Package 'mcca'

February 17, 2026

Type Package

Title Multi-Category Classification Accuracy

Version 0.8.1

Maintainer Jialiang Li <stalj@nus.edu.sg>

Description It contains six common multi-category classification accuracy evaluation measures.
All of these measures could be found in Li and Ming (2019) <doi:10.1002/sim.8103>. Specifically,
Hypervolume Under Manifold (HUM), described in
Li and Fine (2008) <doi:10.1093/biostatistics/kxm050>.
Correct Classification Percentage (CCP), Integrated Discrimination Improvement (IDI), Net Reclassification Improvement (NRI), R-Squared Value (RSQ), described in
Li, Jiang and Fine (2013) <doi:10.1093/biostatistics/kxs047>.
Polytomous Discrimination Index (PDI), described in
Van Calster et al. (2012) <doi:10.1007/s10654-012-9733-3>.
Li et al. (2018) <doi:10.1177/0962280217692830>.
PDI with variance estimation using Dover et al. (2021) <doi:10.1002/sim.9187>.
We described all these above measures and our mcca package in
Li, Gao and D'Agostino (2019) <doi:10.1002/sim.8103>.

License GPL

Encoding UTF-8

Imports nnet,rpart,e1071,MASS,stats,pROC,caret

Suggests rgl

URL https://github.com/gaoming96/mcca

BugReports https://github.com/gaoming96/mcca/issues

NeedsCompilation no

Author Ming Gao [aut],
Jialiang Li [aut, cre],
Pan Liu [aut],
Ben Van Calster [aut]

Repository CRAN

Date/Publication 2026-02-17 07:30:02 UTC

Contents

`mcca-package` *Diagnostic accuracy methods for classifiers*

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. Additionally, PDI with variance estimation using Dover et al. (2021) method is provided via the pdi_var function.

Details

Package:	mcca
Type:	Package
Version:	0.8.0
Date:	2026-02-06
License:	GPL

Functions

`ccp`	Calculate CCP Value
`hum`	Calculate HUM Value
`plot.mcca.hum`	Plot 3D ROC curve
`idi`	Calculate IDI Value
`nri`	Calculate NRI Value
`pdi`	Calculate PDI Value
`pdi_var`	Calculate PDI Value with Variance Estimation (Dover et al. 2021)
`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)
```

Citation

Li J, Gao M, D'Agostino R. Evaluating classification accuracy for modern learning approaches. Statistics in Medicine. 2019;1-27. https://doi.org/10.1002/sim.8103

Author(s)

Ming Gao, Jialiang Li

Maintainer: Ming Gao <gaoming@umich.edu>

References

Li, J., Gao, M., D'Agostino, R. (2019). Evaluating Classification Accuracy for Modern Learning Approaches. Statistics in Medicine (Tutorials in Biostatistics). 38(13): 2477-2503.

Li, Ming G., D'Agostino. (2019). Evaluating Classification Accuracy for Modern Learning Approaches. Statistics in Medicine.

Li, J. and Fine, J. P. (2008): ROC analysis with multiple tests and multiple classes: methodology and applications in microarray studies. Biostatistics. 9 (3): 566-576.

Li, J., Chow, Y., Wong, W.K., and Wong, T.Y. (2014). Sorting Multiple Classes in Multi-dimensional ROC Analysis: Parametric and Nonparametric Approaches. Biomarkers. 19(1): 1-8.

Li, J., Jiang, B. and Fine, J. P. (2013). Multicategory reclassification statistics for assessing Improvements in diagnostic accuracy. Biostatistics. 14(2): 382-394.

Li, J., Jiang, B., and Fine, J. P. (2013). Letter to Editor: Response. Biostatistics. 14(4): 809-810.

Van Calster B, Vergouwe Y, Looman CWN, Van Belle V, Timmerman D and Steyerberg EW. Assessing the discriminative ability of risk models for more than two outcome categories. European Journal of Epidemiology 2012; 27: 761 C 770.

Li, J., Feng, Q., Fine, J.P., Pencina, M.J., Van Calster, B. (2018). Nonparametric estimation and inference for polytomous discrimination index. Statistical Methods in Medical Research. 27(10): 3092—3103.

See Also

CRAN packages **HUM** for HUM.

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

Examples

```
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]
ccp(y = label, d = data, method = "multinom",maxit = 1000,MaxNWts = 2000,trace=FALSE)
ccp(y = label, d = data, method = "multinom")
ccp(y = label, d = data, method = "svm")
ccp(y = label, d = data, method = "svm",kernel="sigmoid",cost=4,scale=TRUE,coef0=0.5)
ccp(y = label, d = data, method = "tree")
p = as.numeric(label)
ccp(y = label, d = p, method = "label")
hum(y = label, d = data,method = "multinom")
hum(y = label, d = data,method = "svm")
hum(y = label, d = data,method = "svm",kernel="linear",cost=4,scale=TRUE)
hum(y = label, d = data, method = "tree")
ests(y = label, d = data,acc="hum",level=0.95,method = "multinom",trace=FALSE)

## $value
## [1] 0.9972

## $sd
## [1] 0.002051529

## $interval
## [1] 0.9935662 1.0000000
```

ccp *Calculate CCP Value*

Description

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

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

Arguments

`y`	The multinomial response vector with two, three or four categories.
`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.
`...`	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

Returns an object of class "mcca.ccp". The CCP value of the classification using a particular learning method on a set of marker(s).

An object of class "mcca.ccp" is a list containing at least the following components:

`call`	the matched call.
`measure`	the value of measure.
`table`	the category-specific value of measure.

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: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Li, J., Gao, M., D'Agostino, R. (2019). Evaluating Classification Accuracy for Modern Learning Approaches. Statistics in Medicine (Tutorials in Biostatistics). 38(13): 2477-2503.

Li, J., Jiang, B. and Fine, J. P. (2013). Multicategory reclassification statistics for assessing Improvements in diagnostic accuracy. Biostatistics. 14(2): 382-394.

Li, J., Jiang, B., and Fine, J. P. (2013). Letter to Editor: Response. Biostatistics. 14(4): 809-810.

See Also

pdi

Examples

```
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]
ccp(y = label, d = data, method = "multinom",maxit = 1000,MaxNWts = 2000,trace=FALSE)

## Call:
## ccp(y = label, d = data, method = "multinom", maxit = 1000,
##     MaxNWts = 2000, trace = FALSE)

## Overall Correct Classification Probability:
##  0.9866667

## Category-specific Correct Classification Probability:
##   CATEGORIES VALUES PREVALENCE
## 1     setosa   1.00  0.3333333
## 2 versicolor   0.98  0.3333333
## 3  virginica   0.98  0.3333333

ccp(y = label, d = data, method = "multinom")
ccp(y = label, d = data, method = "svm")
ccp(y = label, d = data, method = "svm",kernel="sigmoid",cost=4,scale=TRUE,coef0=0.5)
ccp(y = label, d = data, method = "tree")

p = as.numeric(label)
```

```
ccp(y = label, d = p, method = "label")


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_new[i] = 9
  }else{
    mtcars$carb_new[i] = mtcars$carb[i]
  }
}
data <- data.matrix(mtcars[, c(1)])
mtcars$carb_new <- factor(mtcars$carb_new)
label <- mtcars$carb_new
str(mtcars)
ccp(y = as.numeric(label), d = data, method = "svm",kernel="radial",cost=1,scale=TRUE)

## Call:
## ccp(y = as.numeric(label), d = data, method = "svm", kernel = "radial", cost = 1, scale = TRUE)

## Overall Correct Classification Probability:
##  0.4375

## Category-specific Correct Classification Probability:
##   CATEGORIES    VALUES PREVALENCE
## 1          1 0.5714286    0.21875
## 2          2 0.2000000    0.31250
## 3          3 0.8000000    0.31250
## 4          4 0.0000000    0.15625
```

---

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

```
estp(y, m1, m2, acc="idi", level=0.95, method="multinom", 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 probablity 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 probablity 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.

`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: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

See Also

`ests`

Examples

```
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_new[i] = 9
  }else{
    mtcars$carb_new[i] = mtcars$carb[i]
  }
}
data <- data.matrix(mtcars[, c(1,2)])
mtcars$carb_new <- factor(mtcars$carb_new)
label <- mtcars$carb_new
str(mtcars)
estp(y = label, m1 = data[, 1], m2 = data[, 2], acc=”idi”,method=”lda”, 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”,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

```
ests(y, d, acc=”hum”, level=0.95, method=”multinom”, 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.

`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: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

See Also

`estp`

Examples

```
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]
ests(y = label, d = data,acc="hum",level=0.95,method = ”multinom”,B=10,trace=FALSE)

## $value
## [1] 0.9972

## $se
## [1] 0.002051529

## $interval
## [1] 0.9935662 1.0000000

ests(y = label, d = data,acc="pdi",level=0.85,method = ”tree”,B=10)

## $value
## [1] 0.9213333

## $se
## [1] 0.02148812

## $interval
## [1] 0.9019608 0.9629630

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_new[i] = 9
  }else{
    mtcars$carb_new[i] = mtcars$carb[i]
  }
}
data <- data.matrix(mtcars[, c(1:2)])
mtcars$carb_new <- factor(mtcars$carb_new)
label <- mtcars$carb_new
str(mtcars)

ests(y = label, d = data,acc="hum",level=0.95,method = ”multinom”,trace=FALSE,B=5)

## $value
## [1] 0.2822857
## $se
## [1] 0.170327

## $interval
## [1] 0.2662500 0.4494643
```

hum *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", ...)
```

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 probablity 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.

`...` 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: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Li, J., Gao, M., D'Agostino, R. (2019). Evaluating Classification Accuracy for Modern Learning Approaches. Statistics in Medicine (Tutorials in Biostatistics). 38(13): 2477-2503.

Li, J. and Fine, J. P. (2008): ROC analysis with multiple tests and multiple classes: methodology and applications in microarray studies. Biostatistics. 9 (3): 566-576.

Li, J., Chow, Y., Wong, W.K., and Wong, T.Y. (2014). Sorting Multiple Classes in Multi-dimensional ROC Analysis: Parametric and Nonparametric Approaches. Biomarkers. 19(1): 1-8.

See Also

pdi

Examples

```
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]
hum(y = label, d = data,method = "multinom")
## [1] 0.9972
hum(y = label, d = data,method = "svm")
## [1] 0.9964
hum(y = label, d = data,method = "svm",type="C",kernel="linear",cost=4,scale=TRUE)
## [1] 0.9972
hum(y = label, d = data, method = "tree")
## [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 probablity assessment vector
head(pp)
```

```
hum(y = label, d = pp, method = "prob")
## [1] 0.9972

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_new[i] = 9
  }else{
    mtcars$carb_new[i] = mtcars$carb[i]
  }
}
data <- data.matrix(mtcars[, c(1:10)])
mtcars$carb_new <- factor(mtcars$carb_new)
label <- mtcars$carb_new
str(mtcars)

hum(y = label, d = data, method = "tree",control = rpart::rpart.control(minsplit = 5))
## [1] 1
hum(y = label, d = data, method = "svm",kernel="linear",cost=0.7,scale=TRUE)
## [1] 1
hum(y = label, d = data, method = "svm", kernel ="radial",cost=0.7,scale=TRUE)
## [1] 0.53
```

idi *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

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

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 probablity 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 probablity 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;"prob": m1 & m2 are risk matrices resulted from any external classification algorithm obtained by the user.

... 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: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Li, J., Gao, M., D'Agostino, R. (2019). Evaluating Classification Accuracy for Modern Learning Approaches. Statistics in Medicine (Tutorials in Biostatistics). 38(13): 2477-2503.

Li, J., Jiang, B. and Fine, J. P. (2013). Multicategory reclassification statistics for assessing Improvements in diagnostic accuracy. Biostatistics. 14(2): 382—394.

Li, J., Jiang, B., and Fine, J. P. (2013). Letter to Editor: Response. Biostatistics. 14(4): 809-810.

See Also

`nri`

Examples

```
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_new[i] = 9
  }else{
    mtcars$carb_new[i] = mtcars$carb[i]
  }
}
data <- data.matrix(mtcars[, c(1,5)])
mtcars$carb_new <- factor(mtcars$carb_new)
label <- mtcars$carb_new
str(mtcars)

idi(y = label, m1 = data[, 1], m2 = data[, 2], "tree")
## [1] 0.09979413
idi(y = label, m1 = data[, 1], m2 = data[, 2], "tree",control=rpart::rpart.control(minsplit=4))
## [1] 0.2216707
```

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", ...)
```

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 probablity 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 probablity 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.

`...` 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: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Li, J., Gao, M., D'Agostino, R. (2019). Evaluating Classification Accuracy for Modern Learning Approaches. Statistics in Medicine (Tutorials in Biostatistics). 38(13): 2477-2503.

Li, J., Jiang, B. and Fine, J. P. (2013). Multicategory reclassification statistics for assessing Improvements in diagnostic accuracy. Biostatistics. 14(2): 382—394.

Li, J., Jiang, B., and Fine, J. P. (2013). Letter to Editor: Response. Biostatistics. 14(4): 809-810.

See Also

`idi`

Examples

```
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_new[i] = 9
  }else{
    mtcars$carb_new[i] = mtcars$carb[i]
  }
}
data <- data.matrix(mtcars[, c(1,5)])
mtcars$carb_new <- factor(mtcars$carb_new)
label <- mtcars$carb_new
str(mtcars)

nri(y = label, m1 = data[, 1], m2 = data[, 2], "lda")
## [1] 0.09375
nri(y = label, m1 = data[, 1], m2 = data[, 2], "tree")
## [1] 0.0625
nri(y = label, m1 = data[, 1], m2 = data[, 2], "tree",control=rpart::rpart.control(minsplit=4))
## [1] 0.1875
```

pdi *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

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

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 probablity 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.
...	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

Returns an object of class "mcca.pdi". The PDI value of the classification using a particular learning method on a set of marker(s).

An object of class "mcca.pdi" is a list containing at least the following components:

`call`	the matched call.
`measure`	the value of measure.
`table`	the category-specific value of measure.

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: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Li, J., Gao, M., D'Agostino, R. (2019). Evaluating Classification Accuracy for Modern Learning Approaches. Statistics in Medicine (Tutorials in Biostatistics). 38(13): 2477-2503.

Van Calster B, Vergouwe Y, Looman CWN, Van Belle V, Timmerman D and Steyerberg EW. Assessing the discriminative ability of risk models for more than two outcome categories. European Journal of Epidemiology 2012; 27: 761 C 770.

Li, J., Feng, Q., Fine, J.P., Pencina, M.J., Van Calster, B. (2018). Nonparametric estimation and inference for polytomous discrimination index. Statistical Methods in Medical Research. 27(10): 3092—3103.

See Also

`hum`

Examples

```
str(iris)
data <- iris[, 3]
label <- iris[, 5]
pdi(y = label, d = data,method = "multinom")

## Call:
## pdi(y = label, d = data, method = "multinom")

## Overall Polytomous Discrimination Index:
##  0.9845333

## Category-specific Polytomous Discrimination Index:
##   CATEGORIES VALUES
## 1          1 1.0000
## 2          2 0.9768
## 3          3 0.9768

pdi(y = label, d = data,method = "tree")
pdi(y = label, d = data,method = "tree",control = rpart::rpart.control(minsplit = 200))

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 probablity assessment vector
head(pp)
pdi(y = label, d = pp, method = "prob")
```

pdi_var *Calculate PDI Value with Variance Estimation*

Description

Compute the Polytomous Discrimination Index (PDI) value with bootstrap variance estimation and confidence intervals for two or three or four categories classifiers. This function implements the variance estimation method described in Dover et al. (2021).

Usage

```
pdi_var(y, d, method="multinom", B=250, level=0.95, ...)
```

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.

`B` Number of bootstrap samples. Default is 250.

`level` The confidence level. Default value is 0.95.

`...` Additional arguments in the chosen method's function.

Details

The function returns the PDI value with standard error and confidence interval based on bootstrap resampling. This implements the variance estimation method from Dover et al. (2021) for the polytomous discrimination index.

The PDI measures the discrimination ability of a classification model by calculating the probability that a randomly selected subject from each outcome category is correctly ranked by the model's predicted probabilities.

Value

Returns an object of class "mcca.pdi.var". The PDI value with variance estimation using a particular learning method on a set of marker(s).

An object of class "mcca.pdi.var" is a list containing at least the following components:

`call` the matched call.

`measure` the overall PDI value.

`se` the bootstrap standard error of the overall PDI.

`ci` the confidence interval for the overall PDI.

`level` the confidence level used.

`B` the number of bootstrap samples used.

`table` a data frame with category-specific PDI values, standard errors, and confidence intervals.

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.

The number of bootstrap samples (B) controls the precision of the variance estimate. Larger values give more stable estimates but take longer to compute.

Author(s)

Ming Gao: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

Variance estimation method by Anamaria Savu, PhD (Canadian VIGOUR Centre, University of Alberta)

References

Dover, D.C., Savu, A., Engel, B. (2021). Polytomous discrimination index: Estimation and inference. Statistics in Medicine.

Li, J., Gao, M., D'Agostino, R. (2019). Evaluating Classification Accuracy for Modern Learning Approaches. Statistics in Medicine (Tutorials in Biostatistics). 38(13): 2477-2503.

Van Calster B, Vergouwe Y, Looman CWN, Van Belle V, Timmerman D and Steyerberg EW. Assessing the discriminative ability of risk models for more than two outcome categories. European Journal of Epidemiology 2012; 27: 761-770.

Li, J., Feng, Q., Fine, J.P., Pencina, M.J., Van Calster, B. (2018). Nonparametric estimation and inference for polytomous discrimination index. Statistical Methods in Medical Research. 27(10): 3092-3103.

See Also

pdi, ests

Examples

```
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]

# PDI with variance estimation using 50 bootstrap samples
pdi_var(y = label, d = data, method = "multinom", B = 50, trace = FALSE)

## Call:
## pdi_var(y = label, d = data, method = "multinom", B = 50, trace = FALSE)

## Overall Polytomous Discrimination Index:
##  0.9848
## Standard Error:
##  0.0089

## 95% Confidence Interval:
##  [0.9674, 1.0000]

## Bootstrap Samples: 50

## Category-specific Polytomous Discrimination Index:
##  CATEGORIES VALUES     SE LOWER_CI UPPER_CI
##      setosa 1.0000 0.0000   1.0000   1.0000
##  versicolor 0.9772 0.0133   0.9511   1.0000
##   virginica 0.9772 0.0133   0.9511   1.0000


# Using tree method
pdi_var(y = label, d = data, method = "tree", B = 50)

# Using probability matrix directly
require(nnet)
fit <- multinom(label ~ ., data = data.frame(label = label, data),
                maxit = 1000, MaxNWts = 2000, trace = FALSE)
predict.probs <- predict(fit, type = "probs")
pp <- data.frame(predict.probs)
pdi_var(y = label, d = pp, method = "prob", B = 50)
```

plot.mcca.hum *Plot 3D ROC surface*

Description

plot the 3D ROC surface for a three-category classifier using the 3-dimensional point coordinates, computed by obj which is a mcca.hum class.

Usage

```
## S3 method for class 'mcca.hum'
plot(x,labs=levels(x$y),coords=1:3,nticks=5,filename='fig.png',cex=0.7, ...)
```

Arguments

`x`	An mcca.hum class object, containing probability matrix and labels.
`labs`	The label names of three coordinates. Default is ‘levels(x$y)‘.
`coords`	The coordinates markers. Default is ‘c(1,2,3)‘, which means labs[1] is the x-axis (class 1), labs[2] is the z-axis (class 3) and labs[3] is the y-axis (class 2).
`nticks`	Suggested number of ticks.

`filename` Filename to save snapshot.

`cex` Size for text.

`...` further arguments to 'plot.default'.

Details

This function is to plot the 3D ROC surface according to the correct classification probabilities for the three categories, resulted from any statistical or machine learning methods. This function complements the HUM package which can only plot 3D ROC surface for a single diagnostic test or biomarker for three classes.

Value

The function doesn't return any value.

Author(s)

Ming Gao: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Li, J., and Zhou, X. H. (2009). Nonparametric and semiparametric estimation of the three way receiver operating characteristic surface. Journal of Statistical Planning and Inference. 139: 4133—4142.

Li, J., Gao, M., D'Agostino, R. (2019). Evaluating Classification Accuracy for Modern Learning Approaches. Statistics in Medicine (Tutorials in Biostatistics). 38(13): 2477-2503.

Examples

```
data <- iris[, 1]
label <- iris[, 5]
a=hum(y = label, d = data,method = "multinom")
#plot(a,filename='fig.png')
```

pm *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

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

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.
`...`	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: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Li, J. and Fine, J. P. (2008): ROC analysis with multiple tests and multiple classes: methodology and applications in microarray studies. Biostatistics. 9 (3): 566-576.

Li, J., Chow, Y., Wong, W.K., and Wong, T.Y. (2014). Sorting Multiple Classes in Multi-dimensional ROC Analysis: Parametric and Nonparametric Approaches. Biomarkers. 19(1): 1-8.

See Also

`pdi`

Examples

```
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]
pm(y = label, d = data,method = "multinom")
```

print.mcca.ccp *Print Method for mcca ccp class*

Description

An S3 method for the print generic. It is designed for a quick look at CCP values.

Usage

```
## S3 method for class 'mcca.ccp'
print(x, ...)
```

Arguments

x	object of class ‘mcca.ccp‘.
...	further arguments to ‘print.default‘.

Value

An object of class "mcca.ccp" is a list containing at least the following components:

call	the matched call.
measure	the value of measure.
table	the category-specific value of measure.

Author(s)

Ming Gao: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Li, J., Jiang, B. and Fine, J. P. (2013). Multicategory reclassification statistics for assessing Improvements in diagnostic accuracy. Biostatistics. 14(2): 382-394.

Li, J., Jiang, B., and Fine, J. P. (2013). Letter to Editor: Response. Biostatistics. 14(4): 809-810.

Examples

```
data = iris[, 1:4]
label = iris[, 5]
ccp_object=ccp(y = label, d = data, method = "multinom",maxit = 1000,MaxNWts = 2000,trace=FALSE)
print(ccp_object)
```

print.mcca.hum *Print Method for mcca hum class*

Description

An S3 method for the print generic. It is designed for a quick look at hum values.

Usage

```
## S3 method for class 'mcca.hum'
print(x, ...)
```

Arguments

x	object of class ‘mcca.hum‘.
...	further arguments to ‘print.default‘.

Value

An object of class "mcca.hum" is a list containing at least the following components:

call	the matched call.
measure	the value of measure.
table	the category-specific value of measure.

Author(s)

Ming Gao: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Li, J., Jiang, B. and Fine, J. P. (2013). Multicategory reclassification statistics for assessing Improvements in diagnostic accuracy. Biostatistics. 14(2): 382-394.

Li, J., Jiang, B., and Fine, J. P. (2013). Letter to Editor: Response. Biostatistics. 14(4): 809-810.

Examples

```
data = iris[, 1:4]
label = iris[, 5]
hum_object=hum(y = label, d = data, method = "multinom",maxit = 1000,MaxNWts = 2000,trace=FALSE)
print(hum_object)
```

print.mcca.pdi *Print Method for mcca pdi class*

Description

An S3 method for the print generic. It is designed for a quick look at pdi values.

Usage

```
## S3 method for class 'mcca.pdi'
print(x, ...)
```

Arguments

x	object of class ‘mcca.pdi‘.
...	further arguments to ‘print.default‘.

Value

An object of class "mcca.pdi" is a list containing at least the following components:

call	the matched call.
measure	the value of measure.
table	the category-specific value of measure.

Author(s)

Ming Gao: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Li, J., Jiang, B. and Fine, J. P. (2013). Multicategory reclassification statistics for assessing Improvements in diagnostic accuracy. Biostatistics. 14(2): 382-394.

Li, J., Jiang, B., and Fine, J. P. (2013). Letter to Editor: Response. Biostatistics. 14(4): 809-810.

Examples

```
data = iris[, 1:4]
label = iris[, 5]
pdi_object=pdi(y = label, d = data, method = "multinom",maxit = 1000,MaxNWts = 2000,trace=FALSE)
print(pdi_object)
```

print.mcca.pdi.var *Print Method for mcca pdi.var class*

Description

An S3 method for the print generic. It is designed for a quick look at PDI values with variance estimation.

Usage

```
## S3 method for class 'mcca.pdi.var'
print(x, ...)
```

Arguments

x	object of class ‘mcca.pdi.var‘.
...	further arguments to ‘print.default‘.

Value

An object of class "mcca.pdi.var" is a list containing at least the following components:

call	the matched call.
measure	the overall PDI value.
se	the bootstrap standard error of the overall PDI.
ci	the confidence interval for the overall PDI.
level	the confidence level used.
B	the number of bootstrap samples used.
table	a data frame with category-specific PDI values, standard errors, and confidence intervals.

Author(s)

Ming Gao: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Dover, D.C., Savu, A., Engel, B. (2021). Polytomous discrimination index: Estimation and inference. Statistics in Medicine.

Li, J., Gao, M., D’Agostino, R. (2019). Evaluating Classification Accuracy for Modern Learning Approaches. Statistics in Medicine (Tutorials in Biostatistics). 38(13): 2477-2503.

Examples

```
data = iris[, 1:4]
label = iris[, 5]
pdi_var_object = pdi_var(y = label, d = data, method = "multinom",
                         B = 50, trace = FALSE)
print(pdi_var_object)
```

`print.mcca.rsq` *Print Method for mcca rsq class*

Description

An S3 method for the print generic. It is designed for a quick look at rsq values.

Usage

```
## S3 method for class 'mcca.rsq'
print(x, ...)
```

Arguments

`x`	object of class ‘mcca.rsq‘.
`...`	further arguments to ‘print.default‘.

Value

An object of class "mcca.rsq" is a list containing at least the following components:

`call`	the matched call.
`measure`	the value of measure.
`table`	the category-specific value of measure.

Author(s)

Ming Gao: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Li, J., Jiang, B. and Fine, J. P. (2013). Multicategory reclassification statistics for assessing Improvements in diagnostic accuracy. Biostatistics. 14(2): 382-394.

Li, J., Jiang, B., and Fine, J. P. (2013). Letter to Editor: Response. Biostatistics. 14(4): 809-810.

Examples

```
data = iris[, 1:4]
label = iris[, 5]
rsq_object=rsq(y = label, d = data, method = "multinom",maxit = 1000,MaxNWts = 2000,trace=FALSE)
print(rsq_object)
```

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

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

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 probablity 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.

... 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

Returns an object of class "mcca.rsq". The RSQ value of the classification using a particular learning method on a set of marker(s).

An object of class "mcca.rsq" is a list containing at least the following components:

`call` the matched call.

`measure` the value of measure.

`table` the category-specific value of measure.

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: gaoming@umich.edu

Jialiang Li: stalj@nus.edu.sg

References

Li, J., Gao, M., D'Agostino, R. (2019). Evaluating Classification Accuracy for Modern Learning Approaches. Statistics in Medicine (Tutorials in Biostatistics). 38(13): 2477-2503.

Li, J., Jiang, B. and Fine, J. P. (2013). Multicategory reclassification statistics for assessing Improvements in diagnostic accuracy. Biostatistics. 14(2): 382-394.

Li, J., Jiang, B., and Fine, J. P. (2013). Letter to Editor: Response. Biostatistics. 14(4): 809-810.

See Also

ccp

Examples

```
str(iris)
data <- iris[, 1:4]
label <- iris[, 5]
rsq(y = label, d = data, method="multinom")

## Call:
## rsq(y = label, d = data, method = "multinom")

## Overall R-squared value:
##  0.9637932
## Category-specific R-squared value:
##   CATEGORIES    VALUES
## 1     setosa 0.9999824
## 2 versicolor 0.9456770
## 3  virginica 0.9457203

rsq(y = label, d = data, method = "tree")


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")


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_new[i] = 9
  }else{
    mtcars$carb_new[i] = mtcars$carb[i]
  }
}
data <- data.matrix(mtcars[, c(1)])
mtcars$carb_new <- factor(mtcars$carb_new)
label <- mtcars$carb_new
str(mtcars)

rsq(y = label, d = data, method="tree")
rsq(y = label, d = data, method="lda")
rsq(y = label, d = data, method="lda",prior = c(100,1,1,1)/103)
```

Index