Package ‘HyperCube’

June 10, 2015

Type Package

Title Hypercube Estimator

Version 0.1-0

Date 2015-06-08

Author Chi Po Choi, Amy T. Kim

Maintainer Chi Po Choi<cpchoi@ucdavis.edu>, Amy T. Kim <atykim@ucdavis.edu>

Description

The package ``HyperCube'' provides functions and methods for fitting linear model with hypercube estimator. This package is an implementation of the hypercube estimators introduced in [Beran, Rudolf. ``Hypercube estimators: Penalized least squares, submodel selection, and numerical stability.''' Computational Statistics & Data Analysis 71 (2014): 654-666.].

Depends R (>= 3.1.0)

Imports expm, Matrix, MASS, testthat

LazyData true

License GPL

Suggests knitr

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`canadian.earnings` *Canadian earnings*

Description

The canadian.earnings data frame has 205 pairs observations on Canadian workers from a 1971 Canadian Census Public Use Tape (Ullah, 1985).

Usage

```
canadian.earnings
```

Format

A data frame with 205 rows and 2 variables:

age age in years.

log.income logarithm of income.

Source

Ullah, A. (1985). Specification analysis of econometric models. Journal of Quantitative Economics, 2, 187-209

`dental` *The hardness of 120 dental fillings*

Description

A dataset containing the response measures the hardness of dental filling obtained by 5 Dentists using 8 Gold alloys and 3 Condensation methods. The objective of the experiment was to find a dental gold filling with greater hardness.

Usage

```
dental
```

Format

A data frame with 120 rows and 4 variables:

y response mesures the hardness of dental fillings

G the indice of 8 Gold alloys

C the indice of 3 Condensation methods

D the indice of 5 Dentists

Details

Seheult and Tukey (2001) analyzed a three-factor layout in which the response measures the hardness of dental fillings obtained by 5 Dentists (D) using 8 Gold alloys (G) and 3 Condensation methods (C). The objective of the experiment was to find a dental gold lling with greater hardness. Condensation, properly carried out, was known to increase the hardness of a filling. The three condensation techniques used in the experiment were: (1) electromalleting, in which blows are delivered mechanically at a steady frequency; (2) hand malleting, in which a small mallet is used to deliver blows; and (3) hand condensation. The reported hardness observations are each averages of ten measurements that are not available. It was reported anecdotally that dentist 5 appeared to be physically tired before the experiment.

Source

Seheult, A. H. and Tukey, J. W. (2001). Towards robust analysis of variance, Data Analysis from Statistical Foundations (eds. A. K. Mohammed and E. Saleh), 217-244, Nova Science Publishers, New York.

`diffMatrix` *Difference Matrix*

Description

The difference matrix described in equation (3.10) in Beran (2014).

Usage

```
diffMatrix(p, dth)
```

Arguments

`p`	number of coefficients
`dth`	order of difference matrix

Examples

```
p <- 10
D <- diffMatrix(p, 5)
D
```

`estRisk` *Estimate Risk*

Description

Estimate Risk

Usage

```
estRisk(X, y, A, estsig)
```

Arguments

`X`	design matrix
`y`	observation
`A`	hypercuber operator
`estsig`	estimated variance

Value

The estimated risk

References

Beran, Rudolf. "Hypercube estimators: Penalized least squares, submodel selection, and numerical stability." Computational Statistics & Data Analysis 71 (2014): 654-666.

`estSigma` *Estimate Variance*

Description

Estimate Variance

Usage

```
estSigma(mf)
```

Arguments

`mf`	model frame

Value

The estimated variance

References

Beran, Rudolf. "Hypercube estimators: Penalized least squares, submodel selection, and numerical stability." Computational Statistics & Data Analysis 71 (2014): 654-666.

`hypercube` *Hypercube Estimator Fits*

Description

The user interface for fitting linear model with Hypercbue Estimator.

Usage

```
hypercube(...)

## Default S3 method:
hypercube(X, y, V, ...)

## S3 method for class formula
hypercube(formula, data, V, ...)
```

Arguments

`...`	other optional arguments
`X`	design matrix
`y`	observation
`V`	sysmmetric matrix whose eigenvalues all lie in [0,1]
`formula`	formula to get estimate
`data`	data you want to analysis

Value

A object of the class "`hypercube`" containing the following:

coefficients Estimated coefficents $\hat{\beta}$ under hypercube estimator

fitted.values Fitted values ($\hat{\eta}$) under hypercube estimator

residuals residuals, $y - \hat{\eta}$

estsigma2 Estimated variance of ϵ. If the rank of the design matrix X is greater than the number of observation, the variance is estimated using least square regression. If the rank is equal to the number of observation, and if the model has more than one factor, the variance is estimated using additive submodel fit. If neither of the above cases, users need to provide their own estimated variance.

estrisk Esitmated risk, which involved the estimated variance.

V Symmtric matrix with all eigenvalues lie in [0,1]. The matrix V used the hypercube estimator.

modelframe The modelframe constructed from the argument formula and data.

Methods (by class)

- `default`: Default method for `hypercube`
- `formula`: Formula method for `hypercube`

Note

This package is still under development. For data with degree of freedom equal to zero, it may not be handled well. It is due the fact that it is not obvious to estimate the variance of the random error. Users need to provide estimated variance of random error in such cases.

References

Beran, Rudolf. "Hypercube estimators: Penalized least squares, submodel selection, and numerical stability." Computational Statistics & Data Analysis 71 (2014): 654-666.

See Also

hypercubeOptimization

Examples

```
## Example 1 in Beran (2014)
## Fitting Canadian earning data with Hypercube Estimator

# The number of age, p, in Example 1 in Beran (2014).
p <- length(unique(canadian.earnings[,1]))
# D_5 as in equation (3.10) in Beran (2014)
D <- diffMatrix(p, 5)
# The parametor nu in equation (3.11) in Beran (2014)
nu <- 100
# The matrix W in equation (3.11) in Beran (2014)
W <- nu * t(D) %*% D
# Convert W to V, as described in (1.6) in Beran (2014)
V <- plsW2V(W)
# The variable age should be considered as a factor
canadian.earnings[,"age"] <- factor(canadian.earnings[,"age"])
# Hyperpercube Estimator Fit
hcmod <- hypercube( log.income ~ age -1, data=canadian.earnings, V)

# Plot of data
plot(as.numeric(as.character(canadian.earnings$age)),
     canadian.earnings$log.income,
     xlab = "age", ylab = "log(income)")
# Plot of fitted line
lines(levels(canadian.earnings$age), hcmod$coefficients)


## Example 2 in Beran (2014)
## Fitting rat litter data

# Projection matrices as decribed in equation (5.4) in Beran (2014)
litter.proj <- projectCreate( ~ mother:infant -1, data = litter)
# If only additive effect is consider,
# take V = P1 + P2 + P3 (notation in equation (5.4) in Beran (2014))
component <- cbind(c(0,0), c(1,0), c(0,1))
V <- projectWeight(litter.proj, component = component)
# Hypercube Estimator Fit
hcmod <- hypercube( weight ~ mother:infant -1, data = litter, V)

# Estimated Risk
summary(hcmod)
```

```
## Hypercube Estimator with optimal risk
##
hcmodopt <- hypercubeOptimization( weight ~ mother:infant -1,
                                   data = litter
                                   )
# The optimal projection coefficient which minimizes the risk.
hcmodopt$projcoef

# The minimum risk
hcmodopt$estrisk

# The Hypercube Estimator fit with the V of the optimal projection.
summary(hcmodopt$$est)
```

`hypercubeEst` *Estimation on linear Mode for Hypercube*

Description

This function calculates the hypercube estimates. Users should use `hypercube` which is a better user interface.

Usage

```
hypercubeEst(X, y, V, ...)
```

Arguments

`X`	design matrix, or data-incidence matrix (as described Beran (2014))
`y`	response, or observation (as described Beran (2014))
`V`	sysmmetric matrix whose eigenvalues all lie in [0,1] (as in equation (1.3) in Beran (2014))
`...`	other optional arguments

Value

The function `hypercubeEst` returns a list containing the following:

coefficients Estimated coefficents β under hypercube estimator

fitted.values Fitted values (η) under hypercube estimator

residuals residuals, $y - \eta$

V Symmtric matrix with all eigenvalues lie in [0,1]. The matrix V used the hypercube estimator. (as in equation (1.3) in Beran (2014))

References

Beran, Rudolf. "Hypercube estimators: Penalized least squares, submodel selection, and numerical stability." Computational Statistics & Data Analysis 71 (2014): 654-666.

See Also

hypercube is the user interfaces of hypercube estimators. The matrix V can be constructed using the function projectCreate.

Examples

```
## The use of the function \code{hypercubeEst}.
mf <- model.frame(weight ~ mother:infant -1, data = litter)
X <- model.matrix(attr(mf, "terms"), mf)
y <- model.response(mf)
litter.proj <- projectCreate( ~ mother:infant -1, data = litter)
V <- projectWeight(litter.proj, weights = c(1,1,1,0))
hcmod <- hypercubeEst(X, y, V)
hcmod$coefficients
```

---

hypercubeOp *Hypercube Operator*

Description

Hypercube Operator

Usage

```
hypercubeOp(X, V)
```

Arguments

X	design matrix
V	sysmmetric matrix whose eigenvalues all lie in [0,1]

References

Beran, Rudolf. "Hypercube estimators: Penalized least squares, submodel selection, and numerical stability." Computational Statistics & Data Analysis 71 (2014): 654-666.

---

hypercubeOptimization *Hypercube Optimization*

Description

Find the projection coefficients which minimizing the esitmated risk

Usage

```
hypercubeOptimization(formula, data, sigma = NULL)
```

Arguments

`formula`	formula
`data`	data
`sigma`	estimated variance

Value

A list containing the following:

est a object of the classs "`hypercube`".

projcoef optimal coefficients for the projections.

estrisk the minimum estimated risk.

References

Beran, Rudolf. "Hypercube estimators: Penalized least squares, submodel selection, and numerical stability." Computational Statistics & Data Analysis 71 (2014): 654-666.

See Also

`hypercube`

Examples

```
hcmodopt <- hypercubeOptimization( weight ~ mother:infant -1, data = litter)
hcmodopt$projcoef #projction coefficients
hcmodopt$estrisk #estimated risk
```

`litter` *Weigth gain of 61 infant rat litters*

Description

A dataset containing the (average) wight gain of an infant rat litter when the infants in the litter are nursed by a rat foster-mother

Usage

```
litter
```

Format

A data frame with 61 rows and 3 variables:

weight the (averge) weight gain of an infant rat litter

mother the genotype of the foster-mother nursing the infants

infant the genotype of the infant litter

Details

The rat litter data treated by Scheffe (1959) form an unbalnced two-way layout Each response recorded is the average weight-gain of a rat litter when the infants in the litter are nursed by a rat foster-mother. Factor1 with four levels, is the genotype of the foster-moather. Factor2 with the same levels, in the genotype of the infant litter.

The response measured in the experiment is the (average) weight gain of an infant rat litter when the infants in the litter are nursed by a rat foster-mother. Factor 1 is the genotype of the foster-mother nursing the infants. Factor 2 is the genotype of the infant litter.

Source

D.W. Bailey, The Inheritance of Maternal Influences on the Growth of the Rat (1953).

`monkey` *Response of 5 different monkey-pairs*

Description

A dataset containing reports responses to a certain stimulus that were measured for 5 different monkey-pairs (the subjects) in 5 different periods under 5 different conditions

Usage

```
monkey
```

Format

A data frame with 25 rows and 4 variables:

cond the condition

monkeys the monkey pair

period the monkey pair

response responses to a certain stimulus

Source

Data from Query no. 113, edited by G.W. Sender, Biometrics, Vol. 11, 1955, p.112

`motor` *Accelations over time*

Description

A dataset containing 133 observation of motorsycle acceleration against time in a simulated motorcycle accident. The p = 277 possible observation times constitute the vector t = (1, 2, ... , 277) Accelerations were observed at only q < p of these equally spaced time, sometimes with replication.

Usage

```
motor
```

Format

A data frame with 133 rows and 2 variables:

t The time in milliseconds since impact

accel The recorded head acceleration (in g)

Source

Silverman, B.W. (1985) Some aspects of the spline smoothing approach to non-parametric curve fitting. Journal of the Royal Statistical Society, B, 47, 1-52.

`plsW2V` *Covert W matrix to V matrix*

Description

Convert W to V, as described in (1.6) in Beran (2014)

Usage

```
plsW2V(W)
```

Arguments

`W`	a matrix, penalized least square

Value

The function `plsW2V` returns a matrix V for the Hypercube Estimator.

Examples

```
# D_5 as in equation (3.10) in Beran (2014)
p <- 45
D <- diffMatrix(p, 5)
# The matrix W in equation (3.11) in Beran (2014)
W <- t(D) %*% D
# Convert W to V, as described in (1.6) in Beran (2014)
V <- plsW2V(W)
```

`predict.hypercube` *Predict method for Hypercube Estimator Fits*

Description

Predict method for Hypercube Estimator Fits

Usage

```
## S3 method for class hypercube
predict(object, newdata = NULL, ...)
```

Arguments

`object`	an object of the class "`hypercube`"
`newdata`	new data
`...`	other arguments

`print.hypercube` *Print methods for hypercube*

Description

Print methods for hypercube

Usage

```
## S3 method for class hypercube
print(x, ...)
```

Arguments

`x`	an object of class "`hypercube`"
`...`	other arguments

`print.summary.hypercube` *Print methods for summary.hypercube*

Description

Print methods for summary.hypercube

Usage

```
## S3 method for class summary.hypercube
print(x, ...)
```

Arguments

`x`	an object of class "`summary.hypercube`"
`...`	other arguments

`projectCreate` *Functions for object "`projection.hypercube`"*

Description

The standard ANOVA projections as described in equation (5.4) in Beran (2014).

Usage

```
projectCreate(formula, data)

projectSub(proj, component)

projectWeight(proj, component = NULL, weights = NULL)

projectFun(proj, component = NULL)
```

Arguments

`formula`	input formula without response variable. For example, "~ mother:infant -1".
`data`	input data
`proj`	an object of class "`projection.hypercube`".
`component`	a vector or a matrix to specify with components in the `proj` are used. See examples for how to use.
`weights`	The weights for the weighted sum of projection matrix. The order of the weights should be the same as the order of the component.

Details

The standard ANOVA projections as described in equation (5.4) in Beran (2014). The set of projections are sysmmetric, idempotent, mutually orthogonal matrix. The linear combinations of the projection matrices are used as the matrix V (equation (1.3) in Beran (2014)) of the hypercube estimators. Please provide the formula in the form, for example in the data "litter", "~ mother:infant -1". No responese and No intercept. Only the interaction of factors.

Value

The function `projectCreate` creates a object of the "`projection.hyercube`", which is a list containing all the projection matrices. Also, the object "`projection.hyercube`" contains the following attributes:

variables The names of the variables in the formula

nlevels The numbers of levels of the factor in each variable

dims The dimension of the projection matrices

component The names of the projection matrices stored in the objects

The function `projectSub` returns a object of "`projection.hypercube`", which contains only the projection matrices according to the argument `component`.

The function `projectWeight` returns a projection matrix which is the weigthed sum of the projection matrices according to `component` and `weights`.

The function `projectFun` returns a function which computes and returns a weighted sum of projection matrices according to `component` in the argument of `projectFun`. The argument of the returned function is the `weight`. See example for how to use.

Functions

- `projectSub`: Subset of object "`projection.hypercube`"
- `projectWeight`: Weighted sum of matrix in object "`projection.hypercube`"
- `projectFun`: Functions of object "`projection.hypercube`"

References

Beran, Rudolf. "Hypercube estimators: Penalized least squares, submodel selection, and numerical stability." Computational Statistics & Data Analysis 71 (2014): 654-666.

Examples

```
proj <- projectCreate( ~ mother:infant -1, data = litter)

## proj contains the projection matrices :
## proj[["mother0:infant0"]], proj[["mother1:infant0"]], ...
## To see all the names of the projection matrices:
attr(proj, "component")

## If only the additive effects are needed,
## i.e. "mother0:infant0", "mother1:infant0" and "mother0:infant1",
component <- cbind(c(0,0), c(1,0), c(0,1))
proj.sub <- projectSub(proj, component)
attr(proj.sub, "component")
```

```
## If we want
## P = 1 * proj[["mother0:infant0"]] + 0.5 * proj[["mother1:infant0"]]
component <- cbind(c(0,0), c(1,0), c(0,1))
weights <- c(1, 0.5, 0)
proj.weights <- projectWeight(proj, component, weights)

## Create a function for more weighted projection matrices.
component <- cbind(c(0,0), c(1,0), c(0,1))
proj.fun <- projectFun(proj, component)

## Same as proj.weights in above example
weights <- c(1, 0.5, 0)
proj.fun(weights)

## Use the projection matrices for hypercube estimator
V <- proj.fun(weights)
hcmod <- hypercube(weight ~ mother:infant -1, data = litter, V = V)
summary(hcmod)
```

`summary.hypercube` *Summary method for Hypercube Estimator Fits*

Description

Summary method for Hypercube Estimator Fits

Usage

```
## S3 method for class hypercube
summary(object, ...)
```

Arguments

`object`	an object of class "`hypercube`"
`...`	other arguments

`vineyard` *Vineyard*

Description

A dataset containing records the grape yield harvested in each row of a vinyard in three succeed years

Usage

```
vineyard
```

Format

A data frame with 52 rows and 4 variables:

row the vineyard row number

year1 reporting the harvest yield in first year

year2 reporting the harvest yield in second year

year3 reporting the harvest yield in third year

Source

Simonoff, Jeffrey S. Smoothing methods in statistics. Springer Science & Business Media, 2012.

Index