Package: persDx (via r-universe)

September 9, 2024

Type Package

Title Personalized Diagnostics Rules for Subgroup Identification and Personalized Biomarker Discovery

Version 0.5.0

Date 2023-08-15

Description Tailoring the optimal biomarker(s) for disease screening or diagnosis based on subjects' individual characteristics.

License GPL (>= 2)

Depends R (>= 4.2.0), pROC, survivalROC, stats, graphics

NeedsCompilation no

Author Yunro Chung [aut, cre] (<https://orcid.org/0000-0001-9125-9277>)

Maintainer Yunro Chung <yunro.chung@asu.edu>

Date/Publication 2023-08-16 06:30:02 UTC

Repository https://yunrochung.r-universe.dev

RemoteUrl https://github.com/cran/persDx

RemoteRef HEAD

RemoteSha 7bc664b7329865073866926a8f058641694e6869

Contents

persDx-package *Estimating Personalized Diagnostics Rules*

Description

Personalized Recommendation of biomarkers or screening/diagnostic tests based on patients' individual profile.

Details

Package:	persDx
Type:	Package
Version:	0.5.0
Date:	2023-08-15
License:	GPL (>= 2)

Author(s)

Yunro Chung [aut,cre] Maintainer: Yunro Chung <yunro.chung@asu.edu>

References

Yaliang Zhang and Yunro Chung, Nonparametric estimation of linear personalized diagnostics rules via efficient grid algorithm (under revision)

np_lpd *Nonparametric estimation of linear personalized diagnostic rules.*

Description

Nonparametric estimation of the personalized diagnostics rule to find subgroup-specific biomarkers according to linear combination of predictors.

Usage

```
np_lpd(D,YA,YB,X,dirA,dirB,eps,plot,A,B,c,d)
```

Arguments

D	Binary outcome with D=1 for disease (or case) and D=0 for non-diseased (or control) (n X 1 vector).
YA	Biomarker A, measured on a continuous scale (n X 1 vector).
YB	Biomarker B, measured on a continuous scale (n X 1 vector).
X	Predictors (n x p matrix).
dirA	Direction of YA to D, where dirA="<" (or dirA=">") indicates higher (or lower) YA is assoicated with Pr(D=1)). Default is dirA="<".
dirB	Direction of YB to D, where dirB="<" (or dirB=">") indicates higher (or lower) YB is assoicated with Pr(D=1)). Default is dirB="<".
eps	Tuning parameter for predictor selections. Default is eps=0.01.
plot	plot=TRUE (or FALSE) shows (or does not show) the receiver operating characteristics (ROC) curve.
A	Grid search parameter (Discrete). Default is A=0
B	Grid search parameter (Discrete). Default is B=0
c	Grid search parameter. Default is c=2
d	Grid search parameter. Default is d=2

Details

The np.lin.persDx function estimates the personalized diagnostics rule $\tau(X)$, where $\tau(X)$=A recommends YA if $\theta_1 X_1 + ... + \theta_p X_p > \theta_0$ or $\tau(X)$=B recommends YB otherwise by maximizing (empirical) area under the ROC curve (AUC). Here, the AUC is computed based on YC with the direction of "<", i.e. higher YC is associated with Pr(D=1), where $YC = YA$ if $\tau(X)$=A and dirA="<", or $YC = YB$ if $\tau(X)$=B and dirB="<". If dirA=">" (or dirB=">"), negative YA (or YB) is used.

A forward grid rotation algorithm (FGR) is used to estimate $\theta_0, \theta_1, ..., \theta_p$ by sequentially adding each of the predictors to $\tau(X)$ that increases the AUC the most. The stopping criteria is AUC increasement is less than or equal to eps. The eps controls the model complexity. The cross-validation techniques can be used to find the optimal eps.

The FGR results in a suboptimal solution. The accuracy is improved by setting higher A, B, c, d, but it increases increase computational costs, or vice versa. We thus recond this function when p is small or around 10.

Value

A list of class np.lin.persDx:

df	Data frame with D, YA, YB, X, tau, YC, where tau=A or B for recommending YA or YB, respectively.
AUCA	AUC for YA.
AUCB	AUC for YB.
AUC	AUC for YC.

tpfp	Data frame with cutoff, tp, fp, where tp and fp are true and false positive positives at the cutoff values of YC.
theta	Estimated regression parameters.
theta0	Estimated threshold parameter.
PLOT	TRUE or FALSE to show ROC curves.

Author(s)

Yunro Chung [aut, cre]

References

Yaliang Zhang and Yunro Chung, Nonparametric estimation of linear personalized diagnostics rules via efficient grid algorithm (submitted)

Examples

```
#simulate data
set.seed(1)
n=100
D=c(rep(1,n/2),rep(0,n/2))

X1=runif(n,0,1)
X2=runif(n,0,1)
X3=runif(n,0,1)
X=data.frame(X1,X2,X3)

tau=rep("B",n)
tau[X1+X2>=1]="A"

YA=D*(rnorm(n,2,1)*(tau=="A")+rnorm(n,0,1)*(tau=="B"))+
   (1-D)*rnorm(n,0,1)
YB=D*(rnorm(n,1,1)*(tau=="B")+rnorm(n,0,1)*(tau=="A"))+
   (1-D)*rnorm(n,0,1)

#run
fit=np_lpd(D, YA, YB, X)
fit
```

---

np_lpd_survival *Nonparametric estimation of linear personalized diagnostic rules with right-censored survival outcome.*

Description

Nonparametric estimation of personalized diagnostics rule to find subgroup-specific biomarkers according to linear combination of predictors.

Usage

```
np_lpd_survival(Stime,D,YA,YB,X,dirA,dirB,predict.time,span,eps,plot,A,B,c,d)
```

Arguments

Stime	Event time or censoring time for subjects (n x 1 vector).
D	Indicator of status, where D=1 if death or event, and D=0 otherwise (n X 1 vector).
YA	Biomarker A, measured on a continuous or orinal scale (n X 1 vector).
YB	Biomarker B, measured on a continuous or ordinal scale (n X 1 vector).
X	Predictors (n x p matrix).
predict.time	Time point to evaluate YA and YB.
span	Span for the nearest neighbor estimation (NNE).
dirA	Direction of YA to D, where dirA="<" (or dirA=">") indicates higher (or lower) YA is assoicated with Pr(D=1)). Default is dirA="<".
dirB	Direction of YB to D, where dirB="<" (or dirB=">") indicates higher (or lower) YB is assoicated with Pr(D=1)). Default is dirB="<".
eps	Tuning parameter for predictor selections. Default is eps=0.01.
plot	plot=TRUE (or FALSE) shows (or does not show) the receiver operating charactriestics (ROC) curve.
A	Grid search parameter (Discrete). Default is A=0
B	Grid search parameter (Discrete). Default is B=0
c	Grid search parameter. Default is c=2
d	Grid search parameter. Default is d=2

Details

The np.lin.survival.persDx function estimates the personalized diagnostics rule $\tau(X)$, where $\tau(X)$=A recommends YA if $\theta_1 X_1 + ... + \theta_p X_p > \theta_0$ or $\tau(X)$=B recommends YB otherwise by maximizing (empirical) survival area under the ROC curve (AUC) at the predict.time using the Nearest Neighbor Estimation. Here, the survival AUC is computed based on YC with the direction of "<", i.e. higher YC is associated with Pr(D=1), where $YC = YA$ if $\tau(X)$=A and dirA="<", or $YC = YB$ if $\tau(X)$=B and dirB="<". If dirA=">" (or dirB=">"), negative YA (or YB) is used.

A forward grid rotation algorithm (FGR) is used to estimate $\theta_0, \theta_1, ..., \theta_p$ by sequentially adding each of the predictors to $\tau(X)$ that increases the AUC the most. The stopping criteria is AUC increasement is less than or equal to eps. The eps controls the model complexity. The cross-validation techniques can be used to find the optimal eps.

The FGR results in a suboptimal solution. The accuracy is improved by setting higher A, B, c, d, but it increases computational costs. We thus recond this function when p is small or around 10.

Value

A list of class np.lin.persDx:

df	Data frame with Stime, D, YA, YB, X, tau, YC, where tau=A or B for recommending YA or YB, respectively.
AUCA	Survival AUC for YA at predict.time.
AUCB	Survival AUC for YB at predict.time.
AUC	Survival AUC for YC.
tpfp	Data frame with cutoff, tp, fp, where tp and fp are true and false positive positives at the cutoff values of YC.
theta	Estimated regression parameters.
theta0	Estimated threshold parameter.
PLOT	TRUE or FALSE to show survival ROC curves.

Author(s)

Yunro Chung [aut, cre]

References

Yaliang Zhang and Yunro Chung, Nonparametric estimation of linear personalized diagnostics rules via efficient grid algorithm (submitted)

Examples

```
#simulate data
set.seed(1)
n=100
X=abs(rnorm(n,1,1))
C=abs(rnorm(n,1,1))
Stime=pmin(X,C)
D=as.numeric(X<=C)

X1=runif(n,0,1)
X2=runif(n,0,1)
X3=runif(n,0,1)
X=data.frame(X1,X2,X3)

tau=rep("B",n)
tau[X1+X2>=1]="A"

D2=rep(0,n) #event by time 2
D2[which(Stime<=3 & D==1)]=1

YA=D2*(rnorm(n,2,1)*(tau=="A")+rnorm(n,0,1)*(tau=="B"))+
   (1-D2)*rnorm(n,0,1)
YB=D2*(rnorm(n,1,1)*(tau=="B")+rnorm(n,0,1)*(tau=="A"))+
   (1-D2)*rnorm(n,0,1)
```

```
#run
span=0.1
fit=np_lpd_survival(Stime, D, YA, YB, X, predict.time=1, span=span)
fit
```

Index