates of the groups at xnew, or their logarithms if logged is TRUE. A group number without observations receives 0, or -Inf if logged is TRUE. Missing values of xnew receive NA.
h	The bandwidth values used. If h is NULL, this is a named vector with the bandwidth of each group, with NA for a group number without observations. Otherwise it is the supplied h.

Author(s)

Michail Tsagris and Nikolaos Kontemeniotis.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Nikolaos Kontemeniotis <kontemeniotisn@gmail.com>.

References

Wand M.P. and Jones M.C. (1995). Kernel smoothing. CRC Press.

See Also

cv.kdeda

Examples

```
x <- iris[, 4]
groups <- as.numeric(iris[, 5])
a <- kde_da(x, groups, x, h = c(0.1, 0.2))
b <- kde_da(x, groups, x, h = NULL)
d <- kde_da(x, groups, x, h = NULL, logged = TRUE)
```

---

lr.kde *Kernel regression with a binary response*

Description

Kernel regression with a binary response.

Usage

```
lr.kde(xnew, groups, x, h, ncores = 1)
lr.kde2(groups, x, h, nfolds = 10, folds = NULL, ncores = 1)
lr.kde3(groups, x, h = seq(0.01, 2, by = 0.01), nfolds = 10,
folds = NULL, ncores = 1)
```

Arguments

xnew	A numeric matrix, data.frame or vector with the new predictor variables, whose class probabilities are to be estimated.
groups	The binary response values (0s and 1s). This is a vector, while lr.kde() also accepts a matrix with many binary responses.
x	A numeric matrix, data.frame or vector with the available predictor variables. Missing values are not allowed.
h	For lr.kde() the bandwidth value(s) to use. For lr.kde2() a vector with the lower and the upper end of the interval to search. For lr.kde3() the bandwidth values to consider.
nfolds	The number of folds used to choose the bandwidth. This is ignored if folds is supplied.
folds	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
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 is applied. The predictor variables are standardised by the function kern_reg of the package **kernreg**, so the bandwidth is in standard deviation units.

The function lr.kde2() chooses the bandwidth value that maximizes the cross-validated AUC, using the command optimize(). The AUC is a step function of the bandwidth and it may have many local maxima, so optimize() may not find the global maximum. The function lr.kde3() computes the cross-validated AUC for given bandwidth values and plots the results. Cross-validation is used because the AUC computed on the training data favours ever smaller bandwidth values and approaches 1, even for pure noise.

Value

For lr.kde() a list including:

prob	The estimated probabilities of belonging to class 1.
cl	The estimated classes.

For lr.kde2() a vector with the optimal bandwidth (hopt) and its cross-validated AUC (auc).

For lr.kde3() a list including:

perf	The cross-validated AUC for each bandwidth value.
hopt	The optimal bandwidth value.

Author(s)

Michail Tsagris and Nikolaos Kontemeniotis.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Nikolaos Kontemeniotis <kontemeniotisn@gmail.com>.

References

Wand M.P. and Jones M.C. (1995). Kernel smoothing. CRC Press.

See Also

cv.lrkde, mlr.kde

Examples

```
groups <- as.numeric( iris[51:150, 5] ) - 2
x <- iris[51:150, 2:4]
est <- lr.kde(x, groups, x, h = c(0.1, 0.2) )
lr.kde2(groups, x, h = c(0.05, 2), nfolds = 5)
lr.kde3(groups, x, h = seq(0.1, 1, by = 0.1), nfolds = 5)
```

---

mlr.kde *Kernel regression with a multinomial response*

Description

Kernel regression with a multinomial response.

Usage

```
mlr.kde(xnew, groups, x, h, ncores = 1)
```

Arguments

xnew	A numeric matrix, data.frame or vector with the new predictor variables, whose class probabilities are to be estimated.
groups	The multinomial response values (either a factor or a numerical vector).
x	A numeric matrix, data.frame or vector with the available predictor variables. Missing values are not allowed.
h	The bandwidth value(s) to use.
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 is applied. The predictor variables are standardised by the function `kern_reg` of the package **kernreg**, so the bandwidth is in standard deviation units.

The function lr.kde2() chooses the bandwidth value that maximizes the cross-validated AUC, using the command optimize(). The AUC is a step function of the bandwidth and it may have many local maxima, so optimize() may not find the global maximum. The function lr.kde3() computes the cross-validated AUC for given bandwidth values and plots the results. Cross-validation is used because the AUC computed on the training data favours ever smaller bandwidth values and approaches 1, even for pure noise.

Value

A list including:

`prob`	The estimated probabilities of belonging to class 1.
`cl`	The estimated classes.

Author(s)

Michail Tsagris.

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

References

Wand M.P. and Jones M.C. (1995). Kernel smoothing. CRC Press.

See Also

`lr.kde`

Examples

```
groups <- iris[, 5]
x <- iris[, 1:4]
est <- mlr.kde(x, groups, x, h = c(0.1, 0.2) )
```

---

nb.kde *Kernel naive Bayes discriminant analysis*

Description

Kernel naive Bayes discriminant analysis.

Usage

```
nb.kde(xnew, groups, x, h = NULL, ncores = 1L)
```

Arguments

xnew	A numeric vector with the new values whose group is to be predicted. Missing values are allowed.
groups	A vector with the groups (labels) of the data, given as positive integers (1, 2, ...) or as a factor. Zeros, negative values and missing values are not allowed.
x	A numeric vector with continuous data, of the same length as groups. Missing values are not allowed.
h	A vector with one or more positive values for the bandwidth parameter of the Gaussian kernel.
ncores	The number of cores to use. It has an effect only if the package was compiled with OpenMP support.

Details

The density of each group is estimated with a Gaussian kernel, using the same bandwidth for all groups, and each new value is allocated to the group with the highest estimated posterior probability, using the sample proportions as prior probabilities. The bandwidth is on the scale of x, since x is not standardised. For values of xnew very far from the data the estimated densities may underflow to 0, but the predicted groups are still computed correctly.

Value

A list including:

class	A vector with the predicted groups, one for each row of xnew.
score	A matrix with one row for each row of xnew and one column for each group. It contains the sum of the logarithms of the kernel density estimates over the variables plus the logarithm of the prior probability of each group, that is, the logarithm of the unnormalised posterior probabilities.

Author(s)

Michail Tsagris.

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

References

Wand M.P. and Jones M.C. (1995). Kernel smoothing. CRC Press.

See Also

kde_da

Examples

```
x <- iris[, 1:4]
groups <- as.numeric(iris[, 5])
a <- nb.kde(x, groups, x, h = NULL )
```

Index

bw.nrd, *6*

cv.kdeda, 2, *7*
cv.lrkde, 3, *9*
cv.nbkde, 5

kde_da, *3*, *5*, *6*, 6, *11*
kernda-package, 2

lr.kde, *3*, *4*, 7, *10*
lr.kde2 *(*lr.kde*)*, 7
lr.kde3 *(*lr.kde*)*, 7

mlr.kde, *9*, 9

nb.kde, *6*, 10