Package ‘ProSGPV’

October 12, 2022

Title Penalized Regression with Second-Generation P-Values

Version 1.0.0

Date 2021-08-06

Maintainer Yi Zuo <yi.zuo@vanderbilt.edu>

Description Implementation of penalized regression with second-generation p-values for variable selection. The algorithm can handle linear regression, GLM, and Cox regression. S3 methods print(), summary(), coef(), predict(), and plot() are available for the algorithm. Technical details can be found at Zuo et al. (2021) <doi:10.1080/00031305.2021.1946150>.

Depends R (>= 3.5.0), glmnet, brglm2

Imports MASS, survival

License GPL-3

Encoding UTF-8

URL https://github.com/zuoyi93/ProSGPV

BugReports https://github.com/zuoyi93/ProSGPV/issues

LazyData true

RoxygenNote 7.1.1

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation no

Author Yi Zuo [aut, cre] (<https://orcid.org/0000-0002-6643-8326>), Thomas Stewart [aut], Jeffrey Blume [aut]

Repository CRAN

Date/Publication 2021-08-06 21:40:02 UTC
R topics documented:

- coef.sgpv
- gen.sim.data
- get.candidate
- get.coef
- get.var
- gvif
- plot.sgpv
- predict.sgpv
- print.sgpv
- pro.sgpv
- spine
- summary.sgpv
- t.housing

Index

coef.sgpv coef.sgpv: Extract coefficients from the model fit

Description

S3 method coef for an S3 object of class sgpv

Usage

```r
## S3 method for class 'sgpv'
coef(object, ...)
```

Arguments

- `object` An sgpv object
- `...` Other coef arguments

Value

Coefficients in the OLS model

Examples

```r
# prepare the data
x <- t.housing[, -ncol(t.housing)]
y <- t.housing$V9

# run one-stage algorithm
out.sgpv <- pro.sgpv(x = x, y = y)
```
# get coefficients
ccoef(out.sgpv)

gen.sim.data

Description

This function can be used to generate autoregressive simulation data

Usage

gen.sim.data(
  n = 100,
  p = 50,
  s = 10,
  family = c("gaussian", "binomial", "poisson", "cox"),
  beta.min = 1,
  beta.max = 5,
  rho = 0,
  nu = 2,
  sig = 1,
  intercept = 0,
  scale = 2,
  shape = 1,
  rateC = 0.2
)

Arguments

n Number of observations. Default is 100.
p Number of explanatory variables. Default is 50.
s Number of true signals. It can only be an even number. Default is 10.
family A description of the error distribution and link function to be used in the model. It can take the value of \code{gaussian}, \code{binomial}, \code{poisson}, and \code{cox}. Default is \code{gaussian}.
beta.min The smallest effect size in absolute value. Default is 1.
beta.max The largest effect size in absolute value. Default is 5.
rho Autocorrelation level. A numerical value between -1 and 1. Default is 0.
nu Signal to noise ratio in linear regression. Default is 2.
sig Standard deviation in the design matrix. Default is 1.
intercept Intercept of the linear predictor in the GLM. Default is 0.
scale Scale parameter in the Weibull distribution. Default is 2.
shape Shape parameter in the Weibull distribution. Default is 1.
rateC Rate of censoring in the survival data. Default is 0.2.
Value

A list of following components:

- **X**  The generated explanatory variable matrix
- **Y**  A vector of outcome. If `family` is `cox`, a two-column object is returned where the first column is the time and the second column is status (0 is censoring and 1 is event)
- **index**  The indices of true signals
- **beta**  The true coefficient vector of length p

Examples

```r
# generate data for linear regression
data.linear <- gen.sim.data(n = 20, p = 10, s = 4)

# extract x
x <- data.linear[[1]]

# extract y
y <- data.linear[[2]]

# extract the indices of true signals
index <- data.linear[[3]]

# extract the true coefficient vector
true.beta <- data.linear[[4]]

# generate data for logistic regression
data.logistic <- gen.sim.data(n = 20, p = 10, s = 4, family = "binomial")

# extract x
x <- data.logistic[[1]]

# extract y
y <- data.logistic[[2]]

# extract the indices of true signals
index <- data.logistic[[3]]

# extract the true coefficient vector
true.beta <- data.logistic[[4]]
```

Description

Get the indices of the candidate set in the first stage
get.coef

Usage
get.candidate(xs, ys, family)

Arguments
xs Standardized independent variables
ys Standardized dependent variable
family A description of the error distribution and link function to be used in the model. It can take the value of \code{gaussian}, \code{binomial}, \code{poisson}, and \code{cox}.

Value
A list of following components:
candidate.index A vector of indices of selected variables in the candidate set
lambda The lambda selected by generalized information criterion

get.coef get.coef: Get coefficients at each lambda

Description
Get the coefficients and confidence intervals from regression at each lambda as well as the null bound in SGPVs

Usage
get.coef(xs, ys, lambda, lasso, family)

Arguments
xs Standardized design matrix
ys Standardized outcome
lambda lambda in the lasso
lasso An glmnet object
family A description of the error distribution and link function to be used in the model. It can take the value of \code{gaussian}, \code{binomial}, \code{poisson}, and \code{cox}.

Value
A vector that contains the point estimates, confidence intervals and the null bound
get.var

get.var: Get indices

Description
Get the indices of the variables selected by the algorithm

Usage
get.var(candidate.index, xs, ys, family, gvif)

Arguments
candidate.index
Indices of the candidate set
xs
Standardized independent variables
ys
Standardized dependent variable
family
A description of the error distribution and link function to be used in the model. It can take the value of \code{gaussian}, \code{binomial}, \code{poisson}, and \code{cox}.
gvif
A logical operator indicating whether a generalized variance inflation factor-adjusted null bound is used. Default is FALSE.

Value
A list of following components:

out.sg pv A vector of indices of selected variables
null.bound.p Null bound in the SGPV screening
pe Point estimates in the candidate set
lb Lower bounds of effect estimates in the candidate set
ub Upper bounds of effect estimates in the candidate set

gvif
gvif: Get GVIF for each variable

gvif

gvif: Get GVIF for each variable

Description

Usage
gvif(mod, family)
Arguments

mod  A model object with at least two explanatory variables
family  A description of the error distribution and link function to be used in the model.
        It can take the value of \code{gaussian}, \code{binomial}, \code{poisson},
        and \code{cox}.

Value

A vector of GVIF for each variable in the model

Description

S3 method \code{plot} for an object of class \code{sgpv}. When the two-stage algorithm is used, this function
plots the fully relaxed lasso solution path on the standardized scale and the final variable selection
results. When the one-stage algorithm is used, a histogram of all coefficients with selected effects
is shown.

Usage

## S3 method for class 'sgpv'
plot(x, lpv = 3, lambda.max = NULL, short.label = T, 

Arguments

x  An \code{sgpv} object
lpv  Lines per variable. It can take the value of 1 meaning that only the bound that is
      closest to the null will be plotted, or the value of 3 meaning that point estimates
      as well as 95% confidence interval will be plotted. Default is 3.
lambda.max  The maximum lambda on the plot. Default is NULL.
short.label  An indicator if a short label is used for each variable for better visualization.
             Default is \code{TRUE}
...  Other \code{plot} arguments

Examples

# prepare the data
x <- t.housing[, -ncol(t.housing)]
y <- t.housing$V9

# one-stage algorithm
out.sgpv.1 <- pro.sgpv(x = x, y = y, stage = 1)
# plot the selection result
plot(out.sgpv.1)

# two-stage algorithm
out.sgpv.2 <- pro.sgpv(x = x, y = y)

# plot the fully relaxed lasso solution path and final solution
plot(out.sgpv.2)

# zoom in a little bit
plot(out.sgpv.2, lambda.max = 0.01)

# only plot one confidence bound
plot(out.sgpv.2, lpv = 1, lambda.max = 0.01)

---

**predict.sgpv**

**predict.sgpv**: Prediction using the fitted model

---

**Description**

S3 method predict for an object of class sgpv

**Usage**

```r
## S3 method for class 'sgpv'
predict(object, newdata, type, ...)
```

**Arguments**

- **object**: An sgpv object
- **newdata**: Prediction data set
- **type**: The type of prediction required. Can take the value of link, response, and terms. Default is response.
- **...**: Other predict arguments

**Value**

Predicted values

**Examples**

```r
# prepare the data
x <- t.housing[, -ncol(t.housing)]
y <- t.housing$V9
```
```r
# run one-stage algorithm
out.sgpv <- pro.sgpv(x = x, y = y)
predict(out.sgpv)
```

---

### Description

S3 method print for an S3 object of class `sgpv`

### Usage

```r
## S3 method for class 'sgpv'
print(x, ...)
```

### Arguments

- `x` An `sgpv` object
- `...` Other `print` arguments

### Value

Variable selection results

### Examples

```r
# prepare the data
x <- t.housing[, -ncol(t.housing)]
y <- t.housing$V9

# run one-stage algorithm
out.sgpv.1 <- pro.sgpv(x = x, y = y, stage = 1)
out.sgpv.1
```
**pro.sgpv function**

**Description**
This function outputs the variable selection results from either one-stage algorithm or two-stage algorithm.

**Usage**

```r
pro.sgpv(
  x,  
  y,  
  stage = c(1, 2),  
  family = c("gaussian", "binomial", "poisson", "cox"),  
  gvim = F
)
```

**Arguments**

- **x** Independent variables, can be a matrix or a data.frame
- **y** Dependent variable, can be a vector or a column from a data.frame
- **stage** Algorithm indicator. 1 denotes the one-stage algorithm and 2 denotes the two-stage algorithm. Default is 2. When n is less than p, only the two-stage algorithm is available.
- **family** A description of the error distribution and link function to be used in the model. It can take the value of `gaussian`, `binomial`, `poisson`, and `cox`. Default is `gaussian`.
- **gvim** A logical operator indicating whether a generalized variance inflation factor-adjusted null bound is used. Default is FALSE. See Fox (1992) doi: 10.1080/01621459.1992.10475190 for more details on how to calculate GVIF

**Value**

A list of following components:

- **var.index** A vector of indices of selected variables
- **var.label** A vector of labels of selected variables
- **lambda** lambda selected by generalized information criterion in the two-stage algorithm. NULL for the one-stage algorithm
- **x** Input data x
- **y** Input data y
- **family** family from the input
- **stage** stage from the input
Spine data

Description

Lower back pain can be caused by a variety of problems with any parts of the complex, interconnected network of spinal muscles, nerves, bones, discs or tendons in the lumbar spine. This dataset contains 12 biomechanical attributes from 310 patients, of whom 100 are normal and 210 are abnormal (Disk Hernia or Spondylolisthesis). The goal is to differentiate the normal patients from the abnormal using those 12 variables.

Usage

spine
summary.sgpv

Format

- **pelvic_incidence**  pelvic incidence
- **pelvic_tilt**  pelvic tilt
- **lumbar_lordosis_angle**  lumbar lordosis angle
- **sacral_slope**  sacral slope
- **pelvic_radius**  pelvic radius
- **degree_spondylolisthesis**  degree of spondylolisthesis
- **pelvic_slope**  pelvic slope
- **direct_tilt**  direct tilt
- **thoracic_slope**  thoracic slope
- **cervical_tilt**  cervical tilt
- **sacrum_angle**  sacrum angle
- **scoliosis_slope**  scoliosis slope
- **outcome**  1 is abnormal (Disk Hernia or Spondylolisthesis) and 0 is normal

Source

http://archive.ics.uci.edu/ml/datasets/vertebral+column

---

**summary.sgpv**

**summary.sgpv:** Summary of the final model

---

Description

S3 method summary for an S3 object of class sgpv

Usage

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

Arguments

- `object`  An sgpv object
- `...`  Other arguments

Value

Summary of a model
Examples

```r
# prepare the data
x <- t.housing[, -ncol(t.housing)]
y <- t.housing$V9

# run one-stage algorithm
out.sgpv <- pro.sgpv(x = x, y = y)

# get regression summary
summary(out.sgpv)
```

t.housing

Tehran housing data

Description

A dataset containing Tehran housing data. The data set has 372 observations. There are 26 explanatory variables at baseline, including 7 project physical and financial features (V2-V8) and 19 economic variables and indices (V11-V29). The outcome (V9) is the sales price of a real estate single-family residential apartment.

Usage

t.housing

Format

- **V9** Actual sales price
- **V2** Total floor area of the building
- **V3** Lot area
- **V4** Total Preliminary estimated construction cost based on the prices at the beginning of the project
- **V5** Preliminary estimated construction cost based on the prices at the beginning of the project
- **V6** Equivalent preliminary estimated construction cost based on the prices at the beginning of the project in a selected base year
- **V7** Duration of construction
- **V8** Price of the unit at the beginning of the project per square meter
- **V11** The number of building permits issued
- **V12** Building services index for preselected base year
- **V13** Wholesale price index of building materials for the base year
- **V14** Total floor areas of building permits issued by the city/municipality
- **V15** Cumulative liquidity
- **V16** Private sector investment in new buildings
V17  Land price index for the base year
V18  The number of loans extended by banks in a time resolution
V19  The amount of loans extended by banks in a time resolution
V20  The interest rate for loan in a time resolution
V21  The average construction cost by private sector at the completion of construction
V22  The average cost of buildings by private sector at the beginning of construction
V23  Official exchange rate with respect to dollars
V24  Nonofficial (street market) exchange rate with respect to dollars
V25  Consumer price index (CPI) in the base year
V26  CPI of housing, water, fuel & power in the base year
V27  Stock market index
V28  Population of the city
V29  Gold price per ounce

Source

http://archive.ics.uci.edu/ml/datasets/Residential+Building+Data+Set
Index

*datasets*
  spine, 11
  t.housing, 13

coeff.sgv, 2
coeff.sgv(), 11

gen.sim.data, 3
get.candidate, 4
get.coef, 5
get.var, 6
gvif, 6

plot.sgv, 7
plot.sgv(), 11
predict.sgv, 8
predict.sgv(), 11
print.sgv, 9
print.sgv(), 11
pro.sgv, 10

spine, 11
summary.sgv, 12
summary.sgv(), 11

t.housing, 13