Package 'fairadapt'

October 19, 2021

Title Fair Data Adaptation with Quantile Preservation

Description An implementation of the fair data adaptation with quantile preservation described in Plecko & Meinshausen (2019) <arXiv:1911.06685>. The adaptation procedure uses the specified causal graph to pre-process the given training and testing data in such a way to remove the bias caused by the protected attribute. The procedure uses tree ensembles for quantile regression.

Version 0.2.1

License GPL (>= 3)

Encoding UTF-8

Language en-US

LazyData true

URL https://github.com/dplecko/fairadapt

BugReports https://github.com/dplecko/fairadapt/issues

Depends R (>= 3.5.0)

Imports ranger (>= 0.13.1), assertthat, quantreg, qrnn, igraph, ggplot2, cowplot, scales

Suggests testthat (>= 3.0.3), knitr, rmarkdown, markdown, data.table, rticles, mvtnorm, magick, ggraph, pdftools

RoxygenNote 7.1.2

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Drago Plecko [aut, cre], Nicolas Bennett [aut]

Maintainer Drago Plecko <drago.plecko@stat.math.ethz.ch>

Repository CRAN

Date/Publication 2021-10-19 17:20:02 UTC

R topics documented:

`adaptedData` *Convenience function for returning adapted data*

Description

Convenience function for returning adapted data

Usage

```
adaptedData(x, train = TRUE)
```

Arguments

`x`	Object of class `fairadapt`, a result of an adaptation procedure.
`train`	A logical indicating whether train data should be returned. Defaults to `TRUE`. If `FALSE`, test data is returned.

`computeQuants` *Compute Quantiles generic for the Quantile Learning step.*

Description

Compute Quantiles generic for the Quantile Learning step.

Usage

```
computeQuants(x, data, newdata, ind, ...)
```

Arguments

x	Object with an associated ‘computeQuants()‘ method, to be used for inferring quantiles.
data	data.frame containing samples used in the quantile regression.
newdata	data.frame containing counterfactual values for which the quantiles need to be inferred.
ind	A logical vector of length ‘nrow(data)‘, indicating which samples have the baseline value of the protected attribute.
...	Additional arguments to be passed down to respective method functions.

Value

A vector of counterfactual values corresponding to ‘newdata‘.

---

fairadapt *Fairadapt*

Description

Implementation of fair data adaptation with quantile preservation (Plecko & Meinshausen 2019). Uses only plain R.

Usage

```
fairadapt(
  formula,
  prot.attr,
  adj.mat,
  train.data,
  test.data = NULL,
  cfd.mat = NULL,
  top.ord = NULL,
  res.vars = NULL,
  quant.method = rangerQuants,
  visualize.graph = FALSE,
  ...
)
```

Arguments

formula	Object of class formula describing the response and the covariates.
prot.attr	A value of class character describing the binary protected attribute. Must be one of the entries of colnames(adj.mat).

`adj.mat` Matrix of class `matrix` encoding the relationships in the causal graph. `M[i,j] == 1L` implies the existence of an edge from node i to node j. Must include all the variables appearing in the formula object. When the `adj.mat` argument is set to `NULL`, then the `top.ord` argument has to be supplied.

`train.data, test.data` Training data & testing data, both of class `data.frame`. Test data is by default `NULL`.

`cfd.mat` Symmetric matrix of class `matrix` encoding the bidirected edges in the causal graph. `M[i,j] == M[j,i] == 1L` implies the existence of a bidirected edge between nodes i and j. Must include all the variables appearing in the formula object.

`top.ord` A vector of class `character` describing the topological ordering of the causal graph. Default value is `NULL`, but this argument must be supplied if `adj.mat` is not specified. Also must include all the variables appearing in the formula object.

`res.vars` A vector of class `character` listing all the resolving variables, which should not be changed by the adaption procedure. Default value is `NULL`, corresponding to no resolving variables. Resolving variables should be a subset of the descendants of the protected attribute.

`quant.method` A function choosing the method used for quantile regression. Default value is `rangerQuants` (using random forest quantile regression). Other implemented options are `linearQuants` and `mcqrnnQuants`. A custom function can be supplied by the user here, and the associated method for the S3 generic `computeQuants` needs to be added.

`visualize.graph` A `logical` indicating whether the causal graph should be plotted upon calling the `fairadapt()` function. Default value is `FALSE`.

`...` Additional arguments forwarded to the function passed as 'quant.method'.

Details

The procedure takes the training and testing data as an input, together with the causal graph given by an adjacency matrix and the list of resolving variables, which should be kept fixed during the adaptation procedure. The procedure then calculates a fair representation of the data, after which any classification method can be used. There are, however, several valid training options yielding fair predictions, and the best of them can be chosen with cross-validation. For more details we refer the user to the original paper. Most of the running time is due to the quantile regression step using the ranger package.

Value

An object of class `fairadapt`, containing the original and adapted training and testing data, together with the causal graph and some additional meta-information.

Author(s)

Drago Plecko

References

Plecko, D. & Meinshausen, N. (2019). Fair Data Adaptation with Quantile Preservation

Examples

```
uni.adj.mat <- array(0, dim = c(4, 4))
colnames(uni.adj.mat) <- rownames(uni.adj.mat) <-
  c("gender", "edu", "test", "score")

uni.adj.mat["gender", c("edu", "test")] <-
  uni.adj.mat["edu", c("test", "score")] <-
  uni.adj.mat["test", "score"] <- 1L

FA <- fairadapt(score ~ .,
  train.data = uni_admission[1:100, ],
  test.data = uni_admission[101:150, ],
  adj.mat = uni.adj.mat, prot.attr = "gender")

FA
```

`fairTwins` *Fair Twin Inspection convenience function.*

Description

Fair Twin Inspection convenience function.

Usage

```
fairTwins(x, train.id = seq_len(nrow(x$train)), test.id = NULL, cols = NULL)
```

Arguments

`x`	Object of class `fairadapt`, a result of an adaptation procedure.
`train.id`	A vector of indices specifying which rows of the training data should be displayed.
`test.id`	A vector of indices specifying which rows of the test data should be displayed.
`cols`	A `character` vector, subset of `names(train.data)`, which specifies which subset of columns is to be displayed in the result.

Value

A `data.frame`, containing the original and adapted values of the requested individuals. Adapted columns have `_adapted` appended to their original name.

Examples

```
uni.adj.mat <- array(0, dim = c(4, 4))
colnames(uni.adj.mat) <- rownames(uni.adj.mat) <-
  c("gender", "edu", "test", "score")

uni.adj.mat["gender", c("edu", "test")] <-
  uni.adj.mat["edu", c("test", "score")] <-
  uni.adj.mat["test", "score"] <- 1L

FA <- fairadapt(score ~ .,
  train.data = uni_admission[1:100, ],
  test.data = uni_admission[101:150, ],
  adj.mat = uni.adj.mat, prot.attr = "gender")

fairTwins(FA, train.id = 1:5)
```

---

gov_census *Census information of US government employees.*

Description

The dataset contains various demographic, education and work information of the employees of the US government. The data is taken from the 2018 US Census data.

Usage

```
gov_census
```

Format

A data frame with 204,309 rows and 10 variables:

sex gender of the employee

age employee age in years

race race of the employee

hispanic_origin indicator of hispanic origin

citizenship citizenship of the employee

nativity indicator of nativity to the US

marital marital status

family_size size of the employee's family

children number of children of the employee

education_level education level measured in years

english_level

salary yearly salary in US dollars

hours_worked hours worked every week

weeks_worked weeks worked in the given year

occupation occupation classification

industry industry classification

economic_region economic region where the person is employed in the US

Source

https://www.census.gov/programs-surveys/acs/microdata/documentation.html

---

`graphModel` *Obtaining the graphical causal model (GCM)*

Description

Obtaining the graphical causal model (GCM)

Usage

```
graphModel(adj.mat, cfd.mat = NULL, res.vars = NULL)
```

Arguments

`adj.mat`	Matrix of class `matrix` encoding the relationships in the causal graph. `M[i,j] == 1L` implies the existence of an edge from node i to node j.
`cfd.mat`	Symmetric matrix of class `matrix` encoding the bidirected edges in the causal graph. `M[i,j] == M[j,i] == 1L` implies the existence of a bidirected edge between nodes i and j.
`res.vars`	A vector of class `character` listing all the resolving variables, which should not be changed by the adaption procedure. Default value is `NULL`, corresponding to no resolving variables. Resolving variables should be a subset of `colnames(adj.mat)`. Resolving variables are marked with a different color in the output.

Value

An object of class `igraph`, containing the causal graphical, with directed and bidirected edges.

Examples

```
adj.mat <- cfd.mat <- array(0L, dim = c(3, 3))
colnames(adj.mat) <- rownames(adj.mat) <-
  colnames(cfd.mat) <- rownames(cfd.mat) <- c("A", "X", "Y")

adj.mat["A", "X"] <- adj.mat["X", "Y"] <-
  cfd.mat["X", "Y"] <- cfd.mat["Y", "X"] <- 1L

gcm <- graphModel(adj.mat, cfd.mat, res.vars = "X")
```

`linearQuants` *Compute Quantiles using linear quantile regression (‘quantreg‘ package) in the Quantile Learning step.*

Description

Compute Quantiles using linear quantile regression (‘quantreg‘ package) in the Quantile Learning step.

Usage

```
linearQuants(
  data,
  A.root,
  ind,
  tau = c(0.001, seq(0.005, 0.995, by = 0.01), 0.999),
  ...
)
```

Arguments

`data`	A `data.frame` with data to be used for quantile regression.
`A.root`	A `logical(1L)` indicating whether the protected attribute ‘A‘ is a root node of the causal graph. Used for splitting the quantile regression.
`ind`	A `logical` vector of length ‘nrow(data)‘, indicating which samples have the baseline value of the protected attribute.
`tau, ...`	Forwarded to [quantreg::rq()].

Value

A ‘rqs‘ or a ‘quantregsplit‘ ‘S3‘ object, depending on the value of the ‘A.root‘ argument.

`mcqrnnQuants` *Compute Quantiles using monotone quantile regression neural networks (‘mcqrnn‘ package) in the Quantile Learning step.*

Description

Compute Quantiles using monotone quantile regression neural networks (‘mcqrnn‘ package) in the Quantile Learning step.

Usage

```
mcqrnnQuants(
  data,
  A.root,
  ind,
  tau = seq(0.005, 0.995, by = 0.01),
  iter.max = 500,
  ...
)
```

Arguments

`data`	A `data.frame` with data to be used for quantile regression.
`A.root`	A `logical(1L)` indicating whether the protected attribute 'A' is a root node of the causal graph. Used for splitting the quantile regression.
`ind`	A `logical` vector of length 'nrow(data)', indicating which samples have the baseline value of the protected attribute.
`tau, iter.max, ...`	Forwarded to [qrnn::mcqrnn.fit()].

Value

An 'mcqrnn' 'S3' object.

---

rangerQuants — *Compute Quantiles using random forests ('ranger' package) in the Quantile Learning step.*

Description

Compute Quantiles using random forests ('ranger' package) in the Quantile Learning step.

Usage

```
rangerQuants(data, A.root, ind, min.node.size = 20, ...)
```

Arguments

`data`	A `data.frame` with data to be used for quantile regression.
`A.root`	A `logical(1L)` indicating whether the protected attribute 'A' is a root node of the causal graph. Used for splitting the quantile regression.
`ind`	A `logical` vector of length 'nrow(data)', indicating which samples have the baseline value of the protected attribute.
`min.node.size, ...`	Forwarded to [ranger::ranger()].

Value

A ‘ranger‘ or a ‘rangersplit‘ ‘S3‘ object, depending on the value of the ‘A.root‘ argument.

`uni_admission` *Univeristy admission data of 1,000 students.*

Description

A simulated dataset containing the evaluation of students' abilities.

Usage

```
uni_admission
```

Format

A data frame with 1,000 rows and 4 variables:

gender the gender of the student

edu educational achievement, for instance GPA

test performance on a university admission test

score overall final score measuring the quality of a candidate

---

`visualizeGraph` *Visualize Graphical Causal Model*

Description

Visualize Graphical Causal Model

Usage

```
visualizeGraph(x, ...)
```

Arguments

`x`	Object of class `fairadapt`, a result of an adaptation procedure.
`...`	Additional arguments passed to the graph plotting function.

Index