Package 'heatmapFit'

June 6, 2016

Title Fit Statistic for Binary Dependent Variable Models

Version 2.0.4

Maintainer Justin Esarey <justin@justinesarey.com>

Description Generates a fit plot for diagnosing misspecification in models of binary dependent variables, and calculates the related heatmap fit statistic described in Esarey and Pierce (2012) <DOI:10.1093/pan/mps026>.

Depends R (>= 3.1.1), stats, graphics, grDevices

Imports utils

License GPL (>= 2)

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Author Justin Esarey [aut, cre],
Andrew Pierce [aut],
Jericho Du [aut]

Repository CRAN

Date/Publication 2016-06-06 20:36:28

R topics documented:

`heatmap.compress` *Collapse a large data set for* `heatmap.fit`

Description

Reduces the size of large binary data sets by binning them according to their predicted probability [0, 1].

Usage

```
heatmap.compress(y, pred, init.grid)
```

Arguments

`y`	A vector of observations of the dependent variable (in {0,1}).
`pred`	A vector of predicted Pr(y = 1) corresponding to each element of `y`.
`init.grid`	The number of bins on the interval [0, 1] to use for compression of `pred`.

Value

A list with the elements:

`y.out`	The value of `y`, 0 or 1.
`pred.out`	The (binned) predicted Pr(y = 1) matching each observation.
`weight.out`	A weight parameter indicating the proportion of observations in the bin; sums to one.
`pred.total.out`	A vector of unique Pr(y = 1) bin values.
`n.out`	The number of observations (non-empty bins) after the data are collapsed.
`retained.obs`	A vector of indices for non-empty candidate bins (for internal use by `heatmap.fit`).

Author(s)

Justin Esarey <justin@justinesarey.com>

heatmap.fit *Heatmap Fit Statistic for Binary Dependent Variable Models*

Description

Generates a fit plot for diagnosing misspecification in models of binary dependent variables, and calculates the related heatmap fit statistic (Esarey and Pierce, 2012).

Usage

```
heatmap.fit(y, pred, calc.boot = TRUE, reps = 1000, span.l = "aicc",
  color = FALSE, compress.obs = TRUE, init.grid = 2000, ret.obs = FALSE,
  legend = TRUE)
```

Arguments

y	A vector of observations of the dependent variable (in {0,1}).
pred	A vector of model-predicted Pr(y = 1) corresponding to each element of y.
calc.boot	Calculate bootstrap-based p-values (default = TRUE) or not (= FALSE).
reps	Number of bootstrap replicates to generate (default = 1000).
span.l	Bandwidth for the nonparametric fit between y and pred. Defaults to "aicc", calculation of an AICc-minimizing bandwidth. Other options are "gcv", which minimizes the generalized cross-validation statistic, or a numerical bandwidth.
color	Whether the plot should be in color (TRUE) or grayscale (the default, FALSE).
compress.obs	Whether large data sets should be compressed by pre-binning to save computing time (default TRUE). When true, only data sets larger than 10,000 observations will be compressed.
init.grid	If compress.obs = TRUE, the number of bins on the interval [0, 1] to use for compression of pred.
ret.obs	Return the one-tailed bootstrap p-value for each observation in y (TRUE) or not (the default, FALSE).
legend	Print the legend on the heat map plot (the default, TRUE) or not (FALSE).

Details

This function plots the degree to which a binary dependent variable (BDV) model generates predicted probabilities that are an accurate match for observed empirical probabilities of the BDV, in-sample or out-of-sample. For example, if a model predicts that Pr(y = 1) = k%, about k% of observations with this predicted probability should have y = 1. Loess smoothing (with an automatically-selected optimum bandwidth) is used to estimate empirical probabilities in the data set and to overcome sparseness of the data. Systematic deviations are distinguished from sampling variation via bootstrapping of the distribution under the null that the model is an accurate predictor, with p-values indicating the one-tailed proportion of bootstrap samples that are less-extreme than the observed deviation. The plot shows model predicted probabilities on the x-axis and smoothed

empirical probabilities on the y-axis, with a histogram indicating the location and frequency of observations. The ideal fit is a 45-degree line. The shading of the plotted line indicates the degree to which fit deviations are larger than expected due to sampling variation.

A summary statistic for fit (the "heatmap statistic") is also reported. This statistic is the proportion of the sample in a region with one-tailed p-value less than or equal to 10%. Finding more than 20% of the dataset with this p-value in this region is diagnostic of misspecification in the model.

More details for the technique are given in Esarey and Pierce 2012, "Assessing Fit Quality and Testing for Misspecification in Binary Dependent Variable Models," *Political Analysis* 20(4): 480-500.

Value

If `ret.obs = T`, a list with the element:

`heatmap.obs.p` The one-tailed bootstrap p-value corresponding to each observation in `y`.

Note

Code to calculate AICc and GCV written by Michael Friendly (http://tolstoy.newcastle.edu.au/R/help/05/11/15899.html).

Author(s)

Justin Esarey <justin@justinesarey.com>

Andrew Pierce <awpierc@emory.edu>

Jericho Du <jericho.du@gmail.com>

References

Esarey, Justin and Andrew Pierce (2012). "Assessing Fit Quality and Testing for Misspecification in Binary Dependent Variable Models." *Political Analysis* 20(4): 480-500. DOI:10.1093/pan/mps026.

Examples

```
## Not run:
## a correctly specified model
################################

set.seed(123456)
x <- runif(20000)
y <- as.numeric( runif(20000) < pnorm(2*x - 1) )
mod <- glm( y ~ x, family=binomial(link="probit"))
pred <- predict(mod, type="response")

heatmap.fit(y, pred, reps=1000)

## out-of-sample prediction w/o bootstrap p-values

set.seed(654321)
x <- runif(1000)
y <- as.numeric( runif(1000) < pnorm(2*x - 1) )
```

```
pred <- predict(mod, type="response", newdata=data.frame(x))

heatmap.fit(y, pred, calc.boot=FALSE)



## a misspecified model
########################

set.seed(13579)
x <- runif(20000)
y <- as.numeric( runif(20000) < pnorm(sin(10*x)) )
mod <- glm( y ~ x, family=binomial(link="probit"))
pred <- predict(mod, type="response")
heatmap.fit(y, pred, reps=1000)

## Comparison with and without data compression

system.time(heatmap.fit(y, pred, reps=100))
system.time(heatmap.fit(y, pred, reps=100, compress.obs=FALSE))

## End(Not run)
```

---

heatmapFit *heatmapFit.*

Description

heatmapFit.

Index