Package ‘optAUC’

February 20, 2015

Type Package

Title Optimal Combinations of Diagnostic Tests Based on AUC

Version 1.0

Date 2013-03-31

Author Xin Huang, Gengsheng Qin, Yixin Fang

Maintainer Xin Huang <xhuang.fhcrc@gmail.com>

Depends R (>= 2.15.2), MASS

Description Searches for optimal linear combination of multiple diagnostic tests (markers) that maximizes the area under the receiver operating characteristic curve (AUC); performs an approximated cross-validation for estimating the AUC associated with the estimated coefficients.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2013-04-01 07:50:08

R topics documented:

`optAUC-package` *Optimal Combinations of Diagnostic Tests Based on AUC*

Description

Searches for optimal linear combination of multiple diagnostic tests (markers) that maximizes the area under the receiver operating characteristic curve (AUC); performs an approximated cross-validation for estimating the AUC associated with the estimated coefficients.

Details

Package:	optAUC
Type:	Package
Version:	1.0
Date:	2013-03-31
License:	GPL-2

Author(s)

Xin Huang, Gengsheng Qin, Yixin Fang
Maintainer: Xin Huang <xhuang.fhcrc@gmail.com>

References

Huang X, Qin G, Fang Y. (2011) Optimal Combinations of Diagnostic Tests Based on AUC. Biometrics. Jun;67(2):568-76.
http://www.ncbi.nlm.nih.gov/pubmed/20560934

Examples

```
rho<-0
m<-50
n<-50
y1.sd<-0.5
y2.sd<-0.5
y1.mean<-2
y2.mean<-1
lambda <- 5

set.seed(88)
# generate non-diseased population F(X1, X2)
# the sample from 2-dimensinal multinormal distribution with mean 0 and std=1
X1X2<-mvrnorm(m, c(1,1), matrix(c(0.5,rho,rho,0.5),2,2))

# generate  diseased population G(Y1,Y2)
```

```
# the sample from 2-dimensinal multinormal distribution with mean
# (y1.mean,y2.mean) and std=(y1.sd,y2.sd)
Y1Y2<-mvrnorm(n, c(y1.mean,y2.mean), matrix(c(y1.sd^2,rho*y1.sd*y2.sd, rho*y1.sd*y2.sd, y2.sd^2),2,2))

# only the first marker, the "true" model, should have the maximum AUC amount all models
optAUC(X1X2, Y1Y2, column.select=1)
# two markers in the model, the AUC from GCV is smaller than just first marker in the model, because the second mar
# the AUC from ACV (apearent estimate by substituting the estimated beta into the model) is larger than previous m
optAUC(X1X2, Y1Y2, column.select=c(1:2))
```

AUC *Function for AUC with sigmoid estimate*

Description

Function for AUC with sigmoid estimate

Usage

```
AUC(beta, Z, lambda)
```

Arguments

`beta`	linear coefficients for linear combinations of multiple diagnostic tests
`Z`	the Y[i]-X[j]
`lambda`	the smooth parameter for the Sigmoid function used for the AUC

Author(s)

Maintainer: Xin Huang <xhuang.fhcrc@gmail.com>

AUC.emp *Function for AUC when input is X and Y*

Description

NA

Usage

```
AUC.emp(X, Y)
```

Arguments

`X`	m X p data matrix for m non-diseased subjects with p markers
`Y`	n X p data matrix for n diseased subjects with p markers

Author(s)

Maintainer: Xin Huang <xhuang.fhcrc@gmail.com>

`beta2theta` *Function to translate beta into theta, the n-sphere constrain*

Description

Function to translate beta into theta, the n-sphere constrain

Usage

```
beta2theta(beta)
```

Arguments

`beta`	coefficients for linear combination of multiple diagnostic tests

---

`betahat` *Function for estimating beta using kernal function*

Description

Function for estimating beta using kernal function

Usage

```
betahat(X, Y, init, lambda)
```

Arguments

`X`	m X p data matrix for m non-diseased subjects with p markers
`Y`	n X p data matrix for n diseased subjects with p markers
`init`	initial value of the linear coefficients for the optimization algorithm
`lambda`	the smooth parameter for the Sigmoid function used for the AUC

gradAUC.Lang — *Function for gradient of AUC after applying Lagrange Multiplyer*

Description

Function for gradient of AUC after applying Lagrange Multiplyer

Usage

```
gradAUC.Lang(par, Z, lambda)
```

Arguments

`par`	paramter
`Z`	Z
`lambda`	the smooth parameter for the Sigmoid function used for the AUC

gradsqr — *The function of grad_square in the GCV*

Description

The function of grad_square in the GCV

Usage

```
gradsqr(beta, X, Y, lambda)
```

Arguments

`beta`	linear coefficients
`X`	m X p data matrix for m non-diseased subjects with p markers
`Y`	n X p data matrix for n diseased subjects with p markers
`lambda`	the smooth parameter for the Sigmoid function used for the AUC

hessAUC *function for hessian matrix of AUC*

Description

function for hessian matrix of AUC

Usage

```
hessAUC(beta, X, Y, lambda)
```

Arguments

beta	linear coefficients
X	m X p data matrix for m non-diseased subjects with p markers
Y	n X p data matrix for n diseased subjects with p markers
lambda	the smooth parameter for the Sigmoid function used for the AUC

nlsolve *solve nonlinear function*

Description

solve nonlinear function

Usage

```
nlsolve(par, fn, method = "BFGS", nstarts = 1, ...)
```

Arguments

par	parameter
fn	function
method	method
nstarts	tries
...	other paramters pass to fn

optAUC *Optimal Combinations of Diagnostic Tests Based on AUC*

Description

Searches for optimal linear combination of multiple diagnostic tests (markers) that maximizes the area under the receiver operating characteristic curve (AUC); performs an approximated cross-validation for estimating the AUC associated with the estimated coefficients.

Usage

```
optAUC(X, Y, column.select = c(1:ncol(X)), lambda = 5, scale = TRUE)
```

Arguments

`X`	m X p data matrix for m non-diseased subjects with p markers
`Y`	n X p data matrix for n diseased subjects with p markers
`column.select`	which of the p markers are used for the combination, default is all p columns
`lambda`	the smooth parameter for the Sigmoid function used for the AUC
`scale`	a logic indicator whether performs standardization to the dataset before the combination, default is true

Details

When several diagnostic tests are available, one can combine them to achieve better diagnostic accuracy. This program considers the optimal linear combination that maximizes the area under the receiver operating characteristic curve (AUC); the estimates of the combination's coefficients is obtained via a nonparametric procedure. Further, for estimating the AUC associated with the estimated coefficients, this progam outputs two estimates: one is an apparent estimation by re-substitution (ACV), which is too optimistic; the other is an approximated cross-validation (GCV) estimation. Notice that, the GCV can be applied for variable selection to select important diagnostic tests\markers. See reference for more details.

Value

`beta`	the estimated linear coefficients, under a unit-sphere constraint
`ACV`	apparent estimation of AUC of the composite score by re-substitution of the linear coefficients
`GCV`	the approximated cross-validation estimation of AUC of the composite score
`converge`	an indicator for the convergency status of the optimization algorithm, 1 means converge, 0 means converge criteria not meet

Note

It is recommended to rescale or monotonic transfer of the data first if significant outliners exists, e.g. log transfer. The AUC is invariant to any monotonic transformation of the data; however, the sigmoid approximation of the AUC may be affected by outliners.
The estimated linear coefficients are based on the standardized (if the parameter scale=TRUE) input data. Thus, composite scores = beta%*%scale(rbind(X,Y)).

Author(s)

Xin Huang, Gengsheng Qin, Yixin Fang
Maintainer: Xin Huang <xhuang.fhcrc@gmail.com>

References

Huang X, Qin G, Fang Y. (2011) Optimal Combinations of Diagnostic Tests Based on AUC. Biometrics. Jun;67(2):568-76.
http://www.ncbi.nlm.nih.gov/pubmed/20560934

Examples

```
library(MASS)
rho<-0
m<-50
n<-50
y1.sd<-0.5
y2.sd<-0.5
y1.mean<-2
y2.mean<-1
lambda <- 5

set.seed(88)
# generate non-diseased population F(X1, X2)
# the sample from 2-dimensinal multinormal distribution with mean 0 and std=1
X1X2<-mvrnorm(m, c(1,1), matrix(c(0.5,rho,rho,0.5),2,2))

# generate  diseased population G(Y1,Y2)
# the sample from 2-dimensinal multinormal distribution with mean
# (y1.mean,y2.mean) and std=(y1.sd,y2.sd)
Y1Y2<-mvrnorm(n, c(y1.mean,y2.mean), matrix(c(y1.sd^2,rho*y1.sd*y2.sd, rho*y1.sd*y2.sd, y2.sd^2),2,2))

# only the first marker, the "true" model, should have the maximum AUC amount all models
optAUC(X1X2, Y1Y2, column.select=1)
# two markers in the model, the AUC from GCV is smaller than just first marker in the model, because the second mar
# the AUC from ACV (apearent estimate by substituting the estimated beta into the model) is larger than previous m
optAUC(X1X2, Y1Y2, column.select=c(1:2))
```

theta2beta *Function to translate theta to beta, the n-sphere constrain*

Description

Function to translate theta to beta, the n-sphere constrain

Usage

```
theta2beta(theta)
```

Arguments

theta	the parameter on n unit-sphere constrain

Index