Package 'mbrdr'

August 17, 2021

Type Package

Title Model-Based Response Dimension Reduction

Version 1.0.8

Date 2021-08-01

Author Jae Keun Yoo

Maintainer Jae Keun Yoo <peter.yoo@ewha.ac.kr>

Description Functions for model-based response dimension reduction. Usual dimension reduction methods in multivariate regression focus on the reduction of predictors, not responses. The response dimension reduction is theoretically founded in Yoo and Cook (2008) <doi:10.1016/j.csda.2008.07.029>. Later, three model-based response dimension reduction approaches are proposed in Yoo (2016) <doi:10.1080/02331888.2017.1410152> and Yoo (2019) <doi:10.1016/j.jkss.2019.02.001>. The metho parametric ordinary least squares, but the model-based approaches are done through maximum likelihood estimation. For two model-based response dimension reduction methods called principal fitted response reduction and unstructured principal fitted response reduction, chi-squared tests are provided for determining the dimension of the response subspace.

License GPL (>= 2.0)

Depends R(>= 3.5.0)

Imports stats

Repository CRAN

LazyData yes

Encoding UTF-8

RoxygenNote 6.0.1

NeedsCompilation no

Date/Publication 2021-08-17 08:40:02 UTC

R topics documented:

`mbrdr` *Main function for model-based response dimension reduction regression*

Description

This is the main function in the mbrdr package. It creates objects of class mbrdr to estimate the response mean subspace and perform tests concerning its dimension. Several helper functions that require a mbrdr object can then be applied to the output from this function.

Usage

```
mbrdr (formula, data, subset, na.action = na.fail, weights, ...)

mbrdr.compute (y, x, weights, method = "upfrr", ...)
```

Arguments

`formula`	a two-sided formula like `cbind(y1,y2,y3,y4)~x1+x2+x3`, where the left-side variables are a matrix of the response variables, and the right-hand side variables represent the predictors. The left-hand side of the formula must be a matrix, since the package reduces the dimension of the responses variables.
`data`	an optional data frame containing the variables in the model. By default the variables are taken from the environment from which 'mbrdr' is called.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process.
`weights`	an optional vector of weights to be used where appropriate. In the context of dimension reduction methods, weights are used to obtain elliptical symmetry, not constant variance.
`na.action`	a function which indicates what should happen when the data contain 'NA's. The default is 'na.fail,' which will stop calculations. The option 'na.omit' is also permitted, but it may not work correctly when weights are used.
`x`	The design matrix. This will be computed from the formula by `dr` and then passed to `dr.compute`, or you can create it yourself.
`y`	The response vector or matrix
`method`	This character string specifies the method of fitting. The default is `"upfrr"`. The options include `"yc"`, `"prr"`, `"pfrr"`. Each method may have its own additional arguments, or its own defaults; see the details below for more information.
`...`	For `mbrdr`, all additional arguments passed to `mbrdr.compute`. For `mbrdr.compute`, additional arguments may be required for particular dimension reduction method. For example, `numdir` is the maximum number of directions to compute, with default equal to 4. Other methods may have other defaults.

Details

The general regression problem mainly focuses on studying $E(y|x)$, the conditional mean of a response y given a set of predictors x, where y is r-dimensional response variables with $rgeq2$ and

This function provides methods for estimating the response dimension subspace of a general regression problem. That is, we want to find a $r \times d$ matrix B of minimal rank d such that

$$E(y|x) = E(P(B)y|x)$$

, where P(B) is an orthogonal projections onto the column space of B. Both the dimension d and the subspace $P(B)$ are unknown. These methods make few assumptions.

For the methods `"yc"`, `"prr"`, `"pfrr"` and `"upfrr"`, B is estimated and returned. And, only for `"pfrr"` and `"upfrr"`, chi-squared test results for estimating d is provided.

Weights can be used, essentially to specify the relative frequency of each case in the data.

Value

mbrdr returns an object that inherits from mbrdr (the name of the type is the value of the `method` argument), with attributes:

`y`	The response matrix
`x`	The design matrix
`weights`	The weights used, normalized to add to n.
`cases`	Number of cases used.
`call`	The initial call to `mbrdr`.
`evectors`	The eigenvectors from kernel matrices to estimate B computed from each response dimension reduction methods. It is the estimate of B.
`evalues`	The eigenvalues corresponding to the eigenvectors.
`stats`	This is the dimension test statistics for $pfrr$ and `"upfrr"`. It is the cumulatative sum of the eigenvalues for `"yc"` and `"prr"`
`fx`	This returns the user-selection of fx for `"pfrr"` and `"upfrr"`.
`numdir`	The maximum number of directions to be found. The output value of numdir may be smaller than the input value.
`method`	the dimension reduction method used.

Author(s)

Jae Keun Yoo, <peter.yoo@ewhat.ac.kr>.

References

Yoo, JK. (2018). Response dimension reduction: model-based approach. *Statistics : A Journal of Theoretical and Applied Statistic*, 52, 409-425. `"prr"` and `"pfrr"`

Yoo, JK. (2019). Unstructured principal fitted response reduction in multivariate regression. *Journal of the Korean Statistical Society*, 48, 561-567. `"upfrr"`

Yoo, JK. and Cook, R. D. (2008), Response dimension reduction for the conditional mean in multivariate regression. *Statistics and Probability Letters*, 47, 381-389. `"yc"`.

Examples

```
data(mps)
# default fitting method is "upfrr"
s0 <- mbrdr(cbind(A4, B4, A6, B6)~AFDC+Attend+B+Enrol+HS+Minority+Mobility+Poverty+PTR, data=mps)
summary(s0)

# Refit, using different choice of fx.
summary(s1 <- update(s0, fx.choice=2))

# Refit again, using pfrr with fx.choice=2
summary(s2<-update(s1, method="pfrr", fx.choice=1))

# Refit, using prr, which does not require the choice of fx.
summary(s3<- update(s1,method="prr"))

# fit using Yoo-Cook method:
summary(s4 <- update(s1,method="yc"))
```

Index