Package ‘MatchingFrontier’

March 1, 2015

Version 0.15.28
Date 2015-01-20
Type Package
Title Computation of the Balance - Sample Size Frontier in Matching Methods for Causal Inference
Author Gary King, Christopher Lucas, and Richard Nielsen
Maintainer Christopher Lucas <clucas@fas.harvard.edu>
Description MatchingFrontier returns the subset of the data with the minimum imbalance for every possible subset size (N - 1, N - 2, ...), down to the data set with the minimum possible imbalance. The package also includes tools for the estimation of causal effects for each subset size, as well as functions for visualization and data export. MatchingFrontier also contains functions for calculating model dependence, as proposed by Athey and Imbens.

URL http://projects.iq.harvard.edu/frontier
Imports MASS, igraph, segmented
Suggests stargazer
LazyData true
License GPL-3

R topics documented:

estimateEffects .................................................. 2
generateDataset .................................................. 3
lalonde ......................................................... 4
makeFrontier .................................................... 5
modelDependence ............................................... 6
parallelPlot ..................................................... 7
plotEstimates .................................................. 8
plot Frontier .................................................... 9
plotMeans ....................................................... 10

Index 12
**estimateEffects**  

*Estimate Effects on the Frontier*

**Description**

`estimateEffects()` is used to estimate the effect of the treatment along the entire frontier.

**Usage**

```r
estimateEffects(frontier.object, formula, prop.estimated = 1,  
                 mod.dependence.formula, continuous.vars = NA,  
                 seed = 1, means.as.cutpoints = FALSE)
```

**Arguments**

- **frontier.object**
  An object generated by `makeFrontier()`.

- **formula**
  An object of class formula (or one that can be coerced to that class). This will be passed to `lm()` to estimate the point estimates for the causal effect estimates across the frontier.

- **prop.estimated**
  The proportion of points on the frontier to estimate. By default, 100% of the points on the frontier are estimated. To estimate less than 100% of the points, pass the proportion to be estimated to prop.estimated (for example, .6 to estimate 60% of the points).

- **mod.dependence.formula**
  The formula used as the base formula for the Athey-Imbens model dependence estimates.

- **continuous.vars**
  All continuous control variables in mod.dependence.formula must be passed as a character vector to continuous.vars. A cutpoint for each of these variables will be estimated with segmented regression.

- **seed**
  The seed used before estimation of the effects. If prop.estimated is less than 1, this is necessary in order to replicate the exact plot.

- **means.as.cutpoints**
  FALSE by default. If TRUE, cutpoints are calculated as the mean instead of the breakpoint in a segmented regression. This is sometimes much faster.

**References**

generateDataset

Examples

data(lalonde)

match.on <- colnames(lalonde)[!(colnames(lalonde) %in% c('re78', 'treat'))]

my.frontier <- makefrontier(dataset = lalonde, 
                           treatment = 'treat', 
                           outcome = 're78', 
                           match.on = match.on)

my.form <- as.formula(re78 ~ treat + age + black + education + hispanic + 
                       married + nnodegree + re74 + re75)

## Not run:
my.estimates <- estimateEffects(my.frontier, 're78 ~ treat', 
                                mod.dependence.formula = my.form, 
                                continuous.vars = c('age', 'education', 're74', 're75'), 
                                prop.estimated = .1, 
                                means.as.cutpoints = TRUE)

## End(Not run)

---

**generateDataset** Generate a data set that is on the balance - sample size frontier.

Description
generateDataset() allows the user to export a data set that sits on the frontier.

Usage
generateDataset(frontier.object, N)

Arguments

- **frontier.object**
  - An object generated by `makeFrontier()`.

- **N**
  - The number of observations left in the exported data set. If the user selects an undefined point, `generateDataset` returns a dataset from the nearest defined point on the frontier.

References

Examples

data(lalonde)

match.on <- colnames(lalonde)[!(colnames(lalonde) %in% c('re78', 'treat'))]
my.frontier <- makefrontier(dataset = lalonde, 
treatment = 'treat',
outcome = 're78',
match.on = match.on)
n <- 300 # Identify the point from which to select the data
matched.data <- generateDataset(my.frontier, N = n)

---

lalonde

**Modified Lalonde dataset**

Description

This is a modified version of the Lalonde experimental dataset used for explanatory purposes only.

Usage

data(lalonde)

Format

A data frame with 16437 observations on the following 11 variables.

treat treatment variable indicator
age age
education years of education
black race indicator variable
married marital status indicator variable
nodegree marital status indicator variable for not possessing a degree
re74 real earnings in 1974
re75 real earnings in 1975
re78 real earnings in 1978 (post-treatment outcome)
hispanic ethnic indicator variable
**Description**

*makeFrontier()* computes the balance-frontier sample size and can be used with *estimateEffects* to estimate effects along the balance-sample size frontier.

**Usage**

```r
makeFrontier(dataset, treatment, outcome, match.on, 
keep.vars = NULL, QOI = 'FSATT', metric = 'Mahal', 
ratio = 'fixed', breaks = NULL)
```

**Arguments**

- **dataset**: The data set containing the treatment, outcome, and variable to match on.
- **treatment**: The name of the treatment.
- **outcome**: The name of the outcome.
- **match.on**: A vector of colnames indicating which variables are to be matched on.
- **keep.vars**: A character vector of variable names that are not in treatment, outcome, or 'match.on' but that the user would like to store in the data, either for calculation of model dependence intervals or for use in exported data sets.
- **QOI**: The quantity of interest to be estimated. By default, feasible sample average treatment effect on the treated or FSATT. The other option is SATT (sample average treatment effect on the treated).
- **metric**: The metric used to measure imbalance. Defaults to average mahalanobis distance to nearest match. The other option is L1.
- **ratio**: Variable or fixed ratio. See King, Lucas, and Nielsen for details.
- **breaks**: Can be used with L1 to provide user-specified breaks.

**References**


**Examples**

```r
data(lalonde)

match.on <- colnames(lalonde)[!(colnames(lalonde) %in% c('re78', 'treat'))]
my.frontier <- makeFrontier(dataset = lalonde, 
treatment = 'treat',
outcome = 're78',
match.on = match.on)
```
modelDependence

Compute the Athey-Imbens measure of sensitivity to model specification.

Description

modelDependence() is used to compute the Athey-Imbens measure of sensitivity to model specification.

Usage

modelDependence(dataset, treatment, base.form, verbose = TRUE,
seed = 1, cutpoints = NA)

Arguments

dataset A data frame containing the variables in the model.
treatment The treatment (quantity of interest). The measure of model dependence is with
respect to estimates of this quantity. Must be in base.form.
base.form The base formula that is to be evaluated.
verbose If TRUE, additional information is printed.
seed Seed for the random number generator.
cutpoints A list where the keys are variables names and the values are cutpoints. If spec-
ified, cutpoints for these variables will not be estimated. Otherwise, cutpoints
are estimated with segmented regression.

References


Examples

data(lalonde)
treatment <- 'treat'
base.form <- as.formula('re78 ~ treat + age + education
                         + black + hispanic + married +
                         nodegree + re74 + re75')

md <- modelDependence(lalonde, treatment, base.form,
cutpoints = list('age' = mean(lalonde$age)))
print(md$sigma.hat.theta)
parallelPlot

Create a parallel plot for a specified point on the frontier.

Description

parallelPlot() creates a parallel plot for a specified point on the frontier. Wraps parcoord() from MASS.

Usage

parallelPlot(frontier.object, N, variables, treated.col = 'grey', control.col = 'black')

Arguments

frontier.object
An object generated by makeFrontier().

N
The number of observations left in the exported data set. If the user selects an undefined point, generateDataset returns a dataset from the nearest defined point on the frontier.

variables
The variables to be included in the parallel plot.

treated.col
The color of the lines corresponding to observations assigned to the treatment. Grey by default.

treated.col
The color of the lines corresponding to observations assigned to the control. Black by default.

References


Examples

data(lalonde)

match.on <- colnames(lalonde)[!(colnames(lalonde) %in% c('re78', 'treat'))]
mahal.frontier <- makeFrontier(dataset = lalonde,
treatment = 'treat',
outcome = 're78',
match.on = match.on)

parallelPlot(mahal.frontier,
N = 300,
variables = c('age',
're74',
're75',
'black'),
treated.col = 'grey',
treated.col = 'blue')
plotEstimates

Description

plotEstimates() plots estimates along the frontier.

Usage

plotEstimates(estimates.object, 
  xlab = 'Number of Observations Pruned', 
  ylab = 'Estimate', 
  main = 'Effects Plot', 
  xlim = NULL, 
  ylim = NULL, 
  mod.dependence.col = rgb(255,0,0,127, maxColorValue=255), 
  mod.dependence.border.col = rgb(255,0,0,200, maxColorValue=255), 
  line.col = rgb(102,0,0,255, maxColorValue=255), 
  ...)

Arguments

  estimates.object
    An object generated by estimateEffects()

  xlab
    The label for the x-axis. Defaults to 'Number of Observations Pruned'.

  ylab
    The label for the y-axis. Defaults to 'Estimate'.

  main
    The main label. Defaults to 'Effects Plot'.

  xlim
    The x-axis limits.

  ylim
    The y-axis limits.

  ...
    Additional arguments to be passed to plot.

  mod.dependence.col
    The color to shade the model dependence region.

  mod.dependence.border.col
    The model dependence region border color.

  line.col
    The color of the line displaying point estimates.

Details

plotEstimates() wraps plot and uses ... to pass additional arguments to the base plot() function, like color, axis range, etc.

References

Examples

```r
data(lalonde)

match.on <- colnames(lalonde)[!(colnames(lalonde) %in% c('re78', 'treat'))]
my.frontier <- makefrontier(dataset = lalonde,
  treatment = 'treat',
  outcome = 're78',
  match.on = match.on)

base.form <- as.formula('re78 ~ treat + age + education
  + black + hispanic + married +
  nodegree + re74 + re75')

## Not run:
my.estimates <- estimateEffects(my.frontier,
  're78 ~ treat',
  mod.dependence.formula = base.form,
  continuous.vars = c('age', 'education', 're74', 're75'),
  prop.estimated = .1,
  means.as.cutpoints = TRUE)

plotEstimates(my.estimates)

## End(Not run)
```

Description

`plotFrontier()` plots the balance - sample size frontier.

Usage

```r
plotFrontier(frontier.object, xlab = "Number of Observations Pruned",
  ylab = frontier.object$metric, main = "Frontier Plot", ...)
```

Arguments

- `frontier.object`: An object generated by `makeFrontier()`.
- `xlab`: The label for the x-axis. Defaults to 'Number of Observations Pruned'.
- `ylab`: The label for the y-axis. Defaults to the selected metric.
- `main`: The main label. Defaults to 'Effects Plot'.
- `...`: Additional arguments to be passed to `plot`.

Details

`plotEstimates()` wraps `plot` and uses ...to pass additional arguments to the base `plot()` function, like color, axis range, etc.
References


Examples

data(lalonde)

match.on <- colnames(lalonde)[!(colnames(lalonde) %in% c('re78', 'treat'))]
my.frontier <- makeFrontier(dataset = lalonde,
    treatment = 'treat',
    outcome = 're78',
    match.on = match.on)

plotFrontier(my.frontier)

plotMeans

Plot covariate means along the frontier.

Description

plotMeans() plots means along the frontier.

Usage

plotMeans(frontier.object,
    xlab = 'Number of Observations Pruned',
    main = 'Means Plot',
    xlim = c(1, max(frontier.object$frontier$Xs)),
    ylim = c(0, 1),
    cols = rainbow(length(frontier.object$match.on)),
    diff.in.means = FALSE,
    ...)

Arguments

frontier.object
    An object generated by makeFrontier()

xlab
    The label for the x-axis. Defaults to 'Number of Observations Pruned'.

main
    The main label. Defaults to Means Plot'.

xlim
    The x-axis limits. Defaults to the range of the frontier.

ylim
    The y-axis limits. Defaults to (0, 1).

cols
    The line colors. Defaults to the rainbow palette.

diff.in.means
    If TRUE, means are the difference in means between treated and control groups.
    If FALSE (the default), means are the covariate means pooling across treated and control.

...
    Additional arguments to be passed to plot.
**Details**

`plotMeans()` wraps `plot` and uses ... to pass additional arguments to the base `plot()` function.

**References**


**Examples**

```r
data(lalonde)

match.on <- colnames(lalonde)[!(colnames(lalonde) %in% c('re78', 'treat'))]
my.frontier <- makeFrontier(dataset = lalonde,
                           treatment = 'treat',
                           outcome = 're78',
                           match.on = match.on)

plotMeans(my.frontier)
```
Index

estimateEffects, 2
generateDataset, 3
lalonde, 4
makeFrontier, 5
modelDependence, 6
parallelPlot, 7
plotEstimates, 8
plotFrontier, 9
plotMeans, 10