Package 'MAVE'

October 12, 2022

Type Package

Title Methods for Dimension Reduction

Version 1.3.11

Date 2021-03-03

Author Hang Weiqiang<E0010758@u.nus.edu>, Xia Yingcun<staxyc@nus.edu.sg>

Maintainer Hang Weiqiang<E0010758@u.nus.edu>

Description Functions for dimension reduction, using MAVE (Minimum Average Variance Estimation), OPG (Outer Product of Gradient) and KSIR (sliced inverse regression of kernel version). Methods for selecting the best dimension are also included. Xia (2002) <doi:10.1111/1467-9868.03411>; Xia (2007) <doi:10.1214/009053607000000352>; Wang (2008) <doi:10.1198/016214508000000418>.

License GPL (>= 2)

LazyData yes

Imports Rcpp (>= 0.11.0),stats,graphics,mda

Depends R (>= 3.1.0)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.1.1

Suggests knitr

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-03-02 18:00:02 UTC

R topics documented:

`coef.mave` *Directions of CS or CMS of given dimension*

Description

This function returns the basis matrix of CS or CMS of given dimension

Usage

```
## S3 method for class 'mave'
coef(object, dim, ...)

## S3 method for class 'mave.dim'
coef(object, dim = "dim.min", ...)
```

Arguments

`object`	the output of `mave` or the output of `mave.dim`
`dim`	the dimension of CS or CMS. The value of dim should be given when the class of the argument dr is mave. When the class of the argument dr is mave.dim and dim is not given, the function will return the basis matrix of CS or CMS of dimension selected by `mave.dim`. Note that the dimension should be > 0.
`...`	no use.

Value

dir the matrix of CS or CMS of given dimension

See Also

`mave.data` for obtaining the reduced data

Examples

```
x <- matrix(rnorm(400),100,4)
y <- x[,1]+x[,2]+as.matrix(rnorm(100))
dr <- mave(y~x)
dir3 <- coef(dr,3)

dr.dim <- mave.dim(dr)
dir3 <- coef(dr.dim,3)
dir.best <- coef(dr.dim)
```

`Concrete` *Concrete Compressive Strength Data Set*

Description

Concrete strength is very important in civil engineering and is a highly nonlinear function of age and ingredients. This dataset contains 1030 instances and there are 8 features relevant to concrete strength. The description of the varaibles are given below. The description is from https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength. Name – Data Type – Measurement – Description

Format

A data frame with 1030 rows and 8 covariate variables and 1 response variable

Details

Cement (component 1) – quantitative – kg in a m3 mixture – Input Variable

Blast Furnace Slag (component 2) – quantitative – kg in a m3 mixture – Input Variable

Fly Ash (component 3) – quantitative – kg in a m3 mixture – Input Variable

Water (component 4) – quantitative – kg in a m3 mixture – Input Variable

Superplasticizer (component 5) – quantitative – kg in a m3 mixture – Input Variable

Coarse Aggregate (component 6) – quantitative – kg in a m3 mixture – Input Variable

Fine Aggregate (component 7) – quantitative – kg in a m3 mixture – Input Variable

Age – quantitative – Day (1~365) – Input Variable

Concrete compressive strength – quantitative – MPa – Output Variable

Source

https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength

References

-Cheng Yeh, "Modeling of strength of high performance concrete using artificial neural networks," Cement and Concrete Research, Vol. 28, No. 12, pp. 1797-1808 (1998).

Examples

```
data(Concrete)
train = sample(1:1030)[1:500]
x.train = as.matrix(Concrete[train,1:8])
y.train = as.matrix(Concrete[train,9])
x.test  = as.matrix(Concrete[-train,1:8])
y.test  = as.matrix(Concrete[-train,9])

dr = mave.compute(x.train,y.train, method='meanopg',max.dim=8)
```

```
dr.dim = mave.dim(dr)
y.pred = predict(dr.dim,x.test)
#estimation error
mean((y.pred-y.test)^2)
```

---

kc_house_data *House price in King County, USA*

Description

A data set contains 21613 obervations with 19 features plus house price. The names of the columns are given below.

- id
- date: Date house was sold(String)
- price: Price of the sold house
- bedrooms: Numer of Bedrooms
- bathrooms: Numer of bathrooms
- sqft_living: Square footage of the living room
- sqrt_log: Square footage of the log
- floors: Total floors in the house
- waterfront: Whether the house has a view a waterfront(1: yes, 0: not)
- view: unknown
- condtion: Condition of the house
- grade: unknown
- sqft_above: Square footage of house apart from basement
- sqft_basement: Square footage of the basement
- yr_built: Built year
- yr_renovated: Year when the house was renovated
- zipcode: zipcode of the house
- lat: Latitude coordinate
- long Longitude coordinate
- sqft_living15: Living room area in 2015(implies some renovations)
- sqrt_lot15: Lot area in 2015(implies some renovations)

Format

A data frame with 21613 rows and 19 variables

Source

https://www.kaggle.com/harlfoxem/housesalesprediction

Examples

```
data(kc_house_data)
#convert date in string to date in numeric value
kc_house_data[,2]=sapply(kc_house_data[,2],as.double)
train = sample(1:21613)[1:1000]
x.train = as.matrix(kc_house_data[train,c(2,4:21)]) #exclude id, house price
y.train = as.matrix(kc_house_data[train,3]) # house price
x.test  = as.matrix(kc_house_data[-train,c(2,4:21)])
y.test  = as.matrix(kc_house_data[-train,3])
```

`mave` *Dimension reduction*

Description

This function provides several methods to estimate the central space or central mean space of y on x. It returns the matrix of central space or central mean space for different dimensions and contains other information used for dimension selection by `mave.dim`.

Usage

```
mave(
  formula,
  data,
  method = "CSOPG",
  max.dim = 10,
  screen = NULL,
  subset,
  na.action = na.fail
)

mave.compute(
  x,
  y,
  method = "CSOPG",
  max.dim = 10,
  screen = nrow(x)/log(nrow(x))
)
```

Arguments

`formula`	the model used in regression
`data`	the data
`method`	This parameter specify which method will be used in dimension reduction. It provides five methods, including "csMAVE","csOPG","meanOPG","meanMAVE","KSIR" by default, method = 'csOPG'

- 'meanOPG' and 'meanMAVE' estimate dimension reduction space for conditional mean
- 'csMAVE' and 'csOPG' estimate the central dimension reduction space
- 'KSIR' is a kernel version of sliced inverse regression (Li, 1991). It is fast, but with poor accuracy.

`max.dim` the maximum dimension of dimension reduction space. The default is 10. In practice, max.dim will be equal to min(max.dim,ncol(x),screen).

`screen` specify the number of variables retained after screening method. The default is n/log(n). When this number is smaller than max.dim, then max.dim will change to the value of screen

`subset` an optional vector specifying a subset of observations to be used in the fitting process.

`na.action` a function which indicates what should happen when the data contain NAs. The default is na.action, which wil stop calculations. If na.action is set to be na.omit, the incomplete cases will be removed.

`x` The n by p design matrix.

`y` The n by q respond matrix.

Value

dr is a list which contains:

- dir: dir[[d]] is the central space with d-dimension d = 1, 2, ..., p reduced direction of different dimensions
- y: the value of response
- idx: the index of variables which survives after screening
- max.dim: the largest dimensions of CS or CMS which have been calculated in mave function
- ky: parameter used for DIM for selection
- x: the original training data

References

Li K C. Sliced inverse regression for dimension reduction[J]. Journal of the American Statistical Association, 1991, 86(414): 316-327.

Xia Y, Tong H, Li W K, et al. An adaptive estimation of dimension reduction space[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2002, 64(3): 363-410.

Xia Y. A constructive approach to the estimation of dimension reduction directions[J]. The Annals of Statistics, 2007: 2654-2690.

Wang H, Xia Y. Sliced regression for dimension reduction[J]. Journal of the American Statistical Association, 2008, 103(482): 811-821.

See Also

`mave.dim` for dimension selection, `predict.mave` for prediction using the dimension reduction space, `coef.mave` for accessing the basis vectors of dimension reduction space of given dimension, `plot.mave` for plot method for mave class

Examples

```
x <- matrix(rnorm(400*5),400,5)
b1 <- matrix(c(1,1,0,0,0),5,1)
b2 <- matrix(c(0,0,1,1,0),5,1)
eps <- matrix(rnorm(400),400,1)
y <- x%*%b1 + (x%*%b2)*eps

#finding central space based on OPG method
#dr.csopg <- mave.compute(x,y, method = 'csopg')
#or
dr.csopg <- mave(y ~ x, method = 'csopg')

#dr.meanopg <- mave.compute(x,y, method = 'meanopg')
#or
dr.meanopg <- mave(y ~ x, method = 'meanopg')

#find central mean space based on ksir method
dr.ksir <- mave(y~x,method='ksir')
#or
#dr.ksir <- mave.compute(x,y,method='ksir')

#See more examples about screening and mutiple responses in the vignette
#Using screening for high dimensional data
#x <- matrix(rnorm(100*50),100,50)
#y1 = as.matrix(x[,1])+rnorm(100)*.2
#y2 = as.matrix(x[,2]+x[,3])*as.matrix(x[,1]+x[,5])+rnorm(100)*.2
#y = cbind(y1,y2)
#dr.sc = mave(y~x,method='CSOPG',max.dim=5,screen=20)
#dr.sc.dim = mave.dim(dr.sc)
#print the directions of central space with the selected variables
#dr.sc.dim$dir[[3]][dr.sc$idx,]
```

`mave.data` *The reduced data matrix*

Description

The function returns the reduced data matrix of the original data. The reduced data matrix is obtained by the original data multiplied by the dimension reduction directions of given dimension.

Usage

```
mave.data(dr, x, dim = NULL)
```

Arguments

`dr`	the object returned by `mave` or `mave.dim`
`x`	the original data matrix of p dimensions
`dim`	the dimension of the reduced data matrix.

See Also

coef.mave for obtaining the dimension reduction directions

Examples

```
x <- matrix(rnorm(400),100,4)
y <- x[,1]+x[,2]+as.matrix(rnorm(100))
dr <- mave(y~x)
x.reduced <- mave.data(dr,x,3)
```

---

mave.dim *Select best direction using cross-validation*

Description

This function selects the dimension of the central (mean) space based on the calculation of MAVE using cross-validation method.

Usage

```
mave.dim(dr, max.dim = 10)
```

Arguments

dr	the result of MAVE function
max.dim	the maximum dimension for cross-validation.

Value

dr.dim contains all information in dr plus cross-validation values of corresponding direction

- cv0 : the cross-validation value when the null model is used
- cv : the cross-validation value using dimension reduction directions of different dimensions
- dim.min : the dimension of minimum cross-validation value. Note that this value can be 0.

See Also

mave for computing the dimension reduction space, predict.mave.dim for prediction method of mave.dim class

Examples

```
x <- matrix(rnorm(400*5),400,5)
b1 <- matrix(c(1,1,0,0,0),5,1)
b2 <- matrix(c(0,0,1,1,0),5,1)
eps <- matrix(rnorm(400),400,1)
y <- x%*%b1 + (x%*%b2)*eps

#seleted dimension of central space
dr.cs <- mave(y~x,method='csmave')
dr.cs.dim <- mave.dim(dr.cs)

#seleted dimension of central mean space
dr.mean <- mave(y~x,method='meanmave')
dr.mean.dim <- mave.dim(dr.mean)
```

---

plot.mave *Plot of mave or mave.dim object*

Description

Plot the scatterplot of given dimension directions and reponse variables.

Usage

```
## S3 method for class 'mave'
plot(x, dim = 4, plot.method = pairs, ...)

## S3 method for class 'mave.dim'
plot(x, dim = "dim.min", plot.method = pairs, ...)
```

Arguments

x	the object returned by mave
dim	the dimension
plot.method	the method for plotting scatter plot. The default is 'pairs'
...	arguments passed to the plot.method.

See Also

mave for computing the dimension reduction space

Examples

```
x = matrix(rnorm(2000),400,5)
beta1 = as.matrix(c(1,1,0,0,0))
beta2 = as.matrix(c(0,0,1,1,0))
err = as.matrix(rnorm(400))
y = (x%*%beta1)^2+x%*%beta2+err
```

```
dr = mave(y~x, method = 'meanopg')
dr.dim = mave.dim(dr)
plot(dr,dim=3)
plot(dr.dim)
```

`predict.mave` *Make predictions based on the dimension reduction space*

Description

This method make predictions based the reduced dimension of data using `mars` function.

Usage

```
## S3 method for class 'mave'
predict(object, newx, dim, ...)

## S3 method for class 'mave.dim'
predict(object, newx, dim = "dim.min", ...)
```

Arguments

`object`	the object of class 'mave'
`newx`	Matrix of the new data to be predicted
`dim`	the dimension of central space or central mean space. The matrix of the original data will be multiplied by the matrix of dimension reduction directions of given dimension. Then the prediction will be made based on the data of given dimensions. The value of dim should be given when the class of the argument dr is mave. When the class of the argument dr is mave.dim and dim is not given, the function will return the basis matrix of CS or CMS of dimension selected by `mave.dim`
`...`	further arguments passed to `mars` function such as degree.

Value

the prediced response of the new data

See Also

`mave` for computing the dimension reduction space and `mave.dim` for estimating the dimension of the dimension reduction space

Examples

```
X = matrix(rnorm(10000),1000,10)
beta1 = as.matrix(c(1,1,1,1,0,0,0,0,0,0))
beta2 = as.matrix(c(0,0,0,1,1,1,1,1,0,0))
err = as.matrix(rnorm(1000))
Y = X%*%beta1+X%*%beta2+err

train = sample(1:1000)[1:500]
x.train = X[train,]
y.train = as.matrix(Y[train])
x.test = X[-train,]
y.test = as.matrix(Y[-train])

dr = mave(y.train~x.train, method = 'meanopg')

yp = predict(dr,x.test,dim=3,degree=2)
#mean error
mean((yp-y.test)^2)

dr.dim = mave.dim(dr)

yp = predict(dr.dim,x.test,degree=2)
#mean error
mean((yp-y.test)^2)
```

---

spam *4601 email record*

Description

A dataset containing 4601 record of email with 57 features. These features are the relative frequency of most commonly used phrases and punctions. The data of these features are recorded 1 to 57 columns of the spam data. The outcome is spam or email which is denoted as 1 or 0, recorded in the 58th column of the data.

Format

A data frame with 4601 rows and 57 variables

Examples

```
data(spam)
train = sample(1:4601)[1:1000]
x.train <- as.matrix(spam[train,1:57])
y.train <- as.matrix(spam[train,58])
x.test <- as.matrix(spam[-train,1:57])
y.test <- as.matrix(spam[-train,58])
```

```
x.train <- sqrt(x.train)
x.test <- sqrt(x.test)
```

Index