Package ‘smoothic’

June 13, 2023

Type Package
Title Variable Selection Using a Smooth Information Criterion
Version 1.1.0
Depends R (>= 3.5.0)
Maintainer Meadbh O'Neill <meadbh.oneill@ul.ie>
Description Implementation of the SIC epsilon-telescope method, either using single or distributional (multi-parameter) regression. Includes classical regression with normally distributed errors and robust regression, where the errors are from the Laplace distribution. The "smooth generalized normal distribution" is used, where the estimation of an additional shape parameter allows the user to move smoothly between both types of regression. See O'Neill and Burke (2022) "Robust Distributional Regression with Automatic Variable Selection" for more details. <arXiv:2212.07317>. This package also contains the data analyses from O'Neill and Burke (2023). "Variable selection using a smooth information criterion for distributional regression models". <doi:10.1007/s11222-023-10204-8>.
License GPL-3
URL https://github.com/meadbh-oneill/smoothic,
       https://meadbh-oneill.ie/smoothic/
Encoding UTF-8
LazyData true
RoxygenNote 7.2.2
Imports data.table, dplyr, ggplot2, MASS, numDeriv, purrr, rlang, tibble, tidyr
Suggests knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Author Meadbh O'Neill [aut, cre],
       Kevin Burke [aut]
Repository CRAN
Date/Publication 2023-06-13 08:40:02 UTC
**R topics documented:**

- bostonhouseprice ....................................................... 2
- bostonhouseprice2 ..................................................... 3
- diabetes ................................................................. 4
- pcancer ................................................................. 5
- plot_effects ............................................................ 6
- plot_paths .............................................................. 7
- predict.smoothic ....................................................... 8
- smoothic ............................................................... 9
- sniffer ................................................................. 11
- summary.smoothic ..................................................... 12

**Index**

| bostonhouseprice | Boston House Price Data (Original) |

**Description**

Original data, which come from a study by Harrison Jr and Rubinfeld (1978), examining the association between median house prices in a particular community with various community characteristics. See bostonhouseprice2 for the corrected version, with additional variables.

**Usage**

bostonhouseprice

**Format**

A data frame with 506 rows and 9 variables:

- **crime**  crimes committed per capita
- **rooms**  average number of rooms per house
- **radial**  index of accessibility to radial highways
- **stratio**  average student-teacher ratio of schools in the community
- **lowstat**  percentage of the population that are "lower status"
- **linox**  log(annual average nitrogen oxide concentration (pphm))
- **lproptax**  log(property tax per $1000)
- **ldist**  log(weighted distances to five employment centres in the Boston region)
- **lprice**  log(median house price ($))

**Source**

https://CRAN.R-project.org/package=wooldridge
References


Description

Corrected data, which come from a study by Harrison Jr and Rubinfeld (1978), examining the association between median house prices in a particular community with various community characteristics. See `bostonhouseprice` for the original version.

Usage

bostonhouseprice2

Format

A data frame with 506 rows and 13 variables:

- **crim**  per capita crime rate by town
- **zn**  proportion of residential land zoned for lots over 25,000 sq.ft
- **indus**  proportion of non-retail business acres per town
- **rm**  average number of rooms per dwelling
- **age**  proportion of owner-occupied units built prior to 1940
- **rad**  index of accessibility to radial highways
- **ptratio**  pupil-teacher ratio by town
- **lnox**  log(nitric oxides concentration (parts per 10 million))
- **ldis**  log(weighted distances to five Boston employment centres)
- **ltax**  log(full-value property-tax rate per USD 10,000)
- **llstat**  log(percentage of lower status of the population)
- **chas**  Charles River dummy variable (=1 if tract bounds river; 0 otherwise)
- **lcmedv**  log(corrected median value of owner-occupied homes in USD 1000’s)

Source

[https://CRAN.R-project.org/package=mlbench](https://CRAN.R-project.org/package=mlbench)

References


**Diabetes Data**

**Description**

Data relating to a study of disease progression one year after baseline.

**Usage**

`diabetes`

**Format**

A data frame with 442 rows and 11 variables:

- **AGE** age of the patient
- **SEX** sex of the patient
- **BMI** body mass index of the patient
- **BP** blood pressure of the patient
- **S1** blood serum measurement 1
- **S2** blood serum measurement 2
- **S3** blood serum measurement 3
- **S4** blood serum measurement 4
- **S5** blood serum measurement 5
- **S6** blood serum measurement 6
- **Y** quantitative measure of disease progression one year after baseline

**Source**

[https://CRAN.R-project.org/package=lars](https://CRAN.R-project.org/package=lars)

**References**

Prostate Cancer Data

Description

Data, which come from a study by Stamey et al. (1989), examining the correlation between the level of prostate-specific antigen (PSA) and various clinical measures in men who were about to receive a radical prostatectomy.

Usage

pcancer

Format

A data frame with 97 rows and 9 variables:

- lcavol  log(cancer volume (cm^3))
- lweight log(prostate weight (g))
- age  age of the patient
- lbph log(amount of benign prostatic hyperplasia (cm^2))
- svi presence of seminal vesicle invasion (1=yes, 0=no)
- lcp log(capsular penetration (cm))
- gleason Gleason score
- pgg45 percentage of Gleason scores four of five
- lpsa log(PSA (ng/mL))

Source


References

plot_effects

Plot conditional density curves

Description

This function plots the model-based conditional density curves for different effect combinations. For example, take a particular covariate that is selected in the final model. The other selected covariates are fixed at their median values and then the plotted red and blue densities correspond to the modification of the chosen covariate as "low" (Q1, first quartile) and "high" (Q3, third quartile).

Usage

plot_effects(obj, what = "all", density_range)

Arguments

obj
An object of class "smoothic" which is the result of a call to smoothic.

what
The covariate effects to be plotted, default is what = "all". The user may supply a vector of covariate names to be plotted (only covariates selected in the final model can be plotted). Additionally, a "baseline" or "average" individual can be plotted by supplying what = "median".

density_range
Optional range for which the density curves should be plotted.

Value

A plot of the conditional density curves.

Author(s)

Meadhbh O’Neill

Examples

# Sniffer Data -------------------
# MPR Model ----
results <- smoothic(
  formula = y ~ .,
  data = sniffer,
  family = "normal",
  model = "mpr"
)
plot_effects(results)
plot_paths

Plot the epsilon-telescope coefficient paths

Description

This function plots the standardized coefficient values with respect to the epsilon-telescope for the location (and dispersion) components.

Usage

plot_paths(
  obj,
  log_scale_x = TRUE,
  log_scale_x_pretty = TRUE,
  facet_scales = "fixed"
)

Arguments

obj          An object of class "smoothic" which is the result of a call to smoothic.
log_scale_x  Default is log_scale_x = TRUE, which uses a log scale on the x-axis. If log_scale_x = FALSE, then the raw values of the epsilon-telescope are plotted.
log_scale_x_pretty
              Default is log_scale_x_pretty = TRUE, where the x-axis labels are "pretty". epsilon_1 and epsilon_T must be a number to the power of 10 for this to apply.
facet_scales Default is facet_scales = "fixed". This is supplied to facet_wrap.

Value

A plot of the standardized coefficient values through the epsilon-telescope.

Author(s)

Meadhbh O’Neill

Examples

# Sniffer Data --------------------
# MPR Model ----
results <- smoothic(
  formula = y ~ .,
  data = sniffer,
  family = "normal",
  model = "mpr"
)
plot_paths(results)
predict.smoothic  Predict smoothic

Description

predict method class “smoothic”

Usage

## S3 method for class 'smoothic'
predict(object, newdata, ...)

Arguments

object  an object of class “smoothic” which is the result of a call to smoothic.
newdata  new data object
...  further arguments passed to or from other methods.

Value

a matrix containing the predicted values for the location mu and scale s

Author(s)

Meadhbh O’Neill

Examples

# Sniffer Data --------------------
# MPR Model ----
results <- smoothic(
  formula = y ~ .,
  data = sniffer,
  family = "normal",
  model = "mpr"
)
predict(results)
**smoothic**

**Variable Selection Using a Smooth Information Criterion (SIC)**

**Description**

Implements the SIC \( \epsilon \)-telescope method, either using single or multi-parameter regression. Returns estimated coefficients, estimated standard errors (SEE) and the value of the penalized likelihood function. Note that the function will scale the predictors to have unit variance, however, the final estimates are converted back to their original scale.

**Usage**

```r
smoothic(
  formula,  
data,  
family = "sgnd",  
model = "mpr",  
lambda = "log(n)",  
epsilon_1 = 10,  
epsilon_T = 1e-04,  
steps_T = 100,  
zero_tol = 1e-05,  
max_it = 10000,  
optimizer = "nlm",  
kappa,  
tau,  
max_it_vec,  
stepmax_nlm
)
```

**Arguments**

- **formula**: An object of class "formula": a two-sided object with response on the left hand side and the model variables on the right hand side.
- **data**: A data frame containing the variables in the model; the data frame should be unstandardized.
- **family**: The family of the model, default is family = "sgnd" for the "Smooth Generalized Distribution" where the shape parameter kappa is also estimated. Classical regression with normally distributed errors is performed when family = "normal". If family = "laplace", this corresponds to a robust regression with errors from the Laplace distribution.
- **model**: The type of regression to be implemented, either model = "mpr" for multi-parameter regression, or model = "spr" for single parameter regression (i.e., classical normal linear regression). Defaults to model="mpr".
- **lambda**: Value of penalty tuning parameter. Suggested values are "log(n)" and "2" for the BIC and AIC respectively. Defaults to lambda ="log(n)" for the BIC case.
smoothic

epsilon_1  Starting value for $\epsilon$-telescope. Defaults to 10.
epsilon_T Final value for $\epsilon$-telescope. Defaults to 1e-04.
steps_T  Number of steps in $\epsilon$-telescope. Defaults to 100.
zero_tol  Coefficients below this value are treated as being zero. Defaults to 1e-05.
max_it Maximum number of iterations to be performed before the optimization is terminated. Defaults to 1e+04.

optimizer The optimization procedure to be used. Defaults to optimizer = "nlm", where the nlm function from the stats package is used. This tends to be more stable than the manually coded Newton-Raphson procedure that is used when optimizer = "manual".

kappa Optional user-supplied positive kappa value (> 0.2 to avoid computational issues) if family = "sgnd". If supplied, the shape parameter kappa will be fixed to this value in the optimization. If not supplied, kappa is estimated from the data.

tau Optional user-supplied positive smoothing parameter value in the "Smooth Generalized Normal Distribution" if family = "sgnd" or family = "laplace". If not supplied, then tau = "0.15". Smaller values of tau bring the approximation closer to the absolute value function, but this can cause the optimization to become unstable. Some issues with standard error calculation with smaller values of tau when using the Laplace distribution in the robust regression setting.

max_it_vec Optional vector of length steps_T that contains the maximum number of iterations to be performed in each $\epsilon$-telescope step. If not supplied, max_it is the maximum number of iterations performed at each step.

stepmax_nlm Optional maximum allowable scaled step length (positive scalar) to be passed to nlm if optimizer = "nlm". If not supplied, default values in nlm are used.

Value
A list with estimates and estimated standard errors.

- coefficients - vector of coefficients.
- see - vector of estimated standard errors.
- model - the matched type of model which is called.
- plike - value of the penalized likelihood function.
- kappa - value of the estimated/fixed shape parameter kappa if family = "sgnd".

Author(s)
Meadhbh O’Neill

References

**Examples**

```r
# Sniffer Data -----------------
# MPR Model ----
results <- smoothic(
  formula = y ~ .,
  data = sniffer,
  family = "normal",
  model = "mpr"
)
summary(results)
```

---

**sniffer**

<table>
<thead>
<tr>
<th>Sniffer Data</th>
</tr>
</thead>
</table>

**Description**

Data examining the factors that impact the amount of hydrocarbon vapour released when gasoline is pumped into a tank.

**Usage**

`sniffer`

**Format**

A data frame with 125 rows and 5 variables:

- `tanktemp` initial tank temperature (degrees F)
- `gastemp` temperature of the dispensed gasoline (degrees F)
- `tankpres` initial vapour pressure in the tank (psi)
- `gaspres` vapour pressure of the dispensed gasoline (psi)
- `y` hydrocarbons emitted (g)

**Source**

https://CRAN.R-project.org/package=alr4

**References**

Summary Smooth Information Criterion (SIC) Fits

**Description**

`summary` method class “smoothic”

**Usage**

```r
## S3 method for class 'smoothic'
summary(object, ...)  
```

**Arguments**

- `object` an object of class “smoothic” which is the result of a call to `smoothic`
- `...` further arguments passed to or from other methods.

**Value**

A list containing the following components:

- `model` - the matched model from the `smoothic` object.
- `coefmat` - a typical coefficient matrix whose columns are the estimated regression coefficients, estimated standard errors (SEE) and p-values.
- `plike` - value of the penalized likelihood function.

**Author(s)**

Meadhbh O’Neill

**Examples**

```r
# Sniffer Data -------------------
# MPR Model ----
results <- smoothic(
  formula = y ~ .,
  data = sniffer,
  family = "normal",
  model = "mpr"
)
summary(results)
```
Index

* datasets
  bostonhouseprice, 2
  bostonhouseprice2, 3
  diabetes, 4
  pcancer, 5
  sniffer, 11

bostonhouseprice, 2, 3
bostonhouseprice2, 2, 3
diabetes, 4

facet_wrap, 7
formula, 9

nlm, 10

pcancer, 5
plot_effects, 6
plot_paths, 7
predict.smoothic, 8
print.summary.smoothic
  (summary.smoothic), 12

smoothic, 6–8, 9, 12
sniffer, 11
summary.smoothic, 12