Package 'AnchorRegression'

October 12, 2022

Type Package

Title Perform AnchorRegression

Version 0.1.3

Author Simon Zimmermann

Maintainer Simon Zimmermann <zimmersi@hu-berlin.de>

Description Performs AnchorRegression proposed by Rothenhäusler et al. 2020.
The code is adapted from the original paper repository. (<https://github.com/rothenhaeusler/anchor-regression>)
The code was developed independently from the authors of the paper.

License MIT + file LICENSE

URL https://github.com/simzim96/AnchorRegression

BugReports https://github.com/simzim96/AnchorRegression/issues

Depends R (>= 2.0.0)

Imports glmnet (>= 1.4), selectiveInference (>= 1.0.0), mgcv (>= 1.0)

Encoding UTF-8

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2021-01-06 14:40:02 UTC

R topics documented:

---

anchor_prediction   *anchor_prediction*

Description

Perform a prediction for an Anchor Regression model as described in Rothenhäusler et al.2020

Usage

```
anchor_prediction(anchor_model, x, anchor, gamma, target_variable)
```

Arguments

anchor_model	is the Anchor Regression model object
x	is a dataframe containing the matrix x containing the independent variables
anchor	is a dataframe containing the matrix anchor containing the anchor variable
gamma	is the regularization parameter for the Anchor Regression
target_variable	
	is the target variable name contained in the x dataframe

Value

A list of predictions.

Examples

```
x <- as.data.frame(matrix(data = rnorm(100),nrow = 100,ncol = 10))
anchor <- as.data.frame(matrix(data = rnorm(200),nrow = 100,ncol = 2))
colnames(anchor) <- c('X1','X2')
gamma <- 2
target_variable <- 'V2'
anchor_model <- anchor_regression(x, anchor, gamma, target_variable)
anchor_prediction(anchor_model$model, x, anchor, gamma, target_variable)
```

---

anchor_prediction_gam *anchor_prediction*

Description

Perform a prediction for an Anchor Regression model as described in Rothenhäusler et al.2020

Usage

```
anchor_prediction_gam(
  anchor_model,
  x,
  anchor,
  gamma,
  target_variable,
  bin_factor
)
```

Arguments

anchor_model	is the Anchor Regression model object
x	is a dataframe containing the matrix x containing the independent variables
anchor	is a dataframe containing the matrix anchor containing the anchor variable
gamma	is the regularization parameter for the Anchor Regression
target_variable	
	is the target variable name contained in the x dataframe
bin_factor	binary variable that can be transformed to a factor to partial out effects

Value

A list of predictions.

Examples

```
x <- as.data.frame(matrix(data = rnorm(10000),nrow = 1000,ncol = 10))
x$bin <- sample(nrow(x),x = c(1,0),prob = c(0.5,0.5),replace = TRUE)
anchor <- as.data.frame(matrix(data = rnorm(2000),nrow = 1000,ncol = 2))
colnames(anchor) <- c('X1','X2')
gamma <- 2
target_variable <- 'V2'

anchor_model <- anchor_regression_gam(x, anchor, gamma, target_variable,"bin")
anchor_prediction_gam(anchor_model$model, x, anchor, gamma, target_variable,"bin")
```

---

anchor_regression   *anchor_regression*

Description

Perform an Anchor Regression as described in Rothenhäusler et al.2020

Usage

```
anchor_regression(x, anchor, gamma, target_variable, lambda = "CV")
```

Arguments

`x`	is a dataframe containing the matrix x containing the independent variables
`anchor`	is a dataframe containing the matrix anchor containing the anchor variable
`gamma`	is the regularization parameter for the Anchor Regression
`target_variable`	
	is the target variable name contained in the x dataframe
`lambda`	indicates the lambda that is used in the Anchor Regression. 'CV' is used if it should be estimated by cross validation on the full subset.

Value

A list with coefficient values and a list with the respective names `overview_print`. Additionally the transformed data as x and y plus the fixed lambda coefficient.

Examples

```
x <- as.data.frame(matrix(data = rnorm(1000),nrow = 100,ncol = 10))
anchor <- as.data.frame(matrix(data = rnorm(200),nrow = 100,ncol = 2))
colnames(anchor) <- c('X1','X2')
gamma <- 2
target_variable <- 'V2'
anchor_regression(x, anchor, gamma, target_variable)
```

---

anchor_regression_gam   *anchor_regression_gam*

Description

Perform an Generalized Additive Anchor Regression

Usage

```
anchor_regression_gam(x, anchor, gamma, target_variable, bin_factor = NULL)
```

Arguments

`x`	is a dataframe containing the matrix x containing the independent variables
`anchor`	is a dataframe containing the matrix anchor containing the anchor variable
`gamma`	is the regularization parameter for the Anchor Regression
`target_variable`	
	is the target variable name contained in the x dataframe
`bin_factor`	binary variable that can be transformed to a factor to partial out effects

Value

A list with coefficient values and a list with the respective names `overview_print`. Additionally the transformed data as x and y plus the fixed lambda coefficient.

Examples

```
x <- as.data.frame(matrix(data = rnorm(10000),nrow = 1000,ncol = 10))
x$bin <- sample(nrow(x),x = c(1,0),prob = c(0.5,0.5),replace = TRUE)
anchor <- as.data.frame(matrix(data = rnorm(2000),nrow = 1000,ncol = 2))
colnames(anchor) <- c('X1','X2')
gamma <- 2
target_variable <- 'V2'
anchor_regression_gam(x, anchor, gamma, target_variable,bin_factor =  "bin")
```

---

anchor_stability   *anchor_stability*

Description

Perform an Anchor Stability Analysis as described in Rothenhäusler et al.2020

Usage

```
anchor_stability(
  x,
  anchor,
  target_variable,
  lambda = 0,
  alpha = 0.05,
  p_procedure = "naive"
)
```

Arguments

x	is a dataframe containing the matrix x containing the independent variables
anchor	is a dataframe containing the matrix anchor containing the anchor variable
target_variable	
	is the target variable name contained in the x dataframe
lambda	indicates the lambda that is used in the Anchor Regression. 'CV' is used if it should be estimated by cross validation on the full subset.
alpha	significance level for test decision on coefficient significance
p_procedure	procedure to estimate stability. Option 1: naive - stable if effect is non-zero in all cases; Option 2: post-lasso - post selection inference using SelectiveInference package

Value

A dataframe containing the stability values for each coefficient

Examples

```
x <- as.data.frame(matrix(data = rnorm(1000),nrow = 100,ncol = 10))
anchor <- as.data.frame(matrix(data = rnorm(200),nrow = 100,ncol = 2))
colnames(anchor) <- c('X1','X2')
gamma <- 2
target_variable <- 'V2'
anchor_stability(x, anchor, target_variable, lambda, alpha=0.05, p_procedure = "naive")
```

---

weighted_anchor_prediction
weighted_anchor_regression

Description

Perform a prediction for a Weighted Anchor Regression model

Usage

```
weighted_anchor_prediction(names, coeff, x, anchor, gamma, target_variable)
```

Arguments

names	list of variable names corresponding to the coefficients in coeff
coeff	list of coefficients corresponding to the coefficients in names
x	is a dataframe containing the matrix x containing the independent variables
anchor	is a dataframe containing the matrix anchor containing the anchor variable
gamma	is the regularization parameter for the Anchor Regression
target_variable	
	is the target variable name contained in the x dataframe

Value

A list of predictions.

Examples

```
# number of observed environments
environments <- 10

# populate list with generated data of x and anchor
data_x_list <- c()
data_anchor_list <- c()
for(e in 1:environments){
  x <- as.data.frame(matrix(data = rnorm(100),nrow = 100,ncol = 10))
  anchor <- as.data.frame(matrix(data = rnorm(200),nrow = 100,ncol = 2))
  colnames(anchor) <- c('X1','X2')
  data_x_list[[e]] <- x
```

```
    data_anchor_list[[e]]  <- anchor
  }

  # estimate model
  gamma <- 2
  target_variable <- 'V2'
  weighted_anchor_model <- weighted_anchor_regression(data_x_list,
                                                      data_anchor_list,
                                                      gamma,
                                                      target_variable,
                                                      anchor_model_pre=NULL,
                                                      test_split=0.4,
                                                      lambda=0)
  weighted_anchor_prediction(weighted_anchor_model$names,
                             weighted_anchor_model$coeff,
                             x,
                             anchor,
                             gamma,
                             target_variable)
```

---

weighted_anchor_regression

weighted_anchor_regression

Description

Estimates weighted Anchor Regression coefficients

Usage

```
weighted_anchor_regression(
  data_x_list,
  data_anchor_list,
  gamma,
  target_variable,
  anchor_model_pre = NULL,
  test_split = 0.4,
  lambda = 0
)
```

Arguments

data_x_list   list containing coefficient dataframes for different environments

data_anchor_list

                list containing anchor dataframes for different environments

gamma   is the regularization parameter for the Anchor Regression

target_variable

                is the target variable name contained in the x dataframe

anchor_model_pre

> is the pre estimated model for the Anchor Regression. In case of NULL a new model is estimated.

test_split   is desired test/train split for the estimation

lambda   penalization coefficient for Anchor Shrinkage. Initially set to 0.

Value

A list estimated coefficients with names, weights and the raw coefficient matrix

Examples

```
environments <- 10 # number of observed environments

# populate list with generated data of x and anchor
data_x_list <- c()
data_anchor_list <- c()
for(e in 1:environments){
  x <- as.data.frame(matrix(data = rnorm(100),nrow = 100,ncol = 10))
  anchor <- as.data.frame(matrix(data = rnorm(200),nrow = 100,ncol = 2))
  colnames(anchor) <- c('X1','X2')
  data_x_list[[e]] <- x
  data_anchor_list[[e]]  <- anchor
}

# estimate model
gamma <- 2
target_variable <- 'V2'
weighted_anchor_regression(data_x_list,
                           data_anchor_list,
                           gamma,
                           target_variable,
                           anchor_model_pre=NULL,
                           test_split=0.4,
                           lambda=0)
```

Index