Package ‘qLearn’

February 20, 2015

Type Package

Title Estimation and inference for Q-learning

Version 1.0

Date 2012-03-01

Author Jingyi Xin, Bibhas Chakraborty, and Eric B. Laber

Maintainer Bibhas Chakraborty `<bc2425@columbia.edu>`

Description Functions to implement Q-learning for estimating optimal dynamic treatment regimes from two stage sequentially randomized trials, and to perform inference via m-out-of-n bootstrap for parameters indexing the optimal regime.

License GPL-2

LazyLoad yes

Repository CRAN

Date/Publication 2012-03-09 10:12:37

NeedsCompilation no

R topics documented:

`chooseMDoubleBootstrap` *Choose the bootstrap sample size for stage 1 inference*

Description

Choose the resample size for stage 1 bootstrap using double bootstrap. The form of m is: $m = n^{\frac{1+Xi(1-pHat)}{1+Xi}}$, where the tuning parameter Xi is chosen via double bootstrap and pHat is the estimated non-regularity level computed by `getModel`. Example could be found under `qLearn`.

Usage

```
chooseMDoubleBootstrap(s2Formula,s1Formula,
completeData,s2Treat,interact,s2Indicator,alpha=0.05,
boot1Num=500,boot2Num=500,...)
```

Arguments

`s2Formula`	stage 2 regression formula
`s1Formula`	Stage 1 regression formula
`completeData`	data frame containing all the variables
`s2Treat`	character string: name of the stage 2 treatment variable
`interact`	character vector: names of variables that interact with s2Treat
`s2Indicator`	character string: names of the stage 2 treatment indicator variable
`alpha`	level of significance
`boot1Num`	numbers of bootstrap sampling for first order bootstrap
`boot2Num`	numbers of bootstrap sampling for second order bootstrap
`...`	other arguments of the `lm` function

Value

`m`	resample size for stage 1 bootstrap

Author(s)

Jingyi Xin <`jx2167@columbia.edu`>, Bibhas Chakraborty <`bc2425@columbia.edu`>, and Eric B.Laber <`eblaber@ncsu.edu`>

References

Chakraborty, B., and Laber, E.B. (2012). Inference for Optimal Dynamic Treatment Regimes using an Adaptive m-out-of-n Bootstrap Scheme. *Submitted.*

See Also

`qLearn`

Examples

```
set.seed(100)
# Simple Simulation on 1000 subjects
sim<-matrix(0,nrow=1000,ncol=7)
colnames(sim)<-c("H1","A1","Y1","H2","A2","Y2","IS2")
sim<-as.data.frame(sim)

# Randomly generate stage 1 covariates and stage 1 and 2 treatments
sim[,c("H1","A1","A2")]<-2*rbinom(1000*3,1,0.5)-1

# Generate stage 2 covariates based on H1 and T1
expit<-exp(0.5*sim$H1+0.5*sim$A1)/(1+exp(0.5*sim$H1+0.5*sim$A1))
sim$H2<-2*rbinom(1000,1,expit)-1

# Assume stage 1 outcome Y1 is 0
# Generate stage 2 outcome Y2
sim$Y2<-0.5*sim$A2+0.5*sim$A2*sim$A1-0.5*sim$A1+rnorm(1000)

# Randomly assign 500 subjects to S2
sim[sample(1000,500),"IS2"]<-1
sim[sim$IS2==0,c("A2","Y2")]<-NA

# Define models for both stages
s2Formula<-Y2~H1*A1+A1*A2+A2:H2
s1Formula<-Y1~H1*A1

# Use boot1Num=boot2Num=20 in the example to save computational time
# In real case should use greater number
m<-chooseMDoubleBootstrap(s2Formula,s1Formula,sim,s2Treat="A2",
interact=c("A1","H2"),s2Indicator="IS2",boot1Num=20,boot2Num=20)
```

`getModel` *Compute the regression coefficients for both stages*

Description

`getModel` computes the regression coefficients for both stages. For stage 1 regression, peudo-outcomes are constructed based on stage 2 coefficients. Additionally it outputs the standard outputs of the `lm` function; however, the inference for stage 1 parameters based on these outputs is invalid due to non-regularity. Example could be found under `qLearn`.

Usage

```
getModel(s2Formula,s1Formula,completeData,
s2Treat,interact,s2Indicator,...)
```

Arguments

`s2Formula`	stage 2 regression formula
`s1Formula`	Stage 1 regression formula
`completeData`	data frame containing all the variables
`s2Treat`	character string: name of the stage 2 treatment variable
`interact`	character vector: names of variables that interact with s2_treat
`s2Indicator`	character string: names of the stage 2 treatment indicator variable
`...`	other arguments of the `lm` function

Value

A list containing:

`s2Model`	stage2 regression model
`s1Model`	stage1 regression model
`pHat`	estimated non-regularity level

Author(s)

Jingyi Xin <`jx2167@columbia.edu`>, Bibhas Chakraborty <`bc2425@columbia.edu`>, and Eric B.Laber <`eblaber@ncsu.edu`>

References

Chakraborty, B., and Laber, E.B. (2012). Inference for Optimal Dynamic Treatment Regimes using an Adaptive m-out-of-n Bootstrap Scheme. *Submitted.*

See Also

`qLearn`

Examples

```
set.seed(100)
# Simple Simulation on 1000 subjects
sim<-matrix(0,nrow=1000,ncol=7)
colnames(sim)<-c("H1","A1","Y1","H2","A2","Y2","IS2")
sim<-as.data.frame(sim)

# Randomly generate stage 1 covariates and stage 1 and 2 treatments
sim[,c("H1","A1","A2")]<-2*rbinom(1000*3,1,0.5)-1

# Generate stage 2 covariates based on H1 and T1
expit<-exp(0.5*sim$H1+0.5*sim$A1)/(1+exp(0.5*sim$H1+0.5*sim$A1))
sim$H2<-2*rbinom(1000,1,expit)-1

# Assume stage 1 outcome Y1 is 0
# Generate stage 2 outcome Y2
```

```
sim$Y2<-0.5*sim$A2+0.5*sim$A2*sim$A1-0.5*sim$A1+rnorm(1000)

# Randomly assign 500 subjects to S2
sim[sample(1000,500),"IS2"]<-1
sim[sim$IS2==0,c("A2","Y2")]<-NA

# Define models for both stages
s2Formula<-Y2~H1*A1+A1*A2+A2:H2
s1Formula<-Y1~H1*A1

## Fit model for both stages
getModel(s2Formula,s1Formula,sim,s2Treat="A2",interact=c("A1","H2"),
s2Indicator="IS2")
```

`qLearn` *Based on the input contrast vectors, compute point estimates and construct confidence intervals using bootstrap for both stages*

Description

Suppose the goal is to find the point estimates and CIs for stage 1 and stage 2 contrasts $C_1^T\theta_1$ and $C_2^T\theta_2$. Given C_1, C_2, regular n-out-of-n bootstrap will be used in stage 2 and different bootstrap scheme can be used in stage 1 analysis by assigning different value to `s1Method`. "Fixed Xi" will fix the Xi value as `fixedXi` and calculate the corresponding m; "Double Bootstrap" will calculate m using double bootstrap method; and the default "Regular" will skip choosing m and go with a regular bootstrap. Also m can be specified in `s1M` if not using "Fixed Xi" or "Double Bootstrap"

Usage

```
qLearn(s2Formula,s1Formula,completeData,
s2Treat,interact,s2Indicator,s2Contrast,s1Contrast,
alpha=0.05,bootNum=1000,s1Method="Regular",fixedXi,
doubleBoot1Num=500,doubleBoot2Num=500,s1M,...)
```

Arguments

`s2Formula`	stage 2 regression formula
`s1Formula`	Stage 1 regression formula
`completeData`	data frame containing all the variables
`s2Treat`	character string: name of the stage 2 treatment variable
`interact`	character vector: names of variables that interact with s2Treat
`s2Indicator`	character string: names of the stage 2 treatment indicator variable
`s2Contrast`	contrast for the stage 2 coefficients
`s1Contrast`	contrast for the stage 1 coefficients
`alpha`	level of significance

`bootNum`	numbers of bootstrap sampling in constructing CIs
`s1Method`	character string: method to choose stage 1 bootstrap sample size, m; "Double Bootstrap" will calculate m using double bootstrap method; "Fixed Xi" will fix the Xi value and calculate the corresponding m; "Regular" will use a regular n-out-of-n bootstrap for stage 1.
`fixedXi`	fixed xi value if `s1Method`="Fixed Xi"
`doubleBoot1Num`	numbers of bootstrap sampling for first order bootstrap if `s1Method`="Double Bootstrap"
`doubleBoot2Num`	numbers of bootstrap sampling for second order bootstrap if `s1Method`="Double Bootstrap"
`s1M`	specify m if necessary
`...`	other arguments of the `lm` function

Value

A list containing:

`s1Coefficients`	stage 1 regression coefficients
`s2Coefficients`	stage 2 regression coefficients
`s1Inference`	stage 1 coefficients confidence interval based on stage1 contrast
`s2Inference`	stage 2 coefficients confidence interval based on stage2 contrast
`s1Size`	stage 1 bootstrap sample size

Author(s)

Jingyi Xin <`jx2167@columbia.edu`>, Bibhas Chakraborty <`bc2425@columbia.edu`>, and Eric B.Laber <`eblaber@ncsu.edu`>

References

Chakraborty, B., and Laber, E.B. (2012). Inference for Optimal Dynamic Treatment Regimes using an Adaptive m-out-of-n Bootstrap Scheme. *Submitted.*

See Also

`getModel` `chooseMDoubleBootstrap`

Examples

```
set.seed(100)
# Simple Simulation on 1000 subjects
sim<-matrix(0,nrow=1000,ncol=7)
colnames(sim)<-c(”H1”,”A1”,”Y1”,”H2”,”A2”,”Y2”,”IS2”)
sim<-as.data.frame(sim)
```

```
# Randomly generate stage 1 covariates and stage 1 and 2 treatments
sim[,c("H1","A1","A2")]<-2*rbinom(1000*3,1,0.5)-1

# Generate stage 2 covariates based on H1 and T1
expit<-exp(0.5*sim$H1+0.5*sim$A1)/(1+exp(0.5*sim$H1+0.5*sim$A1))
sim$H2<-2*rbinom(1000,1,expit)-1

# Assume stage 1 outcome Y1 is 0
# Generate stage 2 outcome Y2
sim$Y2<-0.5*sim$A2+0.5*sim$A2*sim$A1-0.5*sim$A1+rnorm(1000)

# Randomly assign 500 subjects to S2
sim[sample(1000,500),"IS2"]<-1
sim[sim$IS2==0,c("A2","Y2")]<-NA

# Define models for both stages
s2Formula<-Y2~H1*A1+A1*A2+A2:H2
s1Formula<-Y1~H1*A1

## Fixed Xi as 0.05
qLearn(s2Formula,s1Formula,sim,s2Treat="A2",interact=c("A1","H2"),
s2Indicator="IS2",s1Method="Fixed Xi",fixedXi=0.05,bootNum=100)
```

Index