Package 'mplot'

June 11, 2015

Type Package

Title Graphical Model Stability and Variable Selection Procedures

Version 0.6.2

Date 2015-06-10

Description Model stability and variable importance plots [Mueller and Welsh (2010, <doi:10.1111/j.1751-5823.2010.00108.x>); Murray, Heritier and Mueller (2013, <doi:10.1002/sim.5855>)] as well as the adaptive fence [Jiang et al. (2008, doi:10.1214/07-AOS517>); Jiang et al. (2009, <doi:10.1016/j.spl.2008.10.014>)] for linear and generalised linear models.

License GPL (>= 2)

Suggests knitr, mvoutlier, glmnet

Imports leaps, foreach, parallel, bestglm, doParallel, plyr, shinydashboard, shiny, googleVis

URL https://github.com/garthtarr/mplot

LazyData TRUE

NeedsCompilation no

Author Garth Tarr [aut, cre], Samuel Mueller [aut], Alan Welsh [aut]

Maintainer Garth Tarr <garth.tarr@gmail.com>

Repository CRAN

Date/Publication 2015-06-11 01:33:07

R topics documented:

`mplot-package` *Graphical model stability and model selection procedures*

Description

Graphical model stability and model selection procedures

`af` *The adaptive fence procedure*

Description

This function implements the adaptive fence procedure to first find the optimal cstar value and then finds the corresponding best model as described in Jiang et. al. (2009) with some practical modifications.

Usage

```
af(mf, B = 60, n.c = 20, initial.stepwise = FALSE, force.in = NULL,
  n.cores, nvmax, c.max, screen = FALSE, ...)
```

Arguments

`mf`	a fitted 'full' model, the result of a call to lm or glm (and in the future lme or lmer).
`B`	number of bootstrap replications at each fence boundary value
`n.c`	number of boundary values to be considered
`initial.stepwise`	logical. Performs an initial stepwise procedure to look for the range of model sizes where attention should be focussed. See details for implementation.
`force.in`	the names of variables that should be forced into all estimated models
`n.cores`	number of cores to be used when parallel processing the bootstrap
`nvmax`	size of the largest model that can still be considered as a viable candidate. Included for performance reasons but if it is an active constraint it could lead to missleading results.

c.max	manually specify the upper boundary limit. Only applies when initial.stepwise=FALSE.
screen	logical, whether or not to perform an initial screen for outliers. Highly experimental, use at own risk. Default = FALSE.
...	further arguments (currently unused)

Details

The initial stepwise procedure performs forward stepwise model selection using the AIC and backward stepwise model selection using BIC. In general the backwise selection via the more conservative BIC will tend to select a smaller model than that of the forward selection AIC approach. The size of these two models is found, and we go two dimensions smaller and larger to estimate a sensible range of c values over which to perform a parametric bootstrap.

This procedure can take some time. It is recommended that you start with a relatively small number of bootstrap samples (B) and grid of boundary values (n.c) and increase both as required.

If you use initial.stepwise=TRUE then in general you will need a smaller grid of boundary values than if you select initial.stepwise=FALSE. It can be useful to check initial.stepwise=FALSE with a small number of bootstrap replications over a sparse grid to ensure that the initial.stepwise=TRUE has landed you in a reasonable region.

The best.only=FALSE option when plotting the results of the adaptive fence is a modification to the adaptive fence procedure which considers all models at a particular size that pass the fence hurdle when calculating the p* values. In particular, for each value of c and at each bootstrap replication, if a candidate model is found that passes the fence, then we look to see if there are any other models of the same size that also pass the fence. If no other models of the same size pass the fence, then that model is allocated a weight of 1. If there are two models that pass the fence, then the best model is allocated a weight of 1/2. If three models pass the fence, the best model gets a weight of 1/3, and so on. After B bootstrap replications, we aggregate the weights by summing over the various models. The p* value is the maximum aggregated weight divided by the number of bootstrap replications. This correction penalises the probability associated with the best model if there were other models of the same size that also passed the fence hurdle. The rationale being that if a model has no redundant variables then it will be the only model at that size that passes the fence over a range of values of c. The result is more pronounced peaks which can help to determine the location of the correct peak and identify the optimal c*.

References

Jiang J., Nguyen T., Sunil Rao J. (2009), A simplified adaptive fence procedure, Statistics & Probability Letters, 79(5):625-629. doi: 10.1016/j.spl.2008.10.014

Jiang J., Sunil Rao J., Gu Z, Nguyen T. (2008), Fence methods for mixed model selection, Annals of Statistics, 36(4):1669-1692. doi: 10.1214/07-AOS517

Examples

```
n = 100
set.seed(11)
e = rnorm(n)
x1 = rnorm(n)
x2 = rnorm(n)
x3 = x1^2
```

```
x4 = x2^2
x5 = x1*x2
x6 = rep(c("A","B"),n=50)
y = 1 + x1 + x2 + e
dat = data.frame(y,x1,x2,x3,x4,x5,x6)
lm1 = lm(y~.,data=dat)
## Not run:
af1 = af(lm1, n.cores=4, initial.stepwise=TRUE)
summary(af1)
plot(af1)

## End(Not run)
```

`artificialeg` *Artificial example*

Description

An artificial data set which causes stepwise regression procedures to select a non-parsimonious model. The true model is a simple linear regression of y against x8.

Usage

```
data(artificialeg)
```

Format

A data frame with 50 observations on 10 variables.

Details

Inspired by the pathoeg data set in the MPV pacakge.

Examples

```
data(artificialeg)
full.mod = lm(y~.,data=artificialeg)
step(full.mod)
## Not run:
# generating model
n=50
set.seed(8) # a seed of 2 also works
x1 = rnorm(n,0.22,2)
x7 = 0.5*x1 + rnorm(n,0,sd=2)
x6 = -0.75*x1 + rnorm(n,0,3)
x3 = -0.5-0.5*x6 + rnorm(n,0,2)
x9 = rnorm(n,0.6,3.5)
x4 = 0.5*x9 + rnorm(n,0,sd=3)
x2 = -0.5 + 0.5*x9 + rnorm(n,0,sd=2)
x5 = -0.5*x2+0.5*x3+0.5*x6-0.5*x9+rnorm(n,0,1.5)
```

```
x8 = x1 + x2 -2*x3 - 0.3*x4 + x5 - 1.6*x6 - 1*x7 + x9 +rnorm(n,0,0.5)
y = 0.6*x8 + rnorm(n,0,2)
artificialeg = round(data.frame(x1,x2,x3,x4,x5,x6,x7,x8,x9,y),1)

## End(Not run)
```

---

bodyfat *Body fat data set*

Description

A data frame with 128 observations on 15 variables.

Usage

```
data(bodyfat)
```

Format

A data frame with 128 observations on 15 variables.

Id Identifier

Bodyfat Bodyfat percentage

Age Age (years)

Weight Weight (kg)

Height Height (inches)

Neck Neck circumference (cm)

Chest Chest circumference (cm)

Abdo Abdomen circumference (cm) "at the umbilicus and level with the iliac crest"

Hip Hip circumference (cm)

Thigh Thigh circumference (cm)

Knee Knee circumference (cm)

Ankle Ankle circumference (cm)

Bic Extended biceps circumference (cm)

Fore Forearm circumference (cm)

Wrist Wrist circumference (cm) "distal to the styloid processes"

Details

A subset of the 252 observations available in the `mfp` package. The selected observations avoid known high leverage points and outliers. The unused points from the data set could be used to validate selected models.

References

Johnson W (1996, Vol 4). Fitting percentage of body fat to simple body measurements. Journal of Statistics Education. Bodyfat data retrieved from http://www.amstat.org/publications/jse/v4n1/datasets.johnson.html An expanded version is included in the `mfp` R package.

Examples

```
data(bodyfat)
full.mod = lm(Bodyfat~.,data=subset(bodyfat,select=-Id))
```

`diabetes` *Blood and other measurements in diabetics*

Description

The diabetes data frame has 442 rows and 11 columns. These are the data used in Efron et al. (2004).

Usage

```
data(diabetes)
```

Format

A data frame with 442 observations on 11 variables.

age Age

sex Gender

bmi Body mass index

map Mean arterial pressure (average blood pressure)

tc Total cholesterol (mg/dL)? Desirable range: below 200 mg/dL

ldl Low-density lipoprotein ("bad" cholesterol)? Desirable range: below 130 mg/dL

hdl High-density lipoprotein ("good" cholesterol)? Desirable range: above 40 mg/dL

tch Blood serum measurement

ltg Blood serum measurement

glu Blood serum measurement (glucose?)

y A quantitative measure of disease progression one year after baseline

Details

Data sourced from http://web.stanford.edu/~hastie/Papers/LARS

References

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., (2004). Least angle regression. The Annals of Statistics 32(2) 407-499. DOI: 10.1214/009053604000000067

Examples

```
data(diabetes)
full.mod = lm(y~.,data=diabetes)
```

`mplot` *Model selection and stability curves*

Description

Opens a shiny GUI to investigate a range of model selection and stability issues

Usage

```
mplot(mf, ...)
```

Arguments

`mf`	a fitted model.
`...`	objects of type vis or af.

Examples

```
n = 100
set.seed(11)
e = rnorm(n)
x1 = rnorm(n)
x2 = rnorm(n)
x3 = x1^2
x4 = x2^2
x5 = x1*x2
y = 1 + x1 + x2 + e
dat = round(data.frame(y,x1,x2,x3,x4,x5),2)
lm1 = lm(y~.,data=dat)
## Not run:
v1 = vis(lm1,n.cores=3)
af1 = af(lm1,n.cores=3)
mplot(lm1,v1,af1)

## End(Not run)
```

plot.af *Plot diagnostics for an af object*

Description

Summary plot of the bootstrap results of an af object.

Usage

```
## S3 method for class 'af'
plot(x, pch, classic = FALSE, html.only = FALSE,
  best.only = TRUE, width = 800, height = 400, fontSize = 12,
  left = 50, top = 30, chartWidth = "60%", chartHeight = "80%",
  backgroundColor = "transparent", options = NULL, ...)
```

Arguments

x	af object, the result of af
pch	plotting character, i.e., symbol to use
classic	logical. If classic=TRUE a base graphics plot is provided instead of a googleVis plot. Default is classic=FALSE.
html.only	logical for use with shiny. With rmarkdown using op = options(gvis.plot.tag = "chart") is better.
best.only	logical determining whether the output used the standard fence approach of only considering the best models that pass the fence (TRUE) or if it should take into account all models that pass the fence at each boundary value (FALSE).
width	Width of the googleVis chart canvas area, in pixels. Default: 800.
height	Height of the googleVis chart canvas area, in pixels. Default: 400.
fontSize	font size used in googleVis chart. Default: 12.
left	space at left of chart (pixels?). Default: "50".
top	space at top of chart (pixels?). Default: "30".
chartWidth	googleVis chart area width. A simple number is a value in pixels; a string containing a number followed by % is a percentage. Default: "60%"
chartHeight	googleVis chart area height. A simple number is a value in pixels; a string containing a number followed by % is a percentage. Default: "80%"
backgroundColor	The background colour for the main area of the chart. A simple HTML color string, for example: 'red' or '#00cc00'. Default: 'transparent'
options	If you want to specify the full set of googleVis options.
...	further arguments (currently unused)

plot.vis *Plot diagnostics for a vis object*

Description

A plot method to visualise the results of a vis object.

Usage

```
## S3 method for class 'vis'
plot(x, highlight, classic = FALSE, html.only = FALSE,
  which = c("vip", "lvk", "boot"), width = 800, height = 400,
  fontSize = 12, left = 50, top = 30, chartWidth = "60%",
  chartHeight = "80%", axisTitlesPosition = "out", dataOpacity = 0.5,
  options = NULL, backgroundColor = "transparent", text = FALSE,
  min.prob = 0.4, srt = -30, max.circle = 0.35,
  print.full.model = FALSE, jitterk = 0.1, ...)
```

Arguments

x	vis object, the result of vis
highlight	the name of a variable that will be highlighted
classic	logical. If classic=TRUE a base graphics plot is provided instead of a googleVis plot. Default is classic=FALSE.
html.only	logical. Use html.only=TRUE when including interactive plots in markdown documents (this includes rpres files).
which	a vector specifying the plots to be output. Variable inclusion plots which="vip"; description loss against model size which="lvk"; bootstrapped description loss against model size which="boot".
width	Width of the googleVis chart canvas area, in pixels. Default: 800.
height	Height of the googleVis chart canvas area, in pixels. Default: 400.
fontSize	font size used in googleVis chart. Default: 12.
left	space at left of chart (pixels?). Default: "50".
top	space at top of chart (pixels?). Default: "30".
chartWidth	googleVis chart area width. A simple number is a value in pixels; a string containing a number followed by % is a percentage. Default: "60%"
chartHeight	googleVis chart area height. A simple number is a value in pixels; a string containing a number followed by % is a percentage. Default: "80%"
axisTitlesPosition	Where to place the googleVis axis titles, compared to the chart area. Supported values: "in" - Draw the axis titles inside the the chart area. "out" - Draw the axis titles outside the chart area. "none" - Omit the axis titles.

dataOpacity	The transparency of googleVis data points, with 1.0 being completely opaque and 0.0 fully transparent.
options	a list to be passed to the googleVis function giving complete control over the output. Specifying a value for options overwrites all other plotting variables.
backgroundColor	The background colour for the main area of the chart. A simple HTML color string, for example: 'red' or '#00cc00'. Default: 'null' (there is an issue with GoogleCharts when setting 'transparent' related to the zoom window sticking - once that's sorted out, the default will change back to 'transparent')
text	logical, whether or not to add text labels to classic boot plot. Default = FALSE.
min.prob	when text=TRUE, a lower bound on the probability of selection before a text label is shown.
srt	when text=TRUE, the angle of rotation for the text labels. Default = -30.
max.circle	circles are scaled to make largest dimension this size in inches. Default = 0.35.
print.full.model	logical, when text=TRUE this determines if the full model gets a label or not. Default=FALSE.
jitterk	amount of jittering of the model size in the lvk and boot plots. Default = 0.1.
...	further arguments (currently unused)

References

Mueller, S. and Welsh, A. H. (2010), On model selection curves. International Statistical Review, 78:240-256. doi: 10.1111/j.1751-5823.2010.00108.x

Murray, K., Heritier, S. and Mueller, S. (2013), Graphical tools for model selection in generalized linear models. Statistics in Medicine, 32:4438-4451. doi: 10.1002/sim.5855

See Also

vis

Examples

```
n = 100
set.seed(11)
e = rnorm(n)
x1 = rnorm(n)
x2 = rnorm(n)
x3 = x1^2
x4 = x2^2
x5 = x1*x2
y = 1 + x1 + x2 + e
dat = data.frame(y,x1,x2,x3,x4,x5)
lm1 = lm(y~.,data=dat)
## Not run:
v1 = vis(lm1)
plot(v1,highlight="x1",which="lvk")

## End(Not run)
```

print.af *Print method for an af object*

Description

Prints basic output of the bootstrap results of an af object.

Usage

```
## S3 method for class 'af'
print(x, best.only = TRUE, ...)
```

Arguments

x	an af object, the result of af
best.only	logical determining whether the output used the standard fence approach of only considering the best models that pass the fence (TRUE) or if it should take into account all models that pass the fence at each boundary value (FALSE).
...	further arguments (currently unused)

print.vis *Print method for a vis object*

Description

Prints basic output of the bootstrap results of an vis object.

Usage

```
## S3 method for class 'vis'
print(x, min.prob = 0.3, print.full.model = FALSE, ...)
```

Arguments

x	a vis object, the result of vis
min.prob	a lower bound on the probability of selection before the result is printed
print.full.model	logical, determines if the full model gets printed or not. Default=FALSE.
...	further arguments (currently unused)

summary.af *Summary method for an af object*

Description

Provides comprehensive output of the bootstrap results of an af object.

Usage

```
## S3 method for class 'af'
summary(object, best.only = TRUE, ...)
```

Arguments

object	af object, the result of af
best.only	logical determining whether the output used the standard fence approach of only considering the best models that pass the fence (TRUE) or if it should take into account all models that pass the fence at each boundary value (FALSE).
...	further arguments (currently unused)

vis *Model stability curves and variable inclusion plots*

Description

Calculates and provides the plot methods for standard and bootstrap enhanced model stability curves (lvk and boot) as well as variable inclusion plots (vip).

Usage

```
vis(mf, nvmax, B = 100, lambda.max, nbest = 5, n.cores, force.in = NULL,
  screen = FALSE, redundant = TRUE, ...)
```

Arguments

mf	a fitted 'full' model, the result of a call to lm or glm (and in the future lme or lmer)
nvmax	size of the largest model that can still be considered as a viable candidate
B	number of bootstrap replications
lambda.max	maximum penalty value for the vip plot, defaults to 2*log(n)
nbest	maximum number of models at each model size that will be considered for the lvk plot
n.cores	number of cores to be used when parallel processing the bootstrap

force.in	the names of variables that should be forced into all estimated models. (Not yet implemented.)
screen	logical, whether or not to perform an initial screen for outliers. Highly experimental, use at own risk. Default = FALSE.
redundant	logical, whether or not to add a redundant variable. Default = TRUE.
...	further arguments (currently unused)

Details

The result of this function is essentially just a list. The supplied plot method provides a way to visualise the results.

References

Mueller, S. and Welsh, A. H. (2010), On model selection curves. International Statistical Review, 78:240-256. doi: 10.1111/j.1751-5823.2010.00108.x

Murray, K., Heritier, S. and Mueller, S. (2013), Graphical tools for model selection in generalized linear models. Statistics in Medicine, 32:4438-4451. doi: 10.1002/sim.5855

See Also

plot.vis

Examples

```
n = 100
set.seed(11)
e = rnorm(n)
x1 = rnorm(n)
x2 = rnorm(n)
x3 = x1^2
x4 = x2^2
x5 = x1*x2
y = 1 + x1 + x2 + e
dat = data.frame(y,x1,x2,x3,x4,x5)
lm1 = lm(y~.,data=dat)
## Not run:
v1 = vis(lm1)
plot(v1,highlight="x1")

## End(Not run)
```

Index