Package 'iqLearn'

May 29, 2018

Type Package

Title Interactive Q-Learning

Version 1.5

Date 2018-5-29

Author Kristin A. Linn, Eric B. Laber, Leonard A. Stefanski

Maintainer Kristin A. Linn <klinn@pennmedicine.upenn.edu>

Description Estimate an optimal dynamic treatment regime using Interactive Q-learning.

License GPL-2

LazyData TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-29 21:12:55 UTC

R topics documented:

bmiData *Adolescent BMI data set (generated toy example)*

Description

This data set was generated to mimic data from a two-stage randomized clinical trial studying the effect of meal replacement shakes on adolescent obesity. It contains the following covariates collected at the start of the first stage: "gender," "race," "parent_BMI," and "baseline_BMI". At the second-stage, "month4_BMI" was collected. Variables "A1" and "A2" are the randomized treatments at stage one and two, and "month12_BMI" is the primary outcome collected at the end of stage two.

Usage

```
bmiData
```

Format

A matrix with rows corresponding to patients.

Source

Generated by Kristin A. Linn in R

`IQ1` *IQ-learning: Recommend stage 1 treatment*

Description

Recommends the IQ-estimated optimal first-stage treatment for a patient with observed stage 1 variables.

Usage

```
IQ1(mainObj, cmObj, sigObj, dens, h1main, h1cm, h1sig)
```

Arguments

`mainObj` object of type `learnIQ1main`

`cmObj` object of type `learnIQ1cm`

`sigObj` object of type `learnIQ1var`

`dens` method of density estimation, either "norm" for normal location-scale density estimate or "nonpar" for the empiricial density estimator

`h1main` vector of observed first-stage main effects corresponding to the variables in `H1Main` used in `learnIQ1main()`

`h1cm` vector of observed first-stage main effects corresponding to the variables in `H1CMean` used in `learnIQ1cm()`

`h1sig` vector of observed first-stage main effects corresponding to the variables in `H1CVar` used in `learnIQ1var()`

Details

Use the estimated optimal first-stage decision rule from `learnIQ1()` to recommend the best stage 1 treatment for a patient presenting with history `h1`. It is **essential** that `h1main` include the **same variables and ordering** as `H1Main`. If a formula was used to fit `learnIQ1main()`, we recommend checking `summary(<learnIQ1main object>)` for the correct order of `h2`. Similarly for `h1cm` and `h1sig`. `dens` can be chosen by looking at a normal QQ-plot of the standardized residuals from the contrast mean and variance modeling steps.

Value

`q1Pos` estimated value of the first-stage Q-function when $H1$=$h1$ and $A1$=1

`q1Neg` estimated value of the first-stage Q-function when $H1$=$h1$ and $A1$=-1

`q1opt` estimated optimal first-stage treatment for a patient presenting with $h1$

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

learnIQ1main, learnIQ1cm, learnIQ1var,

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1==”MR”)] = 1
bmiData$A1[which (bmiData$A1==”CD”)] = -1
bmiData$A2[which (bmiData$A2==”MR”)] = 1
bmiData$A2[which (bmiData$A2==”CD”)] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
## second-stage regression
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, ”A2”, c(”parent_BMI”,
  ”month4_BMI”))
summary (fitIQ2)
## model conditional expected value of main effect term
fitIQ1main = learnIQ1main (~ gender + race + parent_BMI + baseline_BMI
  + A1*(gender + parent_BMI), data=bmiData, ”A1”, c (”gender”,
                                ”parent_BMI”), fitIQ2)
## model conditional mean of contrast function
fitIQ1cm = learnIQ1cm (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI + baseline_BMI), data=bmiData, ”A1”, c
  (”gender”, ”parent_BMI”, ”baseline_BMI”), fitIQ2)
## variance modeling
fitIQ1var = learnIQ1var (~ gender + race + parent_BMI + baseline_BMI +
  A1*(parent_BMI), data=bmiData, ”A1”, c (”parent_BMI”), ”hetero”,
  fitIQ1cm)
## new patient
h1 = c (1, 1, 30, 35)
optIQ1 = IQ1 (fitIQ1main, fitIQ1cm, fitIQ1var, ”nonpar”, h1, h1, h1)
optIQ1$q1opt
```

`IQ2` *IQ-learning: Recommend stage 2 treatment*

Description

Recommends the estimated optimal second-stage treatment for a given stage 2 history, $h2$.

Usage

```
IQ2(object, h2)
```

Arguments

`object`	object of type `learnIQ2`
`h2`	vector of observed second-stage main effects corresponding to the variables in `H2` used in `learnIQ2()`

Details

Use the estimated optimal second-stage decision rule from `learnIQ2()` to recommend the best stage 2 treatment for a patient presenting with history `h2`. It is **essential** that `h2` include the **same variables and ordering** as `H2`. If a formula was used to fit `learnIQ2()`, we recommend checking `summary(<learnIQ2 object>)` for the correct order of `h2`.

Value

`q2Pos`	estimated value of the second-stage Q-function when $H2$=$h2$ and $A2$=1
`q2Neg`	estimated value of the second-stage Q-function when $H2$=$h2$ and $A2$=-1
`q2opt`	estimated optimal second-stage treatment for a patient presenting with $h2$

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`learnIQ2`, `summary.learnIQ2`, `plot.learnIQ2`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1==”MR”)] = 1
bmiData$A1[which (bmiData$A1==”CD”)] = -1
bmiData$A2[which (bmiData$A2==”MR”)] = 1
bmiData$A2[which (bmiData$A2==”CD”)] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, ”A2”, c(”parent_BMI”,
  ”month4_BMI”))
summary (fitIQ2)
## new patient
h2 = c (1, 30, 45)
optIQ2 = IQ2 (fitIQ2, h2)
optIQ2$q2opt
```

iqResids *IQ-learning: standardized residuals*

Description

Creates an object containing the standardized residuals from the contrast mean and variance modeling steps.

Usage

```
iqResids(object)
```

Arguments

`object`	object of type `learnIQ1var`

Details

Creates an object containing the standardized residuals from the contrast mean and variance modeling steps to be used with the plotting function `plot.iqResids`. The choice of density estimator in the next step of IQ-learning should be based on a QQ-plot of the standardized residuals.

Value

Returns `object$stdResids` from an object of type `learnIQ1var` in the form of an object of type `iqResids`.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Laber, E.B., Linn, K.A., and Stefanski, L.A. (2013). Interactive Q-learning. *Submitted*.

See Also

`learnIQ1var`, `plot.iqResids`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
## second-stage regression
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
                                  "month4_BMI"))
## model conditional mean of contrast function
fitIQ1cm = learnIQ1cm (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI + baseline_BMI), data=bmiData, "A1", c
  ("gender", "parent_BMI", "baseline_BMI"), fitIQ2)
## variance modeling
fitIQ1var = learnIQ1var (~ gender + race + parent_BMI + baseline_BMI +
  A1*(parent_BMI), data=bmiData, "A1", c ("parent_BMI"), "hetero",
  fitIQ1cm)
## plot standardized residuals
fitResids = iqResids (fitIQ1var)
plot (fitResids)
```

learnIQ1 *IQ-learning: estimate optimal first-stage rule*

Description

Estimates the optimal first-stage decision rule using IQ-learning.

Usage

```
learnIQ1(mainObj, cmObj, sigObj, dens)
```

Arguments

`mainObj`	object of type `learnIQ1main`
`cmObj`	object of type `learnIQ1cm`
`sigObj`	object of type `learnIQ1var`
`dens`	either "norm" or "nonpar"; density estimator to use for the conditional density of the contrast function

Details

If `dens="norm"` the normal location-scale density estimator is used, otherwise when `dens="nonpar"` the empirical density estimator is used.

Value

`optA1`	vector of estimated optimal first-stage treatment for the patients in the training data

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`learnIQ1main`, `learnIQ1cm`, `learnIQ1var`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1==”MR”)] = 1
bmiData$A1[which (bmiData$A1==”CD”)] = -1
bmiData$A2[which (bmiData$A2==”MR”)] = 1
bmiData$A2[which (bmiData$A2==”CD”)] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
s2ints = c (2, 3)
## second-stage regression
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, ”A2”, c(”parent_BMI”,
                                  ”month4_BMI”))
## model conditional expected value of main effect term
fitIQ1main = learnIQ1main (~ gender + race + parent_BMI + baseline_BMI
  + A1*(gender + parent_BMI), data=bmiData, ”A1”, c (”gender”,
                                ”parent_BMI”), fitIQ2)
## model conditional mean of contrast function
fitIQ1cm = learnIQ1cm (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI + baseline_BMI), data=bmiData, ”A1”, c
  (”gender”, ”parent_BMI”, ”baseline_BMI”), fitIQ2)
## variance modeling
fitIQ1var = learnIQ1var (~ gender + race + parent_BMI + baseline_BMI +
  A1*(parent_BMI), data=bmiData, ”A1”, c (”parent_BMI”), ”hetero”,
  fitIQ1cm)
## get optimal first-stage txts
fitIQLearn = learnIQ1 (fitIQ1main, fitIQ1cm, fitIQ1var, ”nonpar”)
```

`learnIQ1cm` *IQ-learning: contrast function mean regression*

Description

Estimates the mean of the contrast function by fitting a linear regression of the estimated contrast function term on first-stage history and treatment.

Usage

```
learnIQ1cm(object, ...)

## S3 method for class 'formula'
```

```
learnIQ1cm(formula, data, treatName, intNames, s2object, ...)
## Default S3 method:
learnIQ1cm(object, H1CMean, A1, s1cmInts, ...)
```

Arguments

`formula` formula for the contrast function mean regression

`data` data frame containing variables used in `formula`

`treatName` character string indicating the stage 1 treatment name

`intNames` vector of characters indicating the names of the variables that interact with the stage 1 treatment in the contrast function mean regression model

`s2object` object of type `learnIQ2`

`object` object of type `learnIQ2`

`H1CMean` matrix or data frame of first-stage covariates to include as main effects in the linear model

`A1` vector of first-stage randomized treatments

`s1cmInts` indices pointing to columns of H1CMean that should be included as treatment interaction effects in the linear model

`...` other arguments to be passed to `lm()`

Details

Fits a model of the form

$$E(H_{21}^T\beta_{21}|H_1, A_1) = H_{10}^T\beta_{10} + A_1H_{11}^T\beta_{11},$$

where $H10$ and $H11$ are summaries of $H1$. Though a slight abuse of notation, these summaries are not required to be the same as $H10$ and $H11$ in the main effect term regression or the variance model. For an object of type `learnIQ1cm`, `summary(object)` and `plot(object)` can be used for evaluating model fit.

Value

`betaHat10` estimated main effect coefficients; first is the intercept

`betaHat11` estimated treatment interaction coefficients; first is the main effect of the first-stage treatment

`s1cmFit` `lm()` object of the contrast mean regression fit

`cmeanResids` residuals from the regression

`cmPos` vector of predicted values with $A1$=1 for all patients

`cmNeg` vector of predicted values with $A1$=1 for all patients

`s1cmInts` indicies of variables in H1CMean included as treatment interactions in the model; same as input `s1cmInts`

`A1` vector of first-stage randomized treatments; same as input `A1`

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

learnIQ2, summary.learnIQ1cm, plot.learnIQ1cm

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
s2ints = c (2, 3)
## second-stage regression
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
                                "month4_BMI"))
fitIQ1cm = learnIQ1cm (~ gender + race + parent_BMI + baseline_BMI +
 A1*(gender + parent_BMI + baseline_BMI), data=bmiData, "A1",
 c ("gender", "parent_BMI", "baseline_BMI"), fitIQ2)
summary (fitIQ1cm)
plot (fitIQ1cm)
```

---

learnIQ1main *IQ-learning: main effect term regression*

Description

Fits a linear regression of the estimated main effect term on first-stage history and treatment.

Usage

```
learnIQ1main(object, ...)

## S3 method for class 'formula'
learnIQ1main(formula, data, treatName, intNames, s2object, ...)
## Default S3 method:
learnIQ1main(object, H1Main, A1, s1mainInts, ...)
```

Arguments

`formula` formula for the main effect term regression

`data` data frame containing variables used in `formula`

`treatName` character string indicating the stage 1 treatment name

`intNames` vector of characters indicating the names of the variables that interact with the stage 1 treatment in the main effect term regression model

`s2object` object of type `learnIQ2`

`object` object of type `learnIQ2`

`H1Main` matrix or data frame of first-stage covariates to include as main effects in the linear model

`A1` vector of first-stage randomized treatments

`s1mainInts` indices pointing to columns of H1Main that should be included as treatment interaction effects in the linear model

`...` other arguments to be passed to `lm()`

Details

Fits a model of the form

$$E(H_{20}^T\beta_{20}|H_1, A_1) = H_{10}^T\alpha_0 + A_1H_{11}^T\alpha_1,$$

where $H10$ and $H11$ are summaries of $H1$. For an object of type `learnIQ1main`, `summary(object)` and `plot(object)` can be used for evaluating model fit.

Value

`alphaHat0` estimated main effect coefficients; first is the intercept

`alphaHat1` estimated treatment interaction coefficients; first is the main effect of the first-stage treatment

`s1MainFit` `lm()` object of the main effect term regression fit

`mainPos` vector of predicted values with $A1$=1 for all patients

`mainNeg` vector of predicted values with $A1$=-1 for all patients

`s1mainInts` indicies of variables in H1Main included as treatment interactions in the model; same as input `s1mainInts`

`A1` vector of first-stage randomized treatments; same as input `A1`

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

learnIQ2, summary.learnIQ1main, plot.learnIQ1main

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
s2ints = c (2, 3)
## second-stage regression
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
                                "month4_BMI"))
## model conditional expected value of main effect term
fitIQ1main = learnIQ1main (fitIQ2, s1vars, a1, c (1, 3))
fitIQ1main = learnIQ1main (~ gender + race + parent_BMI + baseline_BMI
  + A1*(gender + parent_BMI), data=bmiData, "A1", c ("gender",
                              "parent_BMI"), fitIQ2)
summary (fitIQ1main)
plot (fitIQ1main)
```

learnIQ1var *IQ-learning: contrast variance modeling*

Description

Estimates the variance function of the contrast function by fitting a constant variance function or a log linear model to the residuals of the contrast mean fit.

Usage

```
learnIQ1var(object, ...)

## S3 method for class 'formula'
learnIQ1var(formula, data, treatName, intNames,
method, cmObject, ...)
## Default S3 method:
learnIQ1var(object, H1CVar, s1sInts, method, ...)
```

Arguments

`formula` right-hand side formula containing the linear model to be used for the log-transformed, squared residuals from the contrast function mean fit

`data` data frame containing variables used in `formula`

`treatName` character string indicating the stage 1 treatment name

`intNames` vector of characters indicating the names of the variables that interact with the stage 1 treatment in the contrast function variance model

`method` either "homo" for a constant variance function or "hetero" for a log-linear variance function; default method is "homo"

`cmObject` object of type `learnIQ1cm`

`object` object of type `learnIQ1cm`

`H1CVar` matrix or data frame of first-stage covariates to include as main effects in the log-linear model; default is `NULL` for a constant variance fit

`s1sInts` indices pointing to columns of H1CVar that should be included as treatment interaction effects in the log-linear model; default is `NULL`

`...` additional arguments to be passed to `lm()` when fitting the hetero log-linear model

Details

If `method="homo"`, computes the variance of the residuals from the contrast function mean fit. If `method="hetero"`, fits a model of the form

$$E(\log e^2|H_1, A_1) = H_{10}^T\gamma_0 + A_1 H_{11}^T\gamma_1,$$

where $H10$ and $H11$ are summaries of $H1$. Though a slight abuse of notation, these summaries are not required to be the same as $H10$ and $H11$ in the main effect term regression or the contrast mean fit. Also, e^2 = $H21^T\beta21$ - $E(H21^T\beta21|H1, A1)$. For an object of type `learnIQ1var`, `summary(object)` and `plot(object)` can be used for evaluating model fit.

Value

`stdDev` standard deviation of the residuals from the contrast function mean fit when `method="homo"`, otherwise `NULL`

`stdResids` standardized residuals of the contrast function after mean and variance modeling, using either `method="homo"` or `"hetero"`

`gammaHat0` estiamted regression coefficients from the log-linear model main effects when `method="hetero"`, otherwise `NULL`

`gammaHat1` estimated regression coefficients from the log-linear model interaction effects when `method="hetero"`, otherwise `NULL`

`s1VarFit` `lm()` object from the log-linear model when `method="hetero"`, otherwise `NULL`

`homo` logical variable indicating if `method="homo"` was used

`sigPos` vector of predicted values when $A1$=1 for all patients

`sigNeg` vector of predicted values when $A1$=-1 for all patients

`s1sInts` indices of variables in `H1CVar` included as treatment interactions in the model; same as input `s1sInts`

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`learnIQ1cm`, `iqResids`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
```

```
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
## second-stage regression
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
                                "month4_BMI"))
## model conditional expected value of main effect term
fitIQ1main = learnIQ1main (~ gender + race + parent_BMI + baseline_BMI
  + A1*(gender + parent_BMI), data=bmiData, "A1", c ("gender",
                              "parent_BMI"), fitIQ2)
## model conditional mean of contrast function
fitIQ1cm = learnIQ1cm (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI + baseline_BMI), data=bmiData, "A1", c
  ("gender", "parent_BMI", "baseline_BMI"), fitIQ2)
## variance modeling
fitIQ1var = learnIQ1var (fitIQ1cm) ## constant variance fit
fitIQ1var = learnIQ1var (fitIQ1cm, s1vars, c (3, 4), method="hetero")
## non-constant variance fit
fitIQ1var = learnIQ1var (~ gender + race + parent_BMI + baseline_BMI +
  A1*(parent_BMI), data=bmiData, "A1", c ("parent_BMI"),
  "hetero", fitIQ1cm)
## non-constant variance fit using formula specification
```

`learnIQ2` *IQ-learning: second-stage regression*

Description

Fits a linear regression of the response on second-stage history and treatment to estimate the optimal second-stage decision rule.

Usage

```
learnIQ2(H2, ...)

## S3 method for class 'formula'
learnIQ2(formula, data, treatName, intNames, ...)
## Default S3 method:
learnIQ2(H2, Y, A2, s2ints, ...)
```

Arguments

`formula`	stage 2 regression formula
`data`	data frame containing variables used in `formula`
`treatName`	character string indicating the stage 2 treatment name
`intNames`	vector of characters indicating the names of the variables that interact with the stage 2 treatment in the regression model
`H2`	matrix or data frame of second-stage covariates to include as main effects in the linear model

`Y` response vector

`A2` vector of second-stage randomized treatments

`s2ints` indices pointing to columns of H2 that should be included as treatment interaction effects in the linear model

`...` other arguments to be passed to `lm()`

Details

Fits a model of the form

$$E(Y|H_2, A_2) = H_{20}^T \beta_{20} + A_2 H_{21}^T \beta_{21},$$

where $H20$ and $H21$ are summaries of $H2$. For an object of type `learnIQ2`, `summary(object)` and `plot(object)` can be used for evaluating model fit.

Value

`betaHat20` estimated main effect coefficients; first is the intercept

`betaHat21` estimated treatment interaction coefficients; first is the main effect of the second-stage treatment

`s2Fit` `lm()` object of the second-stage regression fit

`optA2` vector of estimated optimal second-stage treatments for the patients in the training data

`main` estimated main effect vector, $H20^T \hat{\beta 20}$

`contrast` estimated contrast function vector, $H21^T \hat{\beta 21}$

`s2ints` indicies of variables in H2 included as treatment interactions in the model; same as input `s2ints`

`A2` vector of second-stage randomized treatments; same as input `A2`

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`summary.learnIQ2`, `plot.learnIQ2`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
s2ints = c (2, 3)
## second-stage regression
fitIQ2 = learnIQ2 (s2vars, y, a2, s2ints)
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
                                 "month4_BMI"))

summary (fitIQ2)
plot (fitIQ2)
```

plot.iqResids *Plot the standardized residuals*

Description

Plot the standardized residuals that arise from the contrast function mean and variance modeling.

Usage

```
## S3 method for class 'iqResids'
plot(x, ...)
```

Arguments

x	object of type iqResids
...	additional arguments to be passed to plot()

Details

Can be used to decide which density estimator ("norm" or "nonpar") should be used for the conditional density of the contrast function given first-stage history and treatment.

Value

Returns a normal QQ-plot of the standardized residuals.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

iqResids

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
s2ints = c (2, 3)
## second-stage regression
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
                                  "month4_BMI"))
## model conditional mean of contrast function
fitIQ1cm = learnIQ1cm (~ gender + race + parent_BMI + baseline_BMI +
 A1*(gender + parent_BMI + baseline_BMI), data=bmiData, "A1",
 c ("gender", "parent_BMI", "baseline_BMI"), fitIQ2)
## variance modeling
fitIQ1var = learnIQ1var (~ gender + race + parent_BMI + baseline_BMI +
  A1*(parent_BMI), data=bmiData, "A1", c ("parent_BMI"), "hetero",
  fitIQ1cm)
## plot standardized residuals
fitResids = iqResids (fitIQ1var)
plot (fitResids)
```

plot.learnIQ1cm *Residual plots for the contrast mean model*

Description

Displays common residual plots that can be used to diagnose model fit for the contrast function mean model.

Usage

```
## S3 method for class 'learnIQ1cm'
plot(x, ...)
```

Arguments

x	object of type `learnIQ1cm`
...	additional arguments to be passed to `plot()`

Value

None.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`learnIQ1cm`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
```

```
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
s2ints = c (2, 3)
## second-stage regression
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, ”A2”, c(”parent_BMI”,
                                   ”month4_BMI”))
## model conditional mean of contrast function
fitIQ1cm = learnIQ1cm (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI + baseline_BMI), data=bmiData, ”A1”, c
  (”gender”, ”parent_BMI”, ”baseline_BMI”), fitIQ2)
plot (fitIQ1cm)
```

`plot.learnIQ1main` *Residual plots for the main effect term model*

Description

Displays common residual plots that can be used to diagnose model fit for the main effect term regression.

Usage

```
## S3 method for class 'learnIQ1main'
plot(x, ...)
```

Arguments

`x`	object of type `learnIQ1main`
`...`	additional arguments to be passed to `plot()`

Value

None.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

learnIQ1main

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
s2ints = c (2, 3)
## second-stage regression
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
                                "month4_BMI"))
## model conditional expected value of main effect term
fitIQ1main = learnIQ1main (fitIQ2, s1vars, a1, c (1, 3))
fitIQ1main = learnIQ1main (~ gender + race + parent_BMI + baseline_BMI
  + A1*(gender + parent_BMI), data=bmiData, "A1", c ("gender",
                              "parent_BMI"), fitIQ2)
plot (fitIQ1main)
```

plot.learnIQ1var   *Residual plots for the contrast variance model*

Description

Displays common residual plots that can be used to diagnose model fit for the contrast function variance model when `method="hetero"`.

Usage

```
## S3 method for class 'learnIQ1var'
plot(x, ...)
```

Arguments

x	object of type `learnIQ1var`
...	additional arguments to be passed to `plot()`

Details

Will only plot residuals when `method="hetero"` was used to create the object of type `learnIQ1var`.

Value

None.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`learnIQ1var`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
  "month4_BMI"))
fitIQ1cm = learnIQ1cm (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI + baseline_BMI), data=bmiData, "A1", c
  ("gender", "parent_BMI", "baseline_BMI"), fitIQ2)
fitIQ1var = learnIQ1var (~ gender + race + parent_BMI + baseline_BMI +
  A1*(parent_BMI), data=bmiData, "A1", c ("parent_BMI"), "hetero",
  fitIQ1cm)
plot (fitIQ1var)
```

plot.learnIQ2 *Residual plots for the second-stage regression*

Description

Displays common residual plots that can be used to diagnose model fit for the second-stage regression model.

Usage

```
## S3 method for class 'learnIQ2'
plot(x, ...)
```

Arguments

x	object of type `learnIQ2`
...	additional arguments to be passed to `plot()`

Value

None.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`learnIQ2.default`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
```

```
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
## second-stage regression
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
                                  "month4_BMI"))
plot (fitIQ2)
```

---

plot.qLearnS1 *Residual plots for the first-stage regression*

Description

Displays common residual plots based on the model for the first-stage regression in Q-learning. Due to the response being a non-smooth, non-monotone transformation of the data, these plots may not be meaningful.

Usage

```
## S3 method for class 'qLearnS1'
plot(x, ...)
```

Arguments

x	object of type qLearnS1
...	additional arguments to be passed to plot()

Value

None.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

qLearnS1

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1==”MR”)] = 1
bmiData$A1[which (bmiData$A1==”CD”)] = -1
bmiData$A2[which (bmiData$A2==”MR”)] = 1
bmiData$A2[which (bmiData$A2==”CD”)] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
## second-stage regression
fitQ2 = qLearnS2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, ”A2”, c(”parent_BMI”,
                                 ”month4_BMI”))
## first-stage regression
fitQ1 = qLearnS1 (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI), data=bmiData, ”A1”, c (”gender”,
                             ”parent_BMI”), fitQ2)
plot (fitQ1)
```

plot.qLearnS2 *Residual plots for the second-stage regression*

Description

Displays common residual plots that can be used to diagnose model fit for the second-stage regression model.

Usage

```
## S3 method for class 'qLearnS2'
plot(x, ...)
```

Arguments

x	object of type qLearnS2
...	other arguments to be passed to plot()

Value

None.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

qLearnS2

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1==”MR”)] = 1
bmiData$A1[which (bmiData$A1==”CD”)] = -1
bmiData$A2[which (bmiData$A2==”MR”)] = 1
bmiData$A2[which (bmiData$A2==”CD”)] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
## second-stage regression
fitQ2 = qLearnS2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, ”A2”, c(”parent_BMI”,
                                  ”month4_BMI”))
plot (fitQ2)
```

qLearnQ1 *Q-learning: Recommend stage 1 treatment*

Description

Recommends the Q-learning estimated optimal first-stage treatment for a given stage 1 history, $h1$.

Usage

```
qLearnQ1(object, h1q)
```

Arguments

object	object of type qLearnS1
h1q	vector of observed first-stage main effects corresponding to the variables in H1q used in qLearnS1()

Details

Use the estimated optimal first-stage decision rule from qLearnS1() to recommend the best stage 1 treatment for a patient presenting with history h1q. It is **essential** that h1q include the **same variables and ordering** as H1q. If a formula was used to fit qLearnS1(), we recommend checking summary(<qLearnS1 object>) for the correct order of h1q.

Value

q1Pos	estimated value of the first-stage Q-function when $H1$=$h1$ and $A1$=1
q1Neg	estimated value of the first-stage Q-function when $H1$=$h1$ and $A1$=-1
q1opt	estimated optimal first-stage treatment for a patient presenting with $h1$

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

qLearnS1, summary.qLearnS1, plot.qLearnS1

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
```

```
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
## second-stage regression
fitQ2 = qLearnS2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
  "month4_BMI"))
## first-stage regression
fitQ1 = qLearnS1 (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI), data=bmiData, "A1", c ("gender",
                              "parent_BMI"), fitQ2)
summary (fitQ1)

h1q = c (1, 1, 35, 45);
optQ1 = qLearnQ1 (fitQ1, h1q);
optQ1$q1opt
```

qLearnQ2 *Q-learning: Recommend stage 2 treatment*

Description

Recommends the estimated optimal second-stage treatment for a given stage 2 history, $h2$. This is the same as `IQ2`.

Usage

```
qLearnQ2(object, h2)
```

Arguments

`object`	object of type `qLearnS2`
`h2`	vector of observed second-stage main effects corresponding to the variables in `H2` used in `qLearnS2()`

Details

Use the estimated optimal second-stage decision rule from `qLearnS2()` to recommend the best stage 2 treatment for a patient presenting with history `h2`. It is **essential** that `h2` include the **same variables and ordering** as `H2`. If a formula was used to fit `qLearnS2()`, we recommend checking `summary(qLearnS2)` for the correct order of `h2`.

Value

`q2Pos`	estimated value of the second-stage Q-function when $H2$=$h2$ and $A2$=1
`q2Neg`	estimated value of the second-stage Q-function when $H2$=$h2$ and $A2$=-1
`q2opt`	estimated optimal second-stage treatment for a patient presenting with $h2$

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

qLearnS2, summary.qLearnS2, plot.qLearnS2

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1==”MR”)] = 1
bmiData$A1[which (bmiData$A1==”CD”)] = -1
bmiData$A2[which (bmiData$A2==”MR”)] = 1
bmiData$A2[which (bmiData$A2==”CD”)] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
fitQ2 = qLearnS2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, ”A2”, c(”parent_BMI”,
  ”month4_BMI”))
summary (fitQ2)

h2 = c (1, 30, 45)
optQ2 = qLearnQ2 (fitQ2, h2)
optQ2$q2opt
```

qLearnS1 *Q-learning: first-stage regression*

Description

Regresses the predicted future outcome maximized over $a2$ on first-stage history and treatment to estimate the optimal first-stage decision rule using Q-learning.

Usage

```
qLearnS1(object, ...)

## S3 method for class 'formula'
qLearnS1(formula, data, treatName, intNames,
qS2object, ...)
## Default S3 method:
qLearnS1(object, H1q, A1, s1ints, ...)
```

Arguments

`formula`	right-hand sided stage 1 regression formula
`data`	data frame containing variables used in `formula`
`treatName`	character string indicating the stage 1 treatment name
`intNames`	vector of characters indicating the names of the variables that interact with the stage 1 treatment in the regression model
`qS2object`	object of type `qLearnS2`
`object`	object of type `qLearnS2`
`H1q`	matrix or data frame of first-stage covariates to include as main effects in the linear model
`A1`	vector of first-stage randomized treatments
`s1ints`	indices pointing to columns of H1q that should be included as treatment interaction effects in the linear model
`...`	other arguments to be passed to `lm()`

Details

Fits a model of the form

$$E(\tilde{Y}|H_1, A_1) = H_{10}^T\beta_{10} + A_1H_{11}^T\beta_{11},$$

where $H10$ and $H11$ are summaries of $H1$. For an object of type `qLearnS1`, `summary(object)` and `plot(object)` can be used for evaluating model fit.

Value

`betaHat10`	estimated main effect coeffients, beginning with the intercept
`betaHat11`	estimated treatment interaction coefficients, beginning with the main effect of treatment
`optA1`	vector of estimated optimal first-stage treatments for the patients in the training data
`s1Fit`	`lm()` object of the first-stage regression fit
`s1ints`	indices of variables in `H1q` included as treatment interactions in the model; same as input `s1ints`

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

summary.qLearnS2, plot.qLearnS2

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
s2ints = c (2, 3)
## second-stage regression
fitQ2 = qLearnS2 (s2vars, y, a2, s2ints)
fitQ2 = qLearnS2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
                                 "month4_BMI"))
## first-stage regression
fitQ1 = qLearnS1 (fitQ2, s1vars, a1, c(3,4))
fitQ1 = qLearnS1 (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI), data=bmiData, "A1", c ("gender",
                             "parent_BMI"), fitQ2)
```

qLearnS2 *Q-learning: second-stage regression*

Description

Same as `learnIQ2`. Estimates the optimal second-stage decision rule using a linear regression of the response on second-stage history and treatment.

Usage

```
qLearnS2(H2, ...)

## S3 method for class 'formula'
qLearnS2(formula, data, treatName, intNames, ...)
## Default S3 method:
qLearnS2(H2, Y, A2, s2ints, ...)
```

Arguments

`formula`	stage 2 regression formula
`data`	data frame containing variables used in `formula`
`treatName`	character string indicating the stage 2 treatment name
`intNames`	vector of characters indicating the names of the variables that interact with the stage 2 treatment in the regression model
`H2`	matrix or data frame of second-stage covariates to include as main effects in the linear model
`Y`	response vector
`A2`	vector of second-stage randomized treatments
`s2ints`	indices pointing to columns of `H2` that should be included as treatment interaction effects in the linear model
`...`	other arguments to be passed to `lm()`

Details

Fits a model of the form

$$E(Y|H_2, A_2) = H_{20}^T \beta_{20} + A_2 H_{21}^T \beta_{21},$$

where $H20$ and $H21$ are summaries of $H2$. For an object of type `qLearnS2`, `summary(object)` and `plot(object)` can be used for evaluating model fit.

Value

`betaHat20`	estimated main effect coefficients; first is the intercept
`betaHat21`	estimated treatment interaction coefficients; first is the main effect of the second-stage treatment
`Ytilde`	Q2 function maximized over treatment $a2$; the predicted future outcome assuming optimal treatment is given at the second stage to be used in the next step of the Q-learning algorithm
`optA2`	vector of estimated optimal second-stage treatments for the patients in the training data
`s2Fit`	`lm()` object of the second-stage regression fit
`s2ints`	indicies of variables in `H2` included as treatment interactions in the model; same as input `s2ints`

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`summary.qLearnS2`, `plot.qLearnS2`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
s2ints = c (2, 3)
## second-stage regression
fitQ2 = qLearnS2 (s2vars, y, a2, s2ints)
fitQ2 = qLearnS2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
                                  "month4_BMI"))

summary (fitQ2)
plot (fitQ2)
```

summary.learnIQ1cm *IQ-learning: contrast mean regression summary*

Description

Output from the contrast function mean regression in IQ-learning.

Usage

```
## S3 method for class 'learnIQ1cm'
summary(object, ...)
```

Arguments

`object`	object of type `learnIQ1cm`
`...`	additional arguments to be passed to `summary()`

Details

Regression output and other summary statistics from the contrast function mean regression. See `summary.lm` for more details.

Value

Computes and returns multiple summary statistics from the linear model in `object`. See `summary.lm` for a list of available summary statistics.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`learnIQ1cm`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1==”MR”)] = 1
bmiData$A1[which (bmiData$A1==”CD”)] = -1
bmiData$A2[which (bmiData$A2==”MR”)] = 1
bmiData$A2[which (bmiData$A2==”CD”)] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
```

```
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
  "month4_BMI"))
fitIQ1cm = learnIQ1cm (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI + baseline_BMI), data=bmiData, "A1", c
  ("gender", "parent_BMI", "baseline_BMI"), fitIQ2)

summary (fitIQ1cm)
```

summary.learnIQ1main *IQ-learning: main effect regression summary*

Description

Output from the main effect term regression in IQ-learning.

Usage

```
## S3 method for class 'learnIQ1main'
summary(object, ...)
```

Arguments

`object`	object of type `learnIQ1main`
`...`	additional arguments to be passed to `summary()`

Details

Regression output and other summary statistics from the main effect term regression. See `summary.lm` for more details.

Value

Computes and returns multiple summary statistics from the linear model in `object`. See `summary.lm` for a list of available summary statistics.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

learnIQ1main

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1==”MR”)] = 1
bmiData$A1[which (bmiData$A1==”CD”)] = -1
bmiData$A2[which (bmiData$A2==”MR”)] = 1
bmiData$A2[which (bmiData$A2==”CD”)] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, ”A2”, c(”parent_BMI”,
  ”month4_BMI”))
fitIQ1main = learnIQ1main (~ gender + race + parent_BMI + baseline_BMI
  + A1*(gender + parent_BMI), data=bmiData, ”A1”, c (”gender”,
                                ”parent_BMI”), fitIQ2)
summary (fitIQ1main)
```

summary.learnIQ1var *IQ-learning: contrast variance modeling summary*

Description

Output from the contrast function variance modeling in IQ-learning.

Usage

```
## S3 method for class 'learnIQ1var'
summary(object, ...)
```

Arguments

object	object of type learnIQ1var
...	additional arguments to be passed to summary()

Details

When method="homo" returns the standard deviation from the constant fit. When method="hetero" returns regression output and other summary statistics from the contrast function log-linear variance model. See summary.lm for more details.

Value

When method="hetero" computes and returns multiple summary statistics from the log-linear model in object. See summary.lm for a list of available summary statistics. Otherwise, when method="homo" returns only the standard deviation from the constant variance fit.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

learnIQ1var

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
  "month4_BMI"))
fitIQ1cm = learnIQ1cm (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI + baseline_BMI), data=bmiData, "A1", c
  ("gender", "parent_BMI", "baseline_BMI"), fitIQ2)
fitIQ1var = learnIQ1var (fitIQ1cm)
summary (fitIQ1var)

fitIQ1var = learnIQ1var (~ gender + race + parent_BMI + baseline_BMI +
  A1*(parent_BMI), data=bmiData, "A1", c ("parent_BMI"), "hetero",
  fitIQ1cm)
summary (fitIQ1var)
```

`summary.learnIQ2` *IQ-learning: second-stage regression summary*

Description

Output from the second-stage regression in IQ-learning.

Usage

```
## S3 method for class 'learnIQ2'
summary(object, ...)
```

Arguments

`object`	object of type `learnIQ2`
`...`	additional arguments to be passed to `summary()`

Details

Regression output and other summary statistics from the second-stage regression. See `summary.lm` for more details.

Value

Computes and returns multiple summary statistics from the linear model in `object`. See `summary.lm` for a list of available summary statistics.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`learnIQ2`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1==”MR”)] = 1
bmiData$A1[which (bmiData$A1==”CD”)] = -1
bmiData$A2[which (bmiData$A2==”MR”)] = 1
bmiData$A2[which (bmiData$A2==”CD”)] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, ”A2”, c(”parent_BMI”,
  ”month4_BMI”))
summary (fitIQ2)
```

summary.qLearnS1 *Q-learning: first-stage regression summary*

Description

Output from the first-stage regression in Q-learning.

Usage

```
## S3 method for class 'qLearnS1'
summary(object, ...)
```

Arguments

`object`	object of type `qLearnS1`
`...`	additional arguments to be passed to `summary()`

Details

Regression output and other summary statistics from the second-stage regression. See `summary.lm` for more details.

Value

Computes and returns multiple summary statistics from the linear model in `object`. See `summary.lm` for a list of available summary statistics.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`qLearnS1`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
## second-stage regression
fitQ2 = qLearnS2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
  "month4_BMI"))
## first-stage regression
fitQ1 = qLearnS1 (fitQ2, s1vars, a1, c(3,4))
fitQ1 = qLearnS1 (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI), data=bmiData, "A1", c ("gender",
                              "parent_BMI"), fitQ2)
summary (fitQ1)
```

`summary.qLearnS2` *Q-learning: second-stage regression summary*

Description

Output from the second-stage regression in Q-learning.

Usage

```
## S3 method for class 'qLearnS2'
summary(object, ...)
```

Arguments

`object`	object of type `qLearnS2`
`...`	additional arguments to be passed to `summary()`

Details

Regression output and other summary statistics from the second-stage regression. See `summary.lm` for more details.

Value

Computes and returns multiple summary statistics from the linear model in `object`. See `summary.lm` for a list of available summary statistics.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`qLearnS2`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1==”MR”)] = 1
bmiData$A1[which (bmiData$A1==”CD”)] = -1
bmiData$A2[which (bmiData$A2==”MR”)] = 1
bmiData$A2[which (bmiData$A2==”CD”)] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
```

```
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
fitQ2 = qLearnS2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
  "month4_BMI"))
summary (fitQ2)
```

summary.value *Estimate plug-in value*

Description

Output plug-in value estimates of treatment regimes based on the estimated IQ-learning parameters.

Usage

```
## S3 method for class 'value'
summary(object, ...)
```

Arguments

object	object of type value
...	other arguments to be passed to summary()

Details

Can be used to print out the estimated plug-in values of non-dynamic regimes and any proposed regime, possibly estimated from IQ- or Q-learning.

Value

None.

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

learnIQ2, learnIQ1

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
## second-stage regression
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
                                "month4_BMI"))
## model conditional expected value of main effect term
fitIQ1main = learnIQ1main (~ gender + race + parent_BMI + baseline_BMI
  + A1*(gender + parent_BMI), data=bmiData, "A1", c ("gender",
                              "parent_BMI"), fitIQ2)
## model conditional mean of contrast function
fitIQ1cm = learnIQ1cm (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI + baseline_BMI), data=bmiData, "A1", c
  ("gender", "parent_BMI", "baseline_BMI"), fitIQ2)
## variance modeling
fitIQ1var = learnIQ1var (~ gender + race + parent_BMI + baseline_BMI +
  A1*(parent_BMI), data=bmiData, "A1", c ("parent_BMI"), "hetero",
  fitIQ1cm)
## get optimal first-stage txts
fitIQLearn = learnIQ1 (fitIQ1main, fitIQ1cm, fitIQ1var, "nonpar")
estVal = value (fitIQLearn$optA1, fitIQ2$optA2, y, a1, a2)
summary (estVal)
```

`value` *Estimate plug-in value*

Description

Estimate the plug-in value of the any fixed treatment regime.

Usage

```
value(d1, d2, Y, A1, A2)
```

Arguments

`d1` vector of first-stage treatments corresponding to the first-stage decision rule of the proposed regime

`d2` vector of second-stage treatments corresponding to the second-stage decision rule of the proposed regime

`Y` vector of responses

`A1` vector of first-stage randomized treatments

`A2` vector of second-stage randomized treatments

Details

The formula for the plug-in value estimate is

$$\frac{\sum_i Y_i * ind1_i * ind1_i}{\sum_i ind1_i * ind2_i}$$

where $ind1$ and $ind2$ are indicators that the first- and second-stage randomized treatments were consistent with the regime of interest.

Value

`value` estimated plug-in value of the regime

`valPosPos` estimated plug-in value of the regime that treats all patients with $A1$=1 and $A2 = 1$

`valPosNeg` estimated plug-in value of the regime that treats all patients with $A1$=1 and $A2 = -1$

`valNegPos` estimated plug-in value of the regime that treats all patients with $A1$=-1 and $A2$=1

`valNegNeg` estimated plug-in value of the regime that treats all patients with $A1$=-1 and $A2$=-1

Author(s)

Kristin A. Linn <kalinn@ncsu.edu>, Eric B. Laber, Leonard A. Stefanski

References

Linn, K. A., Laber, E. B., Stefanski, L. A. (2015) "iqLearn: Interactive Q-Learning in R", Journal of Statistical Software, 64(1), 1–25.

Laber, E. B., Linn, K. A., and Stefanski, L. A. (2014) "Interactive model building for Q-learning", Biometrika, 101(4), 831-847.

See Also

`summary.value`

Examples

```
## load in two-stage BMI data
data (bmiData)
bmiData$A1[which (bmiData$A1=="MR")] = 1
bmiData$A1[which (bmiData$A1=="CD")] = -1
bmiData$A2[which (bmiData$A2=="MR")] = 1
bmiData$A2[which (bmiData$A2=="CD")] = -1
bmiData$A1 = as.numeric (bmiData$A1)
bmiData$A2 = as.numeric (bmiData$A2)
s1vars = bmiData[,1:4]
s2vars = bmiData[,c (1, 3, 5)]
a1 = bmiData[,7]
a2 = bmiData[,8]
## define response y to be the negative 12 month change in BMI from
## baseline
y = -(bmiData[,6] - bmiData[,4])/bmiData[,4]
## second-stage regression
fitIQ2 = learnIQ2 (y ~ gender + parent_BMI + month4_BMI +
  A2*(parent_BMI + month4_BMI), data=bmiData, "A2", c("parent_BMI",
                                  "month4_BMI"))
## model conditional expected value of main effect term
fitIQ1main = learnIQ1main (~ gender + race + parent_BMI + baseline_BMI
  + A1*(gender + parent_BMI), data=bmiData, "A1", c ("gender",
                                "parent_BMI"), fitIQ2)
## model conditional mean of contrast function
fitIQ1cm = learnIQ1cm (~ gender + race + parent_BMI + baseline_BMI +
  A1*(gender + parent_BMI + baseline_BMI), data=bmiData, "A1", c
  ("gender", "parent_BMI", "baseline_BMI"), fitIQ2)
## variance modeling
fitIQ1var = learnIQ1var (~ gender + race + parent_BMI + baseline_BMI +
  A1*(parent_BMI), data=bmiData, "A1", c ("parent_BMI"), "hetero",
  fitIQ1cm)
## get optimal first-stage txts
fitIQLearn = learnIQ1 (fitIQ1main, fitIQ1cm, fitIQ1var, "nonpar")
estVal = value (fitIQLearn$optA1, fitIQ2$optA2, y, a1, a2)
estVal
```

Index