Package ‘mcca’

September 14, 2018

Type Package
Title Multi-Category Classification Accuracy
Version 0.4.0
Author Ming Gao, Jialiang Li
Maintainer Ming Gao <gaoming96@sjtu.edu.cn>
License GPL
Encoding UTF-8
LazyData true
Imports nnet,rpart,e1071,MASS,stats,pROC,caret
NeedsCompilation no
Repository CRAN
Date/Publication 2018-09-14 04:50:02 UTC

R topics documented:

mcca-package ................................................................. 2
ccp ................................................................. 4
estp ................................................................. 6
ests ................................................................. 8
hum ................................................................. 11
idi ................................................................. 13
nri ................................................................. 15
pdi ................................................................. 17
Description

Six common multi-category classification accuracy evaluation measures are included i.e., Correct Classification Percentage (CCP), Hypervolume Under Manifold (HUM), Integrated Discrimination Improvement (IDI), Net Reclassification Improvement (NRI), Polytomous Discrimination Index (PDI) and R-squared (RSQ). It allows users to fit many popular classification procedures, such as multinomial logistic regression, support vector machine, classification tree, and user computed risk values.

Details

Package: mcca
Type: Package
Version: 0.3
Date: 2018-03-28
License: GPL

Functions

- `ccp`: Calculate CCP Value
- `hum`: Calculate HUM Value
- `idi`: Calculate IDI Value
- `nri`: Calculate NRI Value
- `pdi`: Calculate PDI Value
- `rsq`: Calculate RSQ Value
- `pm`: Calculate Probability Matrix
- `ests`: Estimated Information for Single Model Evaluation Value
- `estp`: Estimated Information for Paired Model Evaluation Value

Installing and using

To install this package, make sure you are connected to the internet and issue the following command in the R prompt:
install.packages("mcca")

To load the package in R:

library(mcca)

Author(s)

Ming Gao, Jialiang Li

Maintainer: Ming Gao <gaoming96@sjtu.edu.cn>

References


Van Calster B, Vergouwe Y, Looman CWN, Van Belle V, Timmerman D and Steyerberg EW. As-
sessing the discriminative ability of risk models for more than two outcome categories. European


See Also

CRAN packages HUM for HUM.

CRAN packages nnet, rpart, e1071, MASS employed in this package.

Examples

rm(list=ls())
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]
ccp(y = label, d = data, method = "multinom", k = 3,maxit = 1000,MaxNWts = 2000,trace=FALSE)
## [1] 0.9866667
ccp(y = label, d = data, method = "multinom", k = 3)
## [1] 0.9866667
ccp(y = label, d = data, method = "svm", k = 3)
## [1] 0.9733333
ccp(y = label, d = data, method = "svm", k = 3,kernel="sigmoid",cost=4, scale=TRUE,coef=0.5)
## [1] 0.8333333
### Describe CCP Value

Compute the Correct Classification Percentage (CCP) value of two or three or four categories classifiers with an option to define the specific model or user-defined model.

**usage**

```r
ccp(y, d, method = "multinom", k=3, ...)```

**Arguments**

- `y` The multinomial response vector with two, three or four categories. It can be factor or integer-valued.
- `d` The set of candidate markers, including one or more columns. Can be a data frame or a matrix; if the method is "label", then d should be the label vector.
- `method` Specifies what method is used to construct the classifier based on the marker set in d. Available option includes the following methods: "multinom": Multinomial Logistic Regression which is the default method, requiring R package nnet; "tree": Classification Tree method, requiring R package rpart; "svm": Support Vector Machine (C-classification and radial basis as default), requiring R package e1071; "lda": Linear Discriminant Analysis, requiring R package lda;
"label"; d is a label vector resulted from any external classification algorithm obtained by the user, should be encoded from 1; "prob"; d is a probability matrix resulted from any external classification algorithm obtained by the user.

k  Number of the categories, can be 2 or 3 or 4.

Additional arguments in the chosen method’s function.

Details

The function returns the CCP value for predictive markers based on a user-chosen machine learning method. Currently available methods include logistic regression (default), tree, lda, svm and user-computed risk values. This function is general since we can evaluate the accuracy for marker combinations resulted from complicated classification algorithms.

Value

The CCP value of the classification using a particular learning method on a set of marker(s).

Note

Users are advised to change the operating settings of various classifiers since it is well known that machine learning methods require extensive tuning. Currently only some common and intuitive options are set as default and they are by no means the optimal parameterization for a particular data analysis. Users can put machine learning methods’ parameters after tuning. A more flexible evaluation is to consider “method=label” in which case the input d should be a label vector.

Author(s)

Ming Gao: gaom96@sjtu.edu.cn
Jialiang Li: stalj@nus.edu.sg

References


See Also

pdi

Examples

```r
rm(list=ls())
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]
ccp(y = label, d = data, method = "multinom", k = 3,maxit = 1000,MaxNWts = 2000,trace=FALSE)
## [1] 0.9866667
ccp(y = label, d = data, method = "multinom", k = 3)
```
## estp

*Inference for Accuracy Improvement Measures based on Bootstrap*

### Description

compute the bootstrap standard error and confidence interval for the classification accuracy improvement for a pair of nested models.

### Usage

```r
estp(y, m1, m2, acc="idi", level=0.95, method="multinom", k=3, B=250, balance=FALSE, ...)
```

### Arguments

- **y** The multinomial response vector with two, three or four categories. It can be factor or integer-valued.
- **m1** The set of marker(s) included in the baseline model, can be a data frame or a matrix; if the method is "prob", then m1 should be the prediction probability matrix of the baseline model.
- **m2** The set of additional marker(s) included in the improved model, can be a data frame or a matrix; if the method is "prob", then m2 should be the prediction probability matrix of the improved model.
- **acc** Accuracy measure to be evaluated. Allow two choices: "idi", "nri".
level The confidence level. Default value is 0.95.
method Specifies what method is used to construct the classifier based on the marker set in m1 & m2. Available option includes the following methods: "multinom": Multinomial Logistic Regression which is the default method, requiring R package nnet; "tree": Classification Tree method, requiring R package rpart; "svm": Support Vector Machine (C-classification and radial basis as default), requiring R package e1071; "lda": Linear Discriminant Analysis, requiring R package lda; "prob": m1 & m2 are risk matrices resulted from any external classification algorithm obtained by the user.
k Number of the categories, can be 2, 3 or 4.
b Number of bootstrap resamples.
balance Logical, if TRUE, the class prevalence of the bootstrap sample is forced to be identical to the class prevalence of the original sample. Otherwise the prevalence of the bootstrap sample may be random.
... Additional arguments in the chosen method's function.

Details
The function returns the standard error and confidence interval for a paired model evaluation method. All the other arguments are the same as the function hum.

Value
value The specific value of the classification using a particular learning method on a set of marker(s).
se The standard error of the value.
interval The confidence interval of the value.

Note
Users are advised to change the operating settings of various classifiers since it is well known that machine learning methods require extensive tuning. Currently only some common and intuitive options are set as default and they are by no means the optimal parameterization for a particular data analysis. Users can put machine learning methods' parameters after tuning. A more flexible evaluation is to consider "method=prob" in which case the input m1 & m2 should be a matrix of membership probabilities with k columns and each row of m1 & m2 should sum to one.

Author(s)
Ming Gao: gaoming96@sjtu.edu.cn
Jialiang Li: stalj@nus.edu.sg

See Also
ests
Examples

```r
ests <- 1

table(mtcars$carb)
for (i in 1:length(mtcars$carb)) {
  if (mtcars$carb[i] == 3 | mtcars$carb[i] == 6 | mtcars$carb[i] == 8) {
    mtcars$carb[i] <- 9
  }
}
data <- data.matrix(mtcars[, c(1, 5)])
mtcars$carb <- factor(mtcars$carb, labels = c(1, 2, 3, 4))
label <- as.numeric(mtcars$carb)
str(mtcars)
estp(y = label, m1 = data[, 1], m2 = data[, 2], acc="idi", method="lda", k=4,B=10)
```

```
## $value
## [1] 0.1235644

## $se
## [1] 0.07053541

## $interval
## [1] 0.05298885 0.21915088

estp(y = label, m1 = data[, 1], m2 = data[, 2], acc="nri", method="tree", k=4,B=5)
```

```
## $value
## [1] 0.05

## $se
## [1] 0.09249111

## $interval
## [1] 0.0000000 0.1458333
```

---

**ests**  
Inference for Accuracy Measures based on Bootstrap

**Description**

compute the bootstrap standard error and confidence interval for the classification accuracy for a single classification model.

**Usage**

```r
ests(y, d, acc="hum", level=0.95, method="multinom", k=3, B=250, balance=FALSE, ...)
```
Arguments

- **y**: The multinomial response vector with two, three or four categories. It can be factor or integer-valued.
- **d**: The set of candidate markers, including one or more columns. Can be a data frame or a matrix; if the method is "prob", then d should be the probability matrix.
- **acc**: Accuracy measure to be evaluated. Allow four choices: "hum", "pdi", "ccp" and "rsq".
- **level**: The confidence level. Default value is 0.95.
- **method**: Specifies what method is used to construct the classifier based on the marker set in d. Available option includes the following methods: "multinom": Multinomial Logistic Regression which is the default method, requiring R package nnet; "tree": Classification Tree method, requiring R package rpart; "svm": Support Vector Machine (C-classification and radial basis as default), requiring R package e1071; "lda": Linear Discriminant Analysis, requiring R package lda; "label": d is a label vector resulted from any external classification algorithm obtained by the user, should be encoded from 1; "prob": d is a probability matrix resulted from any external classification algorithm obtained by the user.
- **k**: Number of the categories, can be 2, 3 or 4.
- **B**: Number of bootstrap resamples.
- **balance**: Logical, if TRUE, the class prevalence of the bootstrap sample is forced to be identical to the class prevalence of the original sample. Otherwise the prevalence of the bootstrap sample may be random.
- **...**: Additional arguments in the chosen method’s function.

Details

The function returns the standard error and confidence interval for a single model evaluation method. All the other arguments are the same as the function hum.

Value

- **value**: The specific value of the classification using a particular learning method on a set of marker(s).
- **se**: The standard error of the value.
- **interval**: The confidence interval of the value.

Note

Users are advised to change the operating settings of various classifiers since it is well known that machine learning methods require extensive tuning. Currently only some common and intuitive options are set as default and they are by no means the optimal parameterization for a particular data analysis. Users can put machine learning methods’ parameters after tuning. A more flexible evaluation is to consider "method=prob" in which case the input d should be a matrix of membership probabilities with k columns and each row of d should sum to one.
Author(s)
Ming Gao: gaoming96@sjtu.edu.cn
Jialiang Li: stalj@nus.edu.sg

See Also
estp

Examples
rm(list=ls())
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]
est(y = label, d = data, acc="hum", level=0.95, method = "multinom", k = 3, B=10, trace=FALSE)
## $value
## [1] 0.9972
## $se
## [1] 0.002051529
## $interval
## [1] 0.9935662 1.0000000
est(y = label, d = data, acc="pdi", level=0.85, method = "tree", k = 3, B=10)
## $value
## [1] 0.9213333
## $se
## [1] 0.02148812
## $interval
## [1] 0.9019608 0.9629630
rm(list=ls())
table(mtcars$carb)
for (i in 1:length(mtcars$carb)) {
  if (mtcars$carb[i] == 3 | mtcars$carb[i] == 6 | mtcars$carb[i] == 8) { mtcars$carb[i] <- 9
  }
}
data <- data.matrix(mtcars[, c(1:2)])
mtcars$carb <- factor(mtcars$carb, labels = c(1, 2, 3, 4))
label <- as.numeric(mtcars$carb)
str(mtcars)
est(y = label, d = data, acc="hum", level=0.95, method = "multinom", k = 4, trace=FALSE, B=5)
## $value
## [1] 0.2822857
Calculate HUM Value

Description

compute the Hypervolume Under Manifold (HUM) value of two or three or four categories classifiers with an option to define the specific model or user-defined model.

Usage

hum(y, d, method="multinom", k=3, ...)

Arguments

y
The multinomial response vector with two, three or four categories. It can be factor or integer-valued.

d
The set of candidate markers, including one or more columns. Can be a data frame or a matrix; if the method is "prob", then d should be the probability matrix.

method
Specifies what method is used to construct the classifier based on the marker set in d. Available option includes the following methods: "multinom": Multinomial Logistic Regression which is the default method, requiring R package nnet; "tree": Classification Tree method, requiring R package rpart; "svm": Support Vector Machine (C-classification and radial basis as default), requiring R package e1071; "lda": Linear Discriminant Analysis, requiring R package lda; "prob": d is a risk matrix resulted from any external classification algorithm obtained by the user.

k
Number of the categories, can be 2 or 3 or 4.

... Additional arguments in the chosen method’s function.

Details

The function returns the HUM value for predictive markers based on a user-chosen machine learning method. Currently available methods include logistic regression (default), tree, lda, svm and user-computed risk values. For binary outcome, one can use AUC value (HUM reduces to AUC in such case). This function is more general than the package HUM, since we can evaluate the accuracy for marker combinations resulted from complicated classification algorithms.

Value

The HUM value of the classification using a particular learning method on a set of marker(s).
Note

Users are advised to change the operating settings of various classifiers since it is well known that machine learning methods require extensive tuning. Currently only some common and intuitive options are set as default and they are by no means the optimal parameterization for a particular data analysis. Users can put machine learning methods’ parameters after tuning. A more flexible evaluation is to consider "method=prob" in which case the input d should be a matrix of membership probabilities with k columns and each row of d should sum to one.

Author(s)

Ming Gao: gaoming96@sjtu.edu.cn
Jialiang Li: stalj@nus.edu.sg

References


See Also

pdi

Examples

```r
rm(list=ls())
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]
hum(y = label, d = data, method = "multinom", k = 3)
## [1] 0.9972
hum(y = label, d = data, method = "svm", k = 3)
## [1] 0.9964
hum(y = label, d = data, method = "svm", k = 3, type="C", kernel="linear", cost=4, scale=TRUE)
## [1] 0.9972
hum(y = label, d = data, method = "tree", k = 3)
## [1] 0.998

data <- data.matrix(iris[, 1:4])
label <- as.numeric(iris[, 5])
# multinomial
require(nnet)
# model
fit <- multinom(label ~ data, maxit = 1000, MaxNWts = 2000)
predict.probs <- predict(fit, type = "probs")
pp<- data.frame(predict.probs)
# extract the probability assessment vector
head(pp)
hum(y = label, d = pp, method = "prob", k = 3)
```
Calculate IDI Value

### Description

compute the integrated discrimination improvement (IDI) value of two or three or four categories classifiers with an option to define the specific model or user-defined model.

### Usage

```R
idi(y, m1, m2, method="multinom", k=3, ...)
```

### Arguments

- **y**: The multinomial response vector with two, three or four categories. It can be factor or integer-valued.
- **m1**: The set of marker(s) included in the baseline model, can be a data frame or a matrix; if the method is "prob", then m1 should be the prediction probability matrix of the baseline model.
- **m2**: The set of additional marker(s) included in the improved model, can be a data frame or a matrix; if the method is "prob", then m2 should be the prediction probability matrix of the improved model.
- **method**: Specifies what method is used to construct the classifier based on the marker set in m1 & m2. Available option includes the following methods:"multinom": Multinomial Logistic Regression which is the default method, requiring R package nnet;"tree": Classification Tree method, requiring R package rpart;"svm": Support Vector Machine (C-classification and radial basis as default), requiring R package e1071;"lda": Linear Discriminant Analysis, requiring R package e1071.
ldai="prob": m1 & m2 are risk matrices resulted from any external classification algorithm obtained by the user.

k

Number of the categories, can be 2 or 3 or 4.

Additional arguments in the chosen method’s function.

Details

The function returns the IDI value for predictive markers based on a user-chosen machine learning method. Currently available methods include logistic regression (default), tree, lda, svm and user-computed risk values. This function is general since we can evaluate the accuracy for marker combinations resulted from complicated classification algorithms.

Value

The IDI value of the classification using a particular learning method on a set of marker(s).

Note

Users are advised to change the operating settings of various classifiers since it is well known that machine learning methods require extensive tuning. Currently only some common and intuitive options are set as default and they are by no means the optimal parameterization for a particular data analysis. Users can put machine learning methods’ parameters after tuning. A more flexible evaluation is to consider "method=prob" in which case the input m1 & m2 should be a matrix of membership probabilities with k columns and each row of m1 & m2 should sum to one.

Author(s)

Ming Gao: gaoming96@sjtu.edu.cn
Jialiang Li: stalj@nus.edu.sg

References


See Also

nri

Examples

```r
rm(list=ls())
table(mtcars$carb)
for (i in 1:length(mtcars$carb)) {
  if (mtcars$carb[i] == 3 | mtcars$carb[i] == 6 | mtcars$carb[i] == 8) {
    mtcars$carb[i] <- 9
  }
}
```
nri

Calculate NRI Value

Description

compute the net reclassification improvement (NRI) value of two or three or four categories classifiers with an option to define the specific model or user-defined model.

Usage

nri(y, m1, m2, method="multinom", k=3, ...)

Arguments

y
  The multinomial response vector with two, three or four categories. It can be factor or integer-valued.

m1
  The set of marker(s) included in the baseline model, can be a data frame or a matrix; if the method is "prob", then m1 should be the prediction probability matrix of the baseline model.

m2
  The set of additional marker(s) included in the improved model, can be a data frame or a matrix; if the method is "prob", then m2 should be the prediction probability matrix of the improved model.

method
  Specifies what method is used to construct the classifier based on the marker set in m1 & m2. Available option includes the following methods: "multinom": Multinomial Logistic Regression which is the default method, requiring R package nnet;"tree": Classification Tree method, requiring R package rpart;"svm": Support Vector Machine (C-classification and radial basis as default), requiring R package e1071;"lda": Linear Discriminant Analysis, requiring R package lda;"label": m1 & m2 are label vectors resulted from any external classification algorithm obtained by the user;"prob": m1 & m2 are probability matrices resulted from any external classification algorithm obtained by the user.

k
  Number of the categories, can be 2 or 3 or 4.

... Additional arguments in the chosen method’s function.
Details

The function returns the NRI value for predictive markers based on a user-chosen machine learning method. Currently available methods include logistic regression (default), tree, lda, svm and user-computed risk values. This function is general since we can evaluate the accuracy for marker combinations resulted from complicated classification algorithms.

Value

The NRI value of the classification using a particular learning method on a set of marker(s).

Note

Users are advised to change the operating settings of various classifiers since it is well known that machine learning methods require extensive tuning. Currently only some common and intuitive options are set as default and they are by no means the optimal parameterization for a particular data analysis. Users can put machine learning methods’ parameters after tuning. A more flexible evaluation is to consider "method=prob" in which case the input m1 & m2 should be a matrix of membership probabilities with k columns and each row of m1 & m2 should sum to one.

Author(s)

Ming Gao: gaoming96@sjtu.edu.cn
Jialiang Li: stalj@nus.edu.sg

References


See Also

idi

Examples

```r
rm(list=ls())
table(mtcars$carb)
for (i in (1:length(mtcars$carb))) {
  if (mtcars$carb[i] \in [3 | mtcars$carb[i] <= 6 | mtcars$carb[i] == 8) {
    mtcars$carb[i] <- 9
  }
}
data <- data.matrix(mtcars[, c(1, 5)])
mtcars$carb <- factor(mtcars$carb, labels = c(1, 2, 3, 4))
label <- as.numeric(mtcars$carb)
str(mtcars)
nri(y = label, m1 = data[, 1], m2 = data[, 2], "lda", 4)
  # [1] 0.1
```
Calculate PDI Value

**Description**

compute the Polytomous Discrimination Index (PDI) value of two or three or four categories classifiers with an option to define the specific model or user-defined model.

**Usage**

```r
pdi(y, d, method="multinom", k=3, ...)
```

**Arguments**

- `y` The multinomial response vector with two, three or four categories. It can be factor or integer-valued.
- `d` The set of candidate markers, including one or more columns. Can be a data frame or a matrix; if the method is "prob", then `d` should be the probability matrix.
- `method` Specifies what method is used to construct the classifier based on the marker set in `d`. Available option includes the following methods: "multinom": Multinomial Logistic Regression which is the default method, requiring R package nnet; "tree": Classification Tree method, requiring R package rpart; "svm": Support Vector Machine (C-classification and radial basis as default), requiring R package e1071; "lda": Linear Discriminant Analysis, requiring R package lda; "prob": `d` is a risk matrix resulted from any external classification algorithm obtained by the user.
- `k` Number of the categories, can be 2 or 3 or 4.
- `...` Additional arguments in the chosen method’s function.

**Details**

The function returns the PDI value for predictive markers based on a user-chosen machine learning method. Currently available methods include logistic regression (default), tree, lda, svm and user-computed risk values. This function is general since we can evaluate the accuracy for marker combinations resulted from complicated classification algorithms.

**Value**

The PDI value of the classification using a particular learning method on a set of marker(s).
Note

Users are advised to change the operating settings of various classifiers since it is well known that machine learning methods require extensive tuning. Currently only some common and intuitive options are set as default and they are by no means the optimal parameterization for a particular data analysis. Users can put machine learning methods’ parameters after tuning. A more flexible evaluation is to consider "method=prob" in which case the input d should be a matrix of membership probabilities with k columns and each row of d should sum to one.

Author(s)

Ming Gao: gaoming96@sjtu.edu.cn
Jialiang Li: stalj@nus.edu.sg

References


See Also

hum

Examples

```
rm(list=ls())
str(iris)
data <- iris[, 3]
label <- iris[, 5]
pdi(y = label, d = data, method = "multinom", k = 3)
## [1] 0.9845333
pdi(y = label, d = data, method = "tree", k = 3)
## [1] 0.9882667
pdi(y = label, d = data, method = "tree", k = 3, control = rpart::rpart.control(minsplit = 200))
## [1] 0

data <- data.matrix(iris[, 3])
label <- as.numeric(iris[, 5])
# multinomial
require(nnet)
# model
fit <- multinom(label ~ data, maxit = 1000, MaxNWts = 2000)
predict.probs <- predict(fit, type = "probs")
pp <- data.frame(predict.probs)
# extract the probability assessment vector
head(pp)
pdi(y = label, d = pp, method = "prob", k = 3)
## [1] 0.9845333
```
Calculate Probability Matrix

Description

compute the probability matrix of two or three or four categories classifiers with an option to define the specific model or user-defined model.

Usage

```r
pm(y, d, method="multinom", k=3, ...)
```

Arguments

- **y**
  - The multinomial response vector with two, three or four categories. It can be factor or integer-valued.
- **d**
  - The set of candidate markers, including one or more columns. Can be a data frame or a matrix.
- **method**
  - Specifies what method is used to construct the classifier based on the marker set in d. Available option includes the following methods: "multinom": Multinomial Logistic Regression which is the default method, requiring R package nnet; "tree": Classification Tree method, requiring R package rpart; "svm": Support Vector Machine (C-classification and radial basis as default), requiring R package e1071; "lda": Linear Discriminant Analysis, requiring R package lda.
- **k**
  - Number of the categories, can be 2 or 3 or 4.
- **...**
  - Additional arguments in the chosen method’s function.

Details

The function returns the probability matrix for predictive markers based on a user-chosen machine learning method. Currently available methods include logistic regression (default), tree, lda, svm and user-computed risk values.

Value

The probability matrix of the classification using a particular learning method on a set of marker(s).

Author(s)

Ming Gao: gaoming96@sjtu.edu.cn
Jialiang Li: stalj@nus.edu.sg
References


See Also

di

Examples

```r
rm(list=ls())
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]
pm(y = label, d = data, method = "multinom", k = 3)
```

### rsq

**Calculate RSQ Value**

**Description**

compute the R-squared (RSQ) value of two or three or four categories classifiers with an option to define the specific model or user-defined model.

**Usage**

```r
rsq(y, d, method="multinom", k=3, ...)
```

**Arguments**

- **y** The multinomial response vector with two, three or four categories. It can be factor or integer-valued.
- **d** The set of candidate markers, including one or more columns. Can be a data frame or a matrix; if the method is "prob", then d should be the probability matrix.
- **method** Specifies what method is used to construct the classifier based on the marker set in d. Available option includes the following methods: "multinom": Multinomial Logistic Regression which is the default method, requiring R package nnet; "tree": Classification Tree method, requiring R package rpart; "svm": Support Vector Machine (C-classification and radial basis as default), requiring R package e1071; "lda": Linear Discriminant Analysis, requiring R package lda; "prob": d is a risk matrix resulted from any external classification algorithm obtained by the user.
- **k** Number of the categories, can be 2 or 3 or 4.
- **...** Additional arguments in the chosen method’s function.
Details

The function returns the RSQ value for predictive markers based on a user-chosen machine learning method. Currently available methods include logistic regression (default), tree, lda, svm and user-computed risk values. This function is general since we can evaluate the accuracy for marker combinations resulted from complicated classification algorithms.

Value

The RSQ value of the classification using a particular learning method on a set of marker(s).

Note

Users are advised to change the operating settings of various classifiers since it is well known that machine learning methods require extensive tuning. Currently only some common and intuitive options are set as default and they are by no means the optimal parameterization for a particular data analysis. Users can put machine learning methods’ parameters after tuning. A more flexible evaluation is to consider “method=prob” in which case the input d should be a matrix of membership probabilities with k columns and each row of d should sum to one.

Author(s)

Ming Gao: gaoming96@sjtu.edu.cn
Jialiang Li: stalj@nus.edu.sg

References


See Also

ccp

Examples

```r
rm(list=ls())
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]
rsq(y = label, d = data, method="multinom", k = 3)
## [1] 0.9638708
rsq(y = label, d = data, method = “tree”, k = 3)
## [1] 0.889694

data <- data.matrix(iris[, 1:4])
label <- as.numeric(iris[, 5])
# multinomial
require(nnet)
# model
```
fit <- multinom(label ~ data, maxit = 1000, MaxNWts = 2000)
predict.probs <- predict(fit, type = "probs")
pp <- data.frame(predict.probs)
# extract the probablity assessment vector
head(pp)
rsq(y = label, d = pp, method = "prob", k = 3)
## [1] 0.9638708

rm(list = ls())
table(mtcars$carb)
for (i in 1:length(mtcars$carb)) {
  if (mtcars$carb[i] == 3 | mtcars$carb[i] == 6 | mtcars$carb[i] == 8) {
    mtcars$carb[i] <- 9
  }
}
data <- data.matrix(mtcars[, c(1)])
mtcars$carb <- factor(mtcars$carb, labels = c(1, 2, 3, 4))
label <- as.numeric(mtcars$carb)
str(mtcars)
rsq(y = label, d = data, method = "tree", k = 4)
## [1] 0.1899336
rsq(y = label, d = data, method = "lda", k = 4)
## [1] 0.1456539
rsq(y = label, d = data, method = "lda", k = 4, prior = c(100, 1, 1, 1)/103)
## [1] 0.0431966
Index

*Topic CCP  
ccp, 4  
mcca-package, 2

*Topic CCR  
ests, 8

*Topic HUM  
hum, 11  
mcca-package, 2

*Topic IDI  
estp, 6  
idi, 13  
mcca-package, 2

*Topic NRI  
nri, 15

*Topic PDI  
pdi, 17

*Topic PM  
pm, 19

*Topic RSQ  
rsq, 20

ccp, 2, 4, 21

estp, 2, 6, 10

ests, 2, 7, 8

hum, 2, 7, 9, 11, 18

idi, 2, 13, 16

mcca (mcca-package), 2  
mcca-package, 2

nri, 2, 14, 15

pdi, 2, 5, 12, 17, 20

pm, 2, 19

rsq, 2, 20