Package ‘Ohit’

October 12, 2022

Type Package
Title OGA+HDIC+Trim and High-Dimensional Linear Regression Models
Version 1.0.0
Date 2017-09-06
Author Hai-Tang Chiou, Ching-Kang Ing, Tze Leung Lai
Maintainer Hai-Tang Chiou <htchiou1@gmail.com>
Imports stats
Description Ing and Lai (2011) <doi:10.5705/ss.2010.081> proposed a high-dimensional model selection procedure that comprises three steps: orthogonal greedy algorithm (OGA), high-dimensional information criterion (HDIC), and Trim. The first two steps, OGA and HDIC, are used to sequentially select input variables and determine stopping rules, respectively. The third step, Trim, is used to delete irrelevant variables remaining in the second step. This package aims at fitting a high-dimensional linear regression model via OGA+HDIC+Trim.
License GPL-2
Encoding UTF-8
RoxygenNote 6.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2017-09-06 12:01:26 UTC

R topics documented:

OGA ................................................................. 2
Ohit ............................................................... 3
predict_Ohit ..................................................... 5

Index 7
Orthogonal greedy algorithm

Description
Select valuables via orthogonal greedy algorithm (OGA).

Usage
OGA(X, y, Kn = NULL, c1 = 5)

Arguments
- **X**: Input matrix of \( n \) rows and \( p \) columns.
- **y**: Response vector of length \( n \).
- **Kn**: The number of OGA iterations. \( Kn \) must be a positive integer between 1 and \( p \). Default is \( Kn = \max(1, \min(\text{floor}(c1*\sqrt{n/\log(p)}), p)) \), where \( c1 \) is a tuning parameter.
- **c1**: The tuning parameter for the number of OGA iterations. Default is \( c1 = 5 \).

Value
- **n**: The number of observations.
- **p**: The number of input variables.
- **Kn**: The number of OGA iterations.
- **J_OGA**: The index set of \( Kn \) variables sequentially selected by OGA.

Author(s)
Hai-Tang Chiou, Ching-Kang Ing and Tze Leung Lai.

References

Examples
```r
# Example setup (Example 3 in Section 5 of Ing and Lai (2011))
n = 400
p = 4000
q = 10
beta_1q = c(3, 3.75, 4.5, 5.25, 6, 6.75, 7.5, 8.25, 9, 9.75)
b = sqrt(3/(4 * q))
x_relevant = matrix(rnorm(n * q), n, q)
d = matrix(rnorm(n * (p - q), 0, 0.5), n, p - q)
```
```
x_relevant_sum = apply(x_relevant, 1, sum)
x_irrelevant = apply(d, 2, function(a) a + b * x_relevant_sum)
X = cbind(x_relevant, x_irrelevant)
epsilon = rnorm(n)
y = as.vector((x_relevant %*% beta_1q) + epsilon)

# Select valuables via OGA
OGA(X, y)
```

**Ohit**  
*Fit a high-dimensional linear regression model via OGA+HDIC+Trim*

**Description**

The first step is to sequentially select input variables via orthogonal greedy algorithm (OGA). The second step is to determine the number of OGA iterations using high-dimensional information criterion (HDIC). The third step is to remove irrelevant variables remaining in the second step using HDIC.

**Usage**

```
Ohit(X, y, Kn = NULL, c1 = 5, HDIC_Type = "HDBIC", c2 = 2, c3 = 2.01,
     intercept = TRUE)
```

**Arguments**

- **X**  
  Input matrix of *n* rows and *p* columns.

- **y**  
  Response vector of length *n*.

- **Kn**  
  The number of OGA iterations. Kn must be a positive integer between 1 and *p*. Default is \( Kn = \max(1, \min(\text{floor}(c1*\sqrt{n/log(p)}), p)) \), where *c1* is a tuning parameter.

- **c1**  
  The tuning parameter for the number of OGA iterations. Default is *c1*=5.

- **HDIC_Type**  
  High-dimensional information criterion. The value must be "HDAIC", "HDBIC" or "HDHQ". The formula is \( n*\log(\text{rmse})+k_{use}*\omega_n*n*\log(p) \) where \( \text{rmse} \) is the residual mean squared error and \( k_{use} \) is the number of variables used to fit the model. For HDIC_Type="HDAIC", it is HDIC with \( \omega_n=c2 \). For HDIC_Type="HDBIC", it is HDIC with \( \omega_n=\log(n) \). For HDIC_Type="HDHQ", it is HDIC with \( \omega_n=c3*\log(\log(n)) \). Default is HDIC_Type="HDBIC".

- **c2**  
  The tuning parameter for HDIC_Type="HDAIC". Default is *c2*=2.

- **c3**  
  The tuning parameter for HDIC_Type="HDHQ". Default is *c3*=2.01.

- **intercept**  
  Should an intercept be fitted? Default is intercept=TRUE.
Value

- **n**: The number of observations.
- **p**: The number of input variables.
- **Kn**: The number of OGA iterations.
- **J_OGA**: The index set of Kn variables sequentially selected by OGA.
- **HDIC**: The HDIC values along the OGA path.
- **J_HDIC**: The index set of valuables determined by OGA+HDIC.
- **J_Trim**: The index set of valuables determined by OGA+HDIC+Trim.
- **betahat_HDIC**: The estimated regression coefficients of the model determined by OGA+HDIC.
- **betahat_Trim**: The estimated regression coefficients of the model determined by OGA+HDIC+Trim.

Author(s)

Hai-Tang Chiou, Ching-Kang Ing and Tze Leung Lai.

References


Examples

```r
# Example setup (Example 3 in Section 5 of Ing and Lai (2011))
n = 400
p = 4000
q = 10
beta_1q = c(3, 3.75, 4.5, 5.25, 6, 6.75, 7.5, 8.25, 9, 9.75)
b = sqrt(3/(4 * q))
x_relevant = matrix(rnorm(n * q), n, q)
d = matrix(rnorm(n * (p - q), 0, 0.5), n, p - q)
x_relevant_sum = apply(x_relevant, 1, sum)
x_irrelevant = apply(d, 2, function(a) a + b * x_relevant_sum)
X = cbind(x_relevant, x_irrelevant)
epsilon = rnorm(n)
y = as.vector((x_relevant %*% beta_1q) + epsilon)

# Fit a high-dimensional linear regression model via OGA+HDIC+Trim
Ohit(X, y, intercept = FALSE)
```
**predict_Ohit**

Make predictions based on a fitted "Ohit" object

**Description**

This function returns predictions from a fitted "Ohit" object.

**Usage**

predict_Ohit(object, newX)

**Arguments**

- **object**: Fitted "Ohit" model object.
- **newX**: Matrix of new values for X at which predictions are to be made.

**Value**

- **pred_HDIC**: The predicted value based on the model determined by OGA+HDIC.
- **pred_Trim**: The predicted value based on the model determined by OGA+HDIC+Trim.

**Author(s)**

Hai-Tang Chiou, Ching-Kang Ing and Tze Leung Lai.

**References**


**Examples**

```r
# Example setup (Example 3 in Section 5 of Ing and Lai (2011))
n = 410
p = 4000
q = 10
beta_1q = c(3, 3.75, 4.5, 5.25, 6, 6.75, 7.5, 8.25, 9, 9.75)
b = sqrt(3/(4 * q))
x_relevant = matrix(rnorm(n * q), n, q)
d = matrix(rnorm(n * (p - q), 0, 0.5), n, p - q)
x_relevant_sum = apply(x_relevant, 1, sum)
x_irrelevant = apply(d, 2, function(a) a + b * x_relevant_sum)
X = cbind(x_relevant, x_irrelevant)
epsilon = rnorm(n)
y = as.vector((x_relevant %*% beta_1q) + epsilon)

# with intercept
fit1 = Ohit(X[1:400, ], y[1:400])
```
predict_Ohit(fit1, rbind(X[401:401, ]))
predict_Ohit(fit1, X[401:410, ])
# without intercept
fit2 = Ohit(X[1:400, ], y[1:400], intercept = FALSE)
predict_Ohit(fit2, rbind(X[401:401, ]))
predict_Ohit(fit2, X[401:410, ])
Index

OGA, 2
Ohit, 3

predict_Ohit, 5