Package ‘pvclass’

June 5, 2017

Type Package
Title P-Values for Classification
Version 1.4
Date 2017-06-05
Author Niki Zumbrunnen <niki.zumbrunnen@gmail.com>,
Lutz Duembgen <lutz.duembgen@stat.unibe.ch>.
Maintainer Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
Imports Matrix
Description Computes nonparametric p-values for the potential class
memberships of new observations as well as cross-validated
p-values for the training data. The p-values are based on
permutation tests applied to an estimated Bayesian likelihood
ratio, using a plug-in statistic for the Gaussian model, 'k
nearest neighbors', 'weighted nearest neighbors' or
'penalized logistic regression'.
Additionally, it provides graphical displays and quantitative
analyses of the p-values.
License GPL (>= 2)
LazyLoad yes
NeedsCompilation no
Repository CRAN
Date/Publication 2017-06-05 15:13:51 UTC

R topics documented:

   pvclass-package .............................................. 2
   analyze.pvs .................................................. 3
   buerk ......................................................... 4
   cvpvs .......................................................... 6
   cvpvs.gaussian ............................................. 7
   cvpvs.knn ..................................................... 8
   cvpvs.logreg ................................................ 10
Description

Computes nonparametric p-values for the potential class memberships of new observations as well as cross-validated p-values for the training data. The p-values are based on permutation tests applied to an estimated Bayesian likelihood ratio, using a plug-in statistic for the Gaussian model, 'k nearest neighbors', 'weighted nearest neighbors' or 'penalized logistic regression'. Additionally, it provides graphical displays and quantitative analyses of the p-values.

Details

Use `cvpvs` to compute cross-validated p-values, `pvs` to classify new observations and `analyze.pvs` to analyze the p-values.

Author(s)

Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
Lutz Dümbgen <lutz.duembgen@stat.unibe.ch>
www.imsv.unibe.ch/duembgen/index_ger.html

References


Examples

```r
NewX <- iris[c(50, 100, 150), 1:4]

cv <- cvpvs(X, Y)
analyze.pvs(cv, Y)
```
analyze.pvs

```
pv <- pvs(NewX, X, Y, method = 'k', k = 10)
analyze.pvs(pv)
```

### Description

Graphical displays and quantitative analyses of a matrix of p-values.

### Usage

```
analyze.pvs(pv, Y = NULL, alpha = 0.05, roc = TRUE, pvplot = TRUE, cex = 1)
```

### Arguments

- **pv**: matrix with p-values, e.g. output of `cvpvs` or `pvs`.
- **Y**: optional. Vector indicating the classes which the observations belong to.
- **alpha**: test level, i.e. 1 - confidence level.
- **roc**: logical. If `TRUE` and `Y` is not `NULL`, ROC curves are plotted.
- **pvplot**: logical. If `TRUE` or `Y` is `NULL`, the p-values are displayed graphically.
- **cex**: A numerical value giving the amount by which plotting text should be magnified relative to the default.

### Details

Displays the p-values graphically, i.e. it plots for each p-value a rectangle. The area of this rectangle is proportional to the p-value. The rectangle is drawn blue if the p-value is greater than `alpha` and red otherwise.

If `Y` is not `NULL`, i.e. the class memberships of the observations are known (e.g. cross-validated p-values), then additionally it plots the empirical ROC curves and prints some empirical conditional inclusion probabilities \( I(b, \theta) \) and/or pattern probabilities \( P(b, S) \). Precisely, \( I(b, \theta) \) is the proportion of training observations of class \( b \) whose p-value for class \( \theta \) is greater than \( \alpha \), while \( P(b, S) \) is the proportion of training observations of class \( b \) such that the \((1 - \alpha)\)-prediction region equals \( S \).

### Value

- **T**: Table containing empirical conditional inclusion and/or pattern probabilities for each class \( b \). In case of \( L = 2 \) or \( L = 3 \) classes, all patterns \( S \) are considered. In case of \( L > 3 \), all inclusion probabilities and some special patterns \( S \) are considered.

### Author(s)

Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
Lutz Dümbgen <lutz.duembgen@stat.unibe.ch>
[www.imsv.unibe.ch/duembgen/index_ger.html](http://www.imsv.unibe.ch/duembgen/index_ger.html)
References


See Also

cvpvs, pvs

Examples

```r
NewX <- iris[c(50, 100, 150), 1:4]

cv <- cvpvs(X, Y)
analyze.pvs(cv, Y)

pv <- pvs(NewX, X, Y, method = 'k', k = 10)
analyze.pvs(pv)
```

buerk

*Medical Dataset*

Description

This data set collected by Dr. Bürk at the university hospital in Lübeck contains data of 21556 surgeries in a certain time period (end of the nineties). Besides the mortality and the morbidity it contains 21 variables describing the condition of the patient and the surgery.

Usage

data(buerk)

Format

A data frame with 21556 observations on the following 23 variables.

- **age** Age in years
- **sex** Sex (1 = female, 0 = male)
ASA Score (American Society of Anesthesiologists), describes the physical condition on an ordinal scale:
1 = A normal healthy patient
2 = A patient with mild systemic disease
3 = A patient with severe systemic disease
4 = A patient with severe systemic disease that is a constant threat to life
5 = A moribund patient who is not expected to survive without the operation
6 = A declared brain-dead patient whose organs are being removed for donor purposes

rf_cer Risk factor: cerebral (1 = yes, 0 = no)
rf_car Risk factor: cardiovascular (1 = yes, 0 = no)
rf_pul Risk factor: pulmonary (1 = yes, 0 = no)
rf_ren Risk factor: renal (1 = yes, 0 = no)
rf_hep Risk factor: hepatic (1 = yes, 0 = no)
rf_imu Risk factor: immunological (1 = yes, 0 = no)
rf_metab Risk factor: metabolic (1 = yes, 0 = no)
rf_noc Risk factor: uncooperative, unreliable (1 = yes, 0 = no)
e_malign Etiology: malignant (1 = yes, 0 = no)
e_vasc Etiology: vascular (1 = yes, 0 = no)
antibio Antibiotics therapy (1 = yes, 0 = no)
op Surgery indicated (1 = yes, 0 = no)
opacute Emergency operation (1 = yes, 0 = no)
optime Surgery time in minutes
opsepsis Septic surgery (1 = yes, 0 = no)
opskill Experienced surgeon, i.e. senior physician (1 = yes, 0 = no)
blood Blood transfusion necessary (1 = yes, 0 = no)
icu Intensive care necessary (1 = yes, 0 = no)
mortal Mortality (1 = yes, 0 = no)
morb Morbidity (1 = yes, 0 = no)

Source

References
**Description**

Computes cross-validated nonparametric p-values for the potential class memberships of the training data.

**Usage**

```
cvpvs(X, Y, method = c('gaussian','knn','wnn', 'logreg'), ...)  
```

**Arguments**

- **X** matrix containing training observations, where each observation is a row vector.
- **Y** vector indicating the classes which the training observations belong to.
- **method** one of the following methods:
  - 'gaussian': plug-in statistic for the standard Gaussian model,
  - 'knn': k nearest neighbors,
  - 'wnn': weighted nearest neighbors,
  - 'logreg': multicategory logistic regression with l1-penalization.
- `...` further arguments depending on the method (see `cvpvs.gaussian`, `cvpvs.knn`, `cvpvs.wnn`, `cvpvs.logreg`).

**Details**

Computes cross-validated nonparametric p-values for the potential class memberships of the training data. Precisely, for each feature vector $X[i,]$ and each class $b$ the number $PV[i, b]$ is a p-value for the null hypothesis that $Y[i] = b$.

This p-value is based on a permutation test applied to an estimated Bayesian likelihood ratio, using a plug-in statistic for the Gaussian model, 'k nearest neighbors', 'weighted nearest neighbors' or multicategory logistic regression with l1-penalization (see `cvpvs.gaussian`, `cvpvs.knn`, `cvpvs.wnn`, `cvpvs.logreg`) with estimated prior probabilities $N(b)/n$. Here $N(b)$ is the number of observations of class $b$ and $n$ is the total number of observations.

**Value**

$PV$ is a matrix containing the cross-validated p-values. Precisely, for each feature vector $X[i,]$ and each class $b$ the number $PV[i, b]$ is a p-value for the null hypothesis that $Y[i] = b$.

**Author(s)**

Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
Lutz Dümbgen <lutz.duembgen@stat.unibe.ch>
www.imsv.unibe.ch/duembgen/index_ger.html
References


See Also

cvpvs.gaussian, cvpvs.knn, cvpvs.wnn, cvpvs.logreg, pvs, analyze.pvs

Examples

```r
X <- iris[,1:4]
Y <- iris[,5]
cvpvs(X,Y,method='k',k=10,distance='d')
```

cvpvs.gaussian

*Cross-Validated P-Values (Gaussian)*

Description

Computes cross-validated nonparametric p-values for the potential class memberships of the training data. The p-values are based on a plug-in statistic for the standard Gaussian model. The latter means that the conditional distribution of $X$, given $Y = y$, is Gaussian with mean depending on $y$ and a global covariance matrix.

Usage

```r
cvpvs.gaussian(X, Y, cova = c('standard', 'M', 'sym'))
```

Arguments

- **X** matrix containing training observations, where each observation is a row vector.
- **Y** vector indicating the classes which the training observations belong to.
- **cova** estimator for the covariance matrix:
  - ’standard’: standard estimator,
  - ’M’: M-estimator,
  - ’sym’: symmetrized M-estimator.
Details

Computes cross-validated nonparametric p-values for the potential class memberships of the training data. Precisely, for each feature vector $X[i,]$ and each class $b$ the number $PV[i, b]$ is a p-value for the null hypothesis that $Y[i] = b$. This p-value is based on a permutation test applied to an estimated Bayesian likelihood ratio, using a plug-in statistic for the standard Gaussian model with estimated prior probabilities $N(b)/n$. Here $N(b)$ is the number of observations of class $b$ and $n$ is the total number of observations.

Value

$PV$ is a matrix containing the cross-validated p-values. Precisely, for each feature vector $X[i,]$ and each class $b$ the number $PV[i, b]$ is a p-value for the null hypothesis that $Y[i] = b$.

Author(s)

Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
Lutz Dümbgen <lutz.dueembgen@stat.unibe.ch>
www.imsv.unibe.ch/duembgen/index_ger.html

References


See Also

cvpvs, cvpvs.knn, cvpvs.wnn, cvpvs.logreg

Examples

```r
X <- iris[, 1:4]
Y <- iris[, 5]

cvpvs.gaussian(X, Y, cova = 'standard')
```

Description

Computes cross-validated nonparametric p-values for the potential class memberships of the training data. The p-values are based on ‘k nearest neighbors’.
Usage

```
cvpvs.knn(x, y, k = NULL, distance = c('euclidean', 'ddeuclidean',
    'mahalanobis'), cova = c('standard', 'M', 'sym'))
```

Arguments

- **x**: matrix containing training observations, where each observation is a row vector.
- **y**: vector indicating the classes which the training observations belong to.
- **k**: number of nearest neighbors. If `k` is a vector or `k = NULL`, the program searches for the best `k`. For more information see section 'Details'.
- **distance**: the distance measure:
  - "euclidean": fixed Euclidean distance,
  - "ddeuclidean": data driven Euclidean distance (component-wise standardization),
  - "mahalanobis": Mahalanobis distance.
- **cova**: estimator for the covariance matrix:
  - 'standard': standard estimator,
  - 'M': M-estimator,
  - 'sym': symmetrized M-estimator.

Details

Computes cross-validated nonparametric p-values for the potential class memberships of the training data. Precisely, for each feature vector `X[i,]` and each class `b` the number `PV[i,b]` is a p-value for the null hypothesis that `Y[i] = b`. This p-value is based on a permutation test applied to an estimated Bayesian likelihood ratio, using 'k nearest neighbors' with estimated prior probabilities `N(b)/n`. Here `N(b)` is the number of observations of class `b` and `n` is the total number of observations.

If `k` is a vector, the program searches for the best `k`. To determine the best `k` for the p-value `PV[i,b]`, the class label of the training observation `X[i,]` is set temporarily to `b` and then for all training observations with `Y[j] != b` the proportion of the `k` nearest neighbors of `X[j,]` belonging to class `b` is computed. Then the `k` which minimizes the sum of these values is chosen.

If `k = NULL`, it is set to `2:ceiling(length(Y)/2)`.

Value

- `PV` is a matrix containing the cross-validated p-values. Precisely, for each feature vector `X[i,]` and each class `b` the number `PV[i,b]` is a p-value for the null hypothesis that `Y[i] = b`. If `k` is a vector or `NULL`, `PV` has an attribute "opt.k", which is a matrix and `opt.k[i,b]` is the best `k` for observation `X[i,]` and class `b` (see section 'Details'). `opt.k[i,b]` is used to compute the p-value for observation `X[i,]` and class `b`.

Author(s)

- Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
- Lutz Dümbgen <lutz.duembgen@stat.unibe.ch>
- [www.imsv.unibe.ch/duembgen/index_ger.html](http://www.imsv.unibe.ch/duembgen/index_ger.html)
References

See Also
`cvpvs, cvpvs.gaussian, cvpvs.wnn, cvpvs.logreg`

Examples
```r
X <- iris[, 1:4]
Y <- iris[, 5]
cvpvs.knn(X, Y, k = c(5, 10, 15))
```

---

**cvpvs.logreg** 
*Cross-Validated P-Values (Penalized Multicategory Logistic Regression)*

Description
Computes cross-validated nonparametric p-values for the potential class memberships of the training data. The p-values are based on 'penalized logistic regression'.

Usage
```r
cvpvs.logreg(X, Y, tau.o=10, find.tau=FALSE, delta=2, tau.max=80, tau.min=1, pen.method = c("vectors", "simple", "none"), progress = TRUE)
```

Arguments
- `X` matrix containing training observations, where each observation is a row vector.
- `Y` vector indicating the classes which the training observations belong to.
- `tau.o` the penalty parameter (see section 'Details' below).
- `find.tau` logical. If TRUE the program searches for the best tau. For more information see section 'Details'.
- `delta` factor for the penalty parameter. Should be greater than 1. Only needed if find.tau == TRUE.
- `tau.max` maximal penalty parameter considered. Only needed if find.tau == TRUE.
- `tau.min` minimal penalty parameter considered. Only needed if find.tau == TRUE.
- `pen.method` the method of penalization (see section 'Details' below).
- `progress` optional parameter for reporting the status of the computations.
Details

Computes cross-validated nonparametric p-values for the potential class memberships of the training data. Precisely, for each feature vector \( X[i,] \) and each class \( b \) the number \( PV[i, b] \) is a p-value for the null hypothesis that \( Y[i] = b \), based on the remaining training observations.

This p-value is based on a permutation test applied to an estimated Bayesian likelihood ratio, using 'penalized logistic regression'. This means, the conditional probability of \( Y = y \), given \( X = x \), is assumed to be proportional to \( \exp(a_y + b_y^T x) \). The parameters \( a_y, b_y \) are estimated via penalized maximum log-likelihood. The penalization is either a weighted sum of the euclidean norms of the vectors \( (b_1[j], b_2[j], \ldots, b_L[j]) \) (pen.method='vectors') or a weighted sum of all moduli \( |b_y[j]| \) (pen.method='simple'). The weights are given by \( \tau \) times the sample standard deviation (within groups) of the \( j \)-th components of the feature vectors. In case of pen.method='none', no penalization is used, but this option may be unstable.

If find.tau == TRUE, the program searches for the best penalty parameter. To determine the best parameter \( \tau \) for the p-value \( PV[i, b] \), the class label of the training observation \( X[i,] \) is temporarily set to \( b \) and then for all training observations with \( Y[j] \neq b \) the estimated probability of \( X[j,] \) belonging to class \( b \) is computed. Then the \( \tau \) which minimizes the sum of these values is chosen. First, \( \tau.o \) is compared with \( \tau.o+\delta \) \( \tau \). If \( \tau.o+\delta \) is better, it is compared with \( \tau.o+\delta^2 \), etc. The maximal parameter considered is \( \tau_{max} \). If \( \tau.o \) is better than \( \tau.o+\delta \), it is compared with \( \tau.o+\delta^2 \), etc. The minimal parameter considered is \( \tau_{min} \).

Value

\( PV \) is a matrix containing the cross-validated p-values. Precisely, for each feature vector \( X[i,] \) and each class \( b \) the number \( PV[i, b] \) is a p-value for the null hypothesis that \( Y[i] = b \), based on the remaining training observations.

If find.tau == TRUE, \( PV \) has an attribute "tau.opt", which is a matrix and \( \tau.o[i, b] \) is the best \( \tau \) for observation \( X[i,] \) and class \( b \) (see section 'Details'). \( \tau.o[i, b] \) is used to compute the p-value for observation \( X[i,] \) and class \( b \).

Author(s)

Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
Lutz Dümbgen <lutz.duebgen@stat.unibe.ch>
www.imsv.unibe.ch/duembgen/index_ger.html

References


See Also

cvpvs, cvpvs.gaussian, cvpvs.knn, cvpvs.wnn
Examples

```r
## not run:
x <- iris[, 1:4]
y <- iris[, 5]
cvpvs.logreg(x, y, tau.o=1, pen.method="vectors", progress=TRUE)

## End(not run)

# A bigger data example: Buerk's hospital data.
## not run:
data(buerk)
x.raw <- as.matrix(buerk[,1:21])
y.raw <- buerk[,22]
n0.raw <- sum(1 - y.raw)
n1 <- sum(y.raw)
n0 <- 3*n1

X0 <- x.raw[y.raw==0,]
X1 <- x.raw[y.raw==1,]

tmp0 <- sample(1:n0.raw,size=n0,replace=FALSE)
tmp1 <- sample(1:n1 ,size=n1,replace=FALSE)

X <- rbind(X0[tmp0,],X1)
y <- c(rep(1,n0),rep(2,n1))

str(X)
str(y)

PV <- cvpvs.logreg(X,y,
tau.o=5,pen.method="v",progress=TRUE)
analyze.pvs(y=y,pv=PV,pvplot=FALSE)

## End(Not run)
```

cvpvs.wnn

Cross-Validated P-Values (Weighted Nearest Neighbors)

Description

Computes cross-validated nonparametric p-values for the potential class memberships of the training data. The p-values are based on 'weighted nearest-neighbors'.

Usage

cvpvs.wnn(X, Y, wtype = c('linear', 'exponential'), W = NULL,
tau = 0.3, distance = c('euclidean', 'ddeuclidean', 'mahalanobis'), cova = c('standard', 'M', 'sym'))

Arguments

x
matrix containing training observations, where each observation is a row vector.
y
vector indicating the classes which the training observations belong to.

wtype
type of the weight function (see section 'Details' below).

w
vector of the (decreasing) weights (see section 'Details' below).

tau
parameter of the weight function. If tau is a vector or tau = NULL, the program searches for the best tau. For more information see section 'Details'.

distance
the distance measure:

"euclidean": fixed Euclidean distance,
"ddeuclidean": data driven Euclidean distance (component-wise standardization),
"mahalanobis": Mahalanobis distance.

cova
estimator for the covariance matrix:

'standard': standard estimator,
'M': M-estimator,
'sym': symmetrized M-estimator.

Details

Computes cross-validated nonparametric p-values for the potential class memberships of the training data. Precisely, for each feature vector X[i,] and each class b the number PV[i, b] is a p-value for the null hypothesis that Y[i] equals b. This p-value is based on a permutation test applied to an estimated Bayesian likelihood ratio, using 'weighted nearest neighbors' with estimated prior probabilities N(b)/n. Here N(b) is the number of observations of class b and n is the total number of observations.

The (decreasing) weights for the observations can be either indicated with a n dimensional vector w or (if w = NULL) one of the following weight functions can be used:

linear:

\[ W_i = \max\left(1 - \frac{i}{n}/\tau, 0\right), \]

exponential:

\[ W_i = \left(1 - \frac{i}{n}\right)^\tau. \]

If tau is a vector, the program searches for the best tau. To determine the best tau for the p-value PV[i, b], the class label of the training observation X[i,] is set temporarily to b and then for all training observations with Y[j] != b the sum of the weights of the observations belonging to class b is computed. Then the tau which minimizes the sum of these values is chosen.

If w = NULL and tau = NULL, tau is set to seq(0.1, 0.9, 0.1) if wtype = "l" and to c(1, 5, 10, 20) if wtype = "e".
Value

PV is a matrix containing the cross-validated p-values. Precisely, for each feature vector $X[i,]$ and each class $b$ the number $PV[i, b]$ is a p-value for the null hypothesis that $Y[i] = b$.

If tau is a vector or NULL (and W = NULL), PV has an attribute "opt.tau", which is a matrix and opt.tau[i,b] is the best tau for observation $X[i,]$ and class b (see section 'Details'). "opt.tau" is used to compute the p-values.

Author(s)

Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
Lutz Dümbgen <lutz.duembgen@stat.unibe.ch>
www.imsv.unibe.ch/duembgen/index_ger.html

References


See Also

cvpvs, cvpvs.gaussian, cvpvs.knn, cvpvs.logreg

Examples

```r
X <- iris[, 1:4]
Y <- iris[, 5]
cvpvs.wnn(X, Y, wtype = 'l', tau = 0.5)
```

Description

Computes nonparametric p-values for the potential class memberships of new observations.

Usage

```r
pvs(NewX, X, Y, method = c('gaussian', 'knn', 'wnn', 'logreg'), ...)
```
Arguments

NewX  
data matrix consisting of one or several new observations (row vectors) to be classified.

X  
matrix containing training observations, where each observation is a row vector.

Y  
vector indicating the classes which the training observations belong to.

method  
one of the following methods:
'gaussian': plug-in statistic for the standard Gaussian model,
'knn': k nearest neighbors,
'wnn': weighted nearest neighbors,
'logreg': multicategory logistic regression with $l_1$-penalization.

...  
further arguments depending on the method (see pvs.gaussian, pvs.knn, pvs.wnn, pvs.logreg).

Details

Computes nonparametric p-values for the potential class memberships of new observations. Precisely, for each new observation $\text{NewX}[i,]$ and each class $b$ the number $\text{PV}[i,b]$ is a p-value for the null hypothesis that $Y[i] = b$.

This p-value is based on a permutation test applied to an estimated Bayesian likelihood ratio, using a plug-in statistic for the Gaussian model, 'k nearest neighbors', 'weighted nearest neighbors' or multicategory logistic regression with $l_1$-penalization (see pvs.gaussian, pvs.knn, pvs.wnn, pvs.logreg) with estimated prior probabilities $N(b)/n$. Here $N(b)$ is the number of observations of class $b$ and $n$ is the total number of observations.

Value

$\text{PV}$ is a matrix containing the p-values. Precisely, for each new observation $\text{NewX}[i,]$ and each class $b$ the number $\text{PV}[i,b]$ is a p-value for the null hypothesis that $Y[i] = b$.

Author(s)

Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
Lutz Dümbgen <lutz.duembgen@stat.unibe.ch>
www.imsv.unibe.ch/duembgen/index_ger.html

References


See Also

pvs.gaussian, pvs.knn, pvs.wnn, pvs.logreg, cvpvs, analyze.pvs
Examples

```r
NewX <- iris[c(50, 100, 150), 1:4]
pvs(NewX, X, Y, method = 'k', k = 10)
```

## Description

Computes nonparametric p-values for the potential class memberships of new observations. The p-values are based on a plug-in statistic for the standard Gaussian model. The latter means that the conditional distribution of $X$, given $Y = y$, is Gaussian with mean depending on $y$ and a global covariance matrix.

## Usage

```r
pvs.gaussian(NewX, X, Y, cova = c('standard', 'M', 'sym'))
```

## Arguments

- `NewX`: data matrix consisting of one or several new observations (row vectors) to be classified.
- `X`: matrix containing training observations, where each observation is a row vector.
- `Y`: vector indicating the classes which the training observations belong to.
- `cova`: estimator for the covariance matrix:
  - 'standard': standard estimator,
  - 'M': M-estimator,
  - 'sym': symmetrized M-estimator.

## Details

Computes nonparametric p-values for the potential class memberships of new observations. Precisely, for each new observation `NewX[i,]` and each class $b$, the number `PV[i,b]` is a p-value for the null hypothesis that $Y[i] = b$.

This p-value is based on a permutation test applied to an estimated Bayesian likelihood ratio, using a plug-in statistic for the standard Gaussian model with estimated prior probabilities $N(b)/n$. Here $N(b)$ is the number of observations of class $b$ and $n$ is the total number of observations.

## Value

`PV` is a matrix containing the p-values. Precisely, for each new observation `NewX[i,]` and each class $b$, the number `PV[i,b]` is a p-value for the null hypothesis that $Y[i] = b$. 
Author(s)

Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
Lutz Dümbgen <lutz.duembgen@stat.unibe.ch>
www.imsv.unibe.ch/duembgen/index_ger.html

References


See Also

pvs, pvs.knn, pvs.wnn, pvs.logreg

Examples

```
y <- iris[,c(1:49, 51:99, 101:149), 5]
NewX <- iris[,c(50, 100, 150), 1:4]

pvs.gaussian(NewX, X, Y, cova = 'standard')
```

---

### Description

Computes nonparametric p-values for the potential class memberships of new observations. The p-values are based on "k nearest neighbors".

### Usage

```
pvs.knn(NewX, X, Y, k = NULL, distance = c('euclidean', 'd deutschean', 'mahalanobis'), cova = c('standard', 'M', 'sym'))
```

### Arguments

- **NewX**
  - data matrix consisting of one or several new observations (row vectors) to be classified.
- **X**
  - matrix containing training observations, where each observation is a row vector.
- **Y**
  - vector indicating the classes which the training observations belong to.
k number of nearest neighbors. If k is a vector or k = NULL, the program searches for the best k. For more information see section 'Details'.

distance the distance measure:
'euclidean': fixed Euclidean distance,
'ddeuclidean': data driven Euclidean distance (component-wise standardization),
'mahalanobis': Mahalanobis distance.

cova estimator for the covariance matrix:
'standard': standard estimator,
'M': M-estimator,
'sym': symmetrized M-estimator.

Details
Computes nonparametric p-values for the potential class memberships of new observations. Precisely, for each new observation \( \text{NewX}[i,] \) and each class \( b \) the number \( PV[i, b] \) is a p-value for the null hypothesis that \( Y[i] = b \).

This p-value is based on a permutation test applied to an estimated Bayesian likelihood ratio, using 'k nearest neighbors' with estimated prior probabilities \( N(b)/n \). Here \( N(b) \) is the number of observations of class \( b \) and \( n \) is the total number of observations.

If \( k \) is a vector, the program searches for the best \( k \). To determine the best \( k \) for the p-value \( PV[i, b] \), the new observation \( \text{NewX}[i,] \) is added to the training data with class label \( b \) and then for all training observations with \( Y[j] \neq b \) the proportion of the \( k \) nearest neighbors of \( X[j,] \) belonging to class \( b \) is computed. Then the \( k \) which minimizes the sum of these values is chosen.

If \( k = \text{NULL} \), it is set to \( 2: \ceiling(\text{length}(Y)/2) \).

Value
\( PV \) is a matrix containing the p-values. Precisely, for each new observation \( \text{NewX}[i,] \) and each class \( b \) the number \( PV[i, b] \) is a p-value for the null hypothesis that \( Y[i] = b \).

If \( k \) is a vector or \( \text{NULL} \), \( PV \) has an attribute "opt.k", which is a matrix and \( \text{opt.k}[i, b] \) is the best \( k \) for observation \( \text{NewX}[i,] \) and class \( b \) (see section 'Details'). \( \text{opt.k}[i, b] \) is used to compute the p-value for observation \( \text{NewX}[i,] \) and class \( b \).

Author(s)
Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
Lutz Dümbgen <lutz.duebgen@stat.unibe.ch>
www.imsv.unibe.ch/duembgen/index_ger.html

References


pvs.logreg

See Also

pvs, pvs.gaussian, pvs.wnn, pvs.logreg

Examples

```r
NewX <- iris[c(50, 100, 150), 1:4]

pvs.knn(NewX, X, Y, k = c(5, 10, 15))
```

---

pvs.logreg

P-Values to Classify New Observations (Penalized Multicategory Logistic Regression)

Description

Computes nonparametric p-values for the potential class memberships of new observations. The p-values are based on 'penalized logistic regression'.

Usage

```r
pvs.logreg(NewX, X, Y, tau.o = 10, find.tau=FALSE, delta=2, tau.max=80, tau.min=1,
a0 = NULL, b0 = NULL,
pen.method = c('vectors', 'simple', 'none'),
progress = FALSE)
```

Arguments

- `NewX`: data matrix consisting of one or several new observations (row vectors) to be classified.
- `X`: matrix containing training observations, where each observation is a row vector.
- `Y`: vector indicating the classes which the training observations belong to.
- `tau.o`: the penalty parameter (see section 'Details' below).
- `find.tau`: logical. If TRUE the program searches for the best tau. For more information see section 'Details'.
- `delta`: factor for the penalty parameter. Should be greater than 1. Only needed if find.tau == TRUE.
- `tau.max`: maximal penalty parameter considered. Only needed if find.tau == TRUE.
- `tau.min`: minimal penalty parameter considered. Only needed if find.tau == TRUE.
- `a0, b0`: optional starting values for logistic regression.
- `pen.method`: the method of penalization (see section 'Details' below).
- `progress`: optional parameter for reporting the status of the computations.
Details

Computes nonparametric p-values for the potential class memberships of new observations. Precisely, for each new observation NewX[i,] and each class b the number PV[i,b] is a p-value for the null hypothesis that Y[i] equals b.

This p-value is based on a permutation test applied to an estimated Bayesian likelihood ratio, using 'penalized logistic regression'. This means, the conditional probability of Y = y, given X = x, is assumed to be proportional to \(exp(a_y + b_y^T x)\). The parameters a_y, b_y are estimated via penalized maximum log-likelihood. The penalization is either a weighted sum of the euclidean norms of the vectors (b_1[j], b_2[j],..., b_L[j]) (pen.method=='vectors') or a weighted sum of all moduli \(|b_\theta[j]|\) (pen.method=='simple'). The weights are given by tau.o times the sample standard deviation (within groups) of the j-th components of the feature vectors. In case of pen.method='none', no penalization is used, but this option may be unstable.

If find.tau == TRUE, the program searches for the best penalty parameter. To determine the best parameter tau for the p-value PV[i,b], the new observation NewX[i,] is added to the training data with class label b and then for all training observations with Y[j] != b the estimated probability of X[j,] belonging to class b is computed. Then the tau which minimizes the sum of these values is chosen. First, tau.o is compared with tau.o*delta. If tau.o*delta is better, it is compared with tau.o*delta^2, etc. The maximal parameter considered is tau.max. If tau.o is better than tau.o*delta, it is compared with tau.o*delta^-1, etc. The minimal parameter considered is tau.min.

Value

PV is a matrix containing the p-values. Precisely, for each new observation NewX[i,] and each class b the number PV[i,b] is a p-value for the null hypothesis that Y[i] = b. If find.tau == TRUE, PV has an attribute "tau.opt", which is a matrix and tau.opt[i,b] is the best tau for observation NewX[i,] and class b (see section 'Details'). tau.opt[i,b] is used to compute the p-value for observation NewX[i,] and class b.

Author(s)

Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
Lutz Dümbgen <lutz.duembgen@stat.unibe.ch>
www.imsv.unibe.ch/duembgen/index_ger.html

References


See Also

pvs, pvs.gaussian, pvs.knn, pvs.wnn
**Examples**

```r
NewX <- iris[c(50, 100, 150), 1:4]

pvs.logreg(NewX, X, Y, tau.o=1, pen.method="vectors", progress=TRUE)

# A bigger data example: Buerk's hospital data.
## Not run:
data(buerk)
x.raw <- as.matrix(buerk[,1:21])
y.raw <- buerk[,22]
n0.raw <- sum(1 - y.raw)
n1 <- sum(y.raw)
n0 <- 3*n1

X0 <- x.raw[y.raw==0,]
x1 <- x.raw[y.raw==1,]

tmpi0 <- sample(1:n0.raw, size=3*n1, replace=FALSE)
tmpi1 <- sample(1:n1, size= n1, replace=FALSE)

Xtrain <- rbind(X0[tmpi0[1:(n0-100)],],X1[1:(n1-100),])
Ytrain <- c(rep(1,n0-100),rep(2,n1-100))
Xtest <- rbind(X0[tmpi0[(n0-99):n0],],X1[(n1-99):n1,])
Ytest <- c(rep(1,100),rep(2,100))

PV <- pvs.logreg(Xtest,Xtrain,Ytrain,tau.o=2,progress=TRUE)
analyze.pvs(Y=Ytest,pv=PV,pvplot=FALSE)

## End(Not run)
```

---

**pvs.wnn**  
P-Values to Classify New Observations (Weighted Nearest Neighbors)

### Description
Computes nonparametric p-values for the potential class memberships of new observations. The p-values are based on 'weighted nearest-neighbors'.

### Usage
```
pvs.wnn(NewX, X, Y, wtype = c('linear', 'exponential'), W = NULL, tau = 0.3, distance = c('euclidean', 'dortclustvid', 'mahalanobis'), cova = c('standard', 'M', 'sym'))
```
Arguments

- **NewX**: data matrix consisting of one or several new observations (row vectors) to be classified.
- **X**: matrix containing training observations, where each observation is a row vector.
- **Y**: vector indicating the classes which the training observations belong to.
- **wtype**: type of the weight function (see section 'Details' below).
- **w**: vector of the (decreasing) weights (see section 'Details' below).
- **tau**: parameter of the weight function. If `tau` is a vector or `tau = NULL`, the program searches for the best `tau`. For more information see section 'Details'.
- **distance**: the distance measure:
  - 'euclidean': fixed Euclidean distance,
  - 'ddeuclidean': data driven Euclidean distance (component-wise standardization),
  - 'mahalanobis': Mahalanobis distance.
- **cova**: estimator for the covariance matrix:
  - 'standard': standard estimator,
  - 'M': M-estimator,
  - 'sym': symmetrized M-estimator.

Details

Computes nonparametric p-values for the potential class memberships of new observations. Precisely, for each new observation `NewX[i,]` and each class `b` the number `PV[i,b]` is a p-value for the null hypothesis that `Y[i] = b`.

This p-value is based on a permutation test applied to an estimated Bayesian likelihood ratio, using 'weighted nearest neighbors' with estimated prior probabilities `N(b)/n`. Here `N(b)` is the number of observations of class `b` and `n` is the total number of observations.

The (decreasing) weights for the observation can be either indicated with a `n` dimensional vector `w` or (if `w = NULL`) one of the following weight functions can be used:

- **linear:**
  \[ W_i = \max\left(1 - \frac{i}{n}/\tau, 0\right) \]
- **exponential:**
  \[ W_i = (1 - \frac{i}{n})^{\tau} \]

If `tau` is a vector, the program searches for the best `tau`. To determine the best `tau` for the p-value `PV[i,b]`, the new observation `NewX[i,]` is added to the training data with class label `b` and then for all training observations with `Y[j] != b` the sum of the weights of the observations belonging to class `b` is computed. Then the `tau` which minimizes the sum of these values is chosen.

If `tau = NULL`, it is set to `seq(0.1, 0.9, 0.1)` if `wtype = "l"` and to `c(1, 5, 10, 20)` if `wtype = "e"`.

Value

- **PV**: a matrix containing the p-values. Precisely, for each new observation `NewX[i,]` and each class `b` the number `PV[i,b]` is a p-value for the null hypothesis that `Y[i] = b`.

If `tau` is a vector or `NULL` (and `w = NULL`), `PV` has an attribute "opt.tau", which is a matrix and `opt.tau[i,b]` is the best `tau` for observation `NewX[i,]` and class `b` (see section 'Details').

`opt.tau[i,b]` is used to compute the p-value for observation `NewX[i,]` and class `b`. 
Author(s)

Niki Zumbrunnen <niki.zumbrunnen@gmail.com>
Lutz Dümbgen <lutz.duembgen@stat.unibe.ch>
www.imsv.unibe.ch/duembgen/index_ger.html

References


See Also

`pvs, pvs.gaussian, pvs.knn, pvs.logreg`

Examples

```r
NewX <- iris[c(50, 100, 150), 1:4]

pvs.wnn(NewX, X, Y, wtype = 'l', tau = 0.5)
```
Index

*Topic **classif**
  analyze.pvs, 3
  cvpvs, 6
  cvpvs.gaussian, 7
  cvpvs.knn, 8
  cvpvs.logreg, 10
  cvpvs.wnn, 12
  pvclass-package, 2
  pvs, 14
  pvs.gaussian, 16
  pvs.knn, 17
  pvs.logreg, 19
  pvs.wnn, 21
*Topic **datasets**
  buerk, 4
*Topic **package**
  pvclass-package, 2

analyze.pvs, 2, 3, 7, 15

buerk, 4

cvpvs, 2–4, 6, 8, 10, 11, 14, 15
cvpvs.gaussian, 6, 7, 7, 10, 11, 14
cvpvs.knn, 6–8, 8, 11, 14
cvpvs.logreg, 6–8, 10, 10, 14
cvpvs.wnn, 6–8, 10, 11, 12

pvclass (pvclass-package), 2
pvclass-package, 2
pvs, 2–4, 7, 14, 17, 19, 20, 23
pvs.gaussian, 15, 16, 19, 20, 23
pvs.knn, 15, 17, 17, 20, 23
pvs.logreg, 15, 17, 19, 19, 23
pvs.wnn, 15, 17, 19, 20, 21