Package 'maclogp'

July 22, 2025

Title Measures of Uncertainty for Model Selection

Version 0.1.1

Maintainer Yuanyuan Li <yynli9696@gmail.com>

Description Following the common types of measures of uncertainty for parameter estimation, two measures of uncertainty were proposed for model selection, see Liu, Li and Jiang (2020) <doi:10.1007/s11749-020-00737-9>. The first measure is a kind of model confidence set that relates to the variation of model selection, called Mac. The second measure focuses on error of model selection, called LogP. They are all computed via bootstrapping. This package provides functions to compute these two measures. Furthermore, a similar model confidence set adapted from Bayesian Model Averaging can also be computed using this package.

License GPL (>= 3)

URL https://github.com/YuanyuanLi96/maclogp

BugReports https://github.com/YuanyuanLi96/maclogp/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1.9001

Imports BMA, plot.matrix, rlist, utils

NeedsCompilation no

Depends R (>= 3.5.0)

Author Yuanyuan Li [aut, cre],
Jiming Jiang [ths]

Repository CRAN

Date/Publication 2021-04-22 07:40:02 UTC

Contents

`bms` *Bayesian Model Confidence Set*

Description

This function allows you to obtain a bayesian model confidence set with approximate posterior model probability.

Usage

```
bms(data, alpha, eps = 1e-06)
```

Arguments

`data`	a list including
	x covariates matrix, of dimension nobs and nvars;each row is an observation vector.
	y response variable.
`alpha`	a vector of significance levels. The confidence levels are `1-alpha`. Default value is 0.05.
`eps`	toterance level in choosing models with total posteriors at least `1-alpha`. Default value is `1e-6`.

Value

Returns a list containing:

`models`	A list with one entry for each model. Each entry is an integer vector that specifies the columns of matrix `x` to be used as a regressor in that model. Models is ordered with decreasing posteriors.
`con_sets`	a list with with one entry for a `1-alpha` model confidence set. Each entry is an integer vector that specifies the models selected in this set. The model indexes used in `con_sets` are their orders in `models`.
`length_con`	lengths of confidence sets.
`probs_inorder`	Model posteriors in decreasing order.
`beta_ls`	a list with one entry for each model. Each entry is a vector of estimated coefficients for that model.

References

Liu, X., Li, Y. & Jiang, J.(2020). Simple measures of uncertainty for model selection. *TEST*, 1-20.

Raftery, Adrian E. (1995). Bayesian model selection in social research (with Discussion). *Sociological Methodology* 1995 (Peter V. Marsden, ed.), pp. 111-196.

See Also

bic.glm

Examples

```
n= 50
B= 100
p= 5
x = matrix(rnorm(n*p, mean=0, sd=1), n, p)
true_b = c(1:3, rep(0,p-3))
y = x%*% true_b+rnorm(n)
alpha=c(0.1,0.05,0.01)
data=list(x=x,y=y)
result=bms(data,alpha)
```

---

diabetes *Diabetes data*

Description

These data consist of observations on 442 patients, with the response of interest being a quantitative measure of disease progression one year after baseline. There are ten baseline variables and have been normalized to have mean 0 and Euclidean norm 1. The response variable has been centered (mean 0).

Usage

```
diabetes
```

Format

A data frame with 442 rows and 11 variables:

V1 age

V2 sex

V3 body-mass index

V4 average blood pressure

V5 blood serum measurement 1

V6 blood serum measurement 2

V7 blood serum measurement 3

V8 blood serum measurement 4

V9 blood serum measurement 5

V10 blood serum measurement 6

V11 disease progression

Source

https://web.stanford.edu/~hastie/Papers/LARS/diabetes.sdata.txt

References

Efron, Hastie, Johnstone and Tibshirani (2003), Least Angle Regression. *Annals of Statistics*.

MAC *Mac and LogP measure*

Description

This function allows you to obtain a model confidence set using Mac procedure and the LogP uncertainty measure for a selection method based on an information criterion.

Usage

```
MAC(models, data, B, alpha, method = "bic", delta = 1e-04, eps = 1e-06)
```

Arguments

models	A list with one entry for each model. Each entry is an integer vector that specifies the columns of matrix data$x to be used as a regressor in that model. An intercept will be fitted automatically.
data	a list including **x** covariates matrix, of dimension nobs and nvars;each row is an observation vector. **y** response variable.
B	number of bootstrap replicates to perform; Default value is 200.
alpha	a vector of significance levels. The confidence levels of the model confidence sets are 1-alpha. Default value is 0.05.
method	Information criterion. Users can choose from "bic", "aic". Default value is "bic".
delta	A small positive number added inside of LogP when the bootstrap probability of selected model is 1. Default value is 1e-4.
eps	toterance level in choosing models with total bootstrap probabilities at least 1-alpha. Default value is 1e-6.

Value

Returns an object of class "MAC". An object of class "MAC" is a list containing at least the following components:

`hat_M`	numeric index of selected model.
`con_sets`	a list with with one entry for a `1-alpha` model confidence set. Each entry is an integer vector that specifies the models selected in this set. The model indexes used in `con_sets` are their orders in `models`.
`length_con`	lengths of confidence sets.
`order`	Model indexes with increasing information scores based on original data.
`probs_inorder`	Bootstrap probabilities for the models in `order`.
`beta_ls`	a list with one entry for each model. Each entry is a vector of estimated coefficients based on original data for that model.
`hat_prob`	the Bootstrap probability for single selected model.
`hat_logp`	the LogP measure.

References

Liu, X., Li, Y. & Jiang, J.(2020). Simple measures of uncertainty for model selection. *TEST*, 1-20.

See Also

`plot_MAC`

Examples

```
set.seed(0)
n= 50
B= 100
p= 5
x = matrix(rnorm(n*p, mean=0, sd=1), n, p)
true_b = c(1:3, rep(0,p-3))
y = x%*% true_b+rnorm(n)
alpha=c(0.1,0.05,0.01)
data=list(x=x,y=y)
models=Models_gen(1:p)
result=MAC(models, data, B, alpha)
```

Models_gen *Generate all subset models*

Description

This function generates a list including all subset models given a vector of candidate predictors.

Usage

```
Models_gen(predictors)
```

Arguments

predictors	a vector including the indexes of all predictors, such as 1:p.

Value

Returns a list with one entry for each model. Each entry is an integer vector that specifies the columns of matrix x to be used as a regressor in that model.

See Also

combn, list.flatten

Examples

```
Models_gen(1:5)
```

---

plot_MAC *Visualize model confidence sets*

Description

This funcion generates a heat map for a given model confidence set. Each row represents a model in the confidence set, and colored cell represents the variables in that model.

Usage

```
plot_MAC(models, alpha, con_sets, p, xnames = NULL, color = "lightblue")
```

Arguments

models	A list with one entry for each model. Each entry is an integer vector that specifies the columns of matrix X without intercept to be used as a regressor in that model. Intercept will be fitted automatically for every model. such as 1:p.
alpha	Significance levels. The confidence levels for confidence sets are 1-alpha.
con_sets	a list with with one entry for a 1-alpha model confidence set. Each entry is an integer vector that specifies the models selected in this set. The model indexes used in con_sets are their orders in models.
p	the number of candidate variables.
xnames	variable names of all candidate variables. Default is 1:p.
color	the color that indicates a variable is selected. Default is "lightblue".

Value

Returns a logical matrix per confidence set with one row per model and one column per variable indicating whether that variable is in the model.

Generates a corresponding heat map per confidence set with one row per model and one column per variable indicating whether that variable is in the model. A cell in white means the variable is not in that model; a cell in user-specified color means the variable is in that model.

See Also

MAC

Examples

```
n= 50
B= 100
p= 5
x = matrix(rnorm(n*p, mean=0, sd=1), n, p)
true_b = c(1:3, rep(0,p-3))
y = x%*% true_b+rnorm(n)
alpha=c(0.1,0.05,0.01)
data=list(x=x,y=y)
models=Models_gen(1:p)
result=MAC(models, data, B, alpha)
plot_MAC(models, alpha, result$con_sets, p)
result2=bms(data, alpha)
plot_MAC(result2$models, alpha, result2$con_sets, p)
```

Index