Package ‘seedCCA’

August 30, 2017

Type Package
Title Seeded Canonical Correlation Analysis
Version 1.0
Date 2017-08-01
Author Jae Keun Yoo, Bo-Young Kim
Maintainer Jae Keun Yoo <peter.yoo@ewha.ac.kr>
Description Functions for dimension reduction through the seeded canonical correlation analysis are provided. A classical canonical correlation analysis (CCA) is one of useful statistical methods in multivariate data analysis, but it is limited in use due to the matrix inversion for large p small n data. To overcome this, a seeded CCA has been proposed in Im, Gang and Yoo (2015) <DOI:10.1002/cem.2691>. The seeded CCA is a two-step procedure. The sets of variables are initially reduced by successively projecting \text{cov}(X,Y) or \text{cov}(Y,X) onto \text{cov}(X) and \text{cov}(Y), respectively, without loss of information on canonical correlation analysis, following Cook, Li and Chiaromonte (2007) <DOI:10.1093/biomet/asm038> and Lee and Yoo (2014) <DOI:10.1111/anzs.12057> and the canonical correlation is finalized with the initially-reduced two sets of variables.

License GPL (>= 2.0)
Depends R(>= 2.10.0)
Imports CCA, corpcor, stats, graphics
Repository CRAN
LazyData yes
Encoding UTF-8
RoxygenNote 6.0.1
NeedsCompilation no
Date/Publication 2017-08-30 15:19:27 UTC

R topics documented:

  cookie ................................................................. 2
  covplot ................................................................. 3
  finalCCA ............................................................... 4
Description

This data set contains measurements from quantitative NIR spectroscopy. The example studied arises from an experiment done to test the feasibility of NIR spectroscopy to measure the composition of biscuit dough pieces (formed but unbaked biscuits). Two similar sample sets were made up, with the standard recipe varied to provide a large range for each of the four constituents under investigation: fat, sucrose, dry flour, and water. The calculated percentages of these four ingredients represent the 4 responses. There are 40 samples in the calibration or training set (with sample 23 being an outlier) and a further 32 samples in the separate prediction or validation set (with example 21 considered as an outlier).

An NIR reflectance spectrum is available for each dough piece. The spectral data consist of 700 points measured from 1100 to 2498 nanometers (nm) in steps of 2 nm.

Usage

data(cookie)

Format

A data frame of dimension 72 x 704. The first 700 columns correspond to the NIR reflectance spectrum, the last four columns correspond to the four constituents fat, sucrose, dry flour, and water. The first 40 rows correspond to the calibration data, the last 32 rows correspond to the prediction data.

References

Please cite the following papers if you use this data set.


covplot

Examples

data(cookie) # load data
X<-as.matrix(cookie[,1:700]) # extract NIR spectra
Y<-as.matrix(cookie[,701:704]) # extract constituents
Xtrain<-X[1:40,] # extract training data
Ytrain<-Y[1:40,] # extract training data
Xtest<-X[41:72,] # extract test data
Ytest<-Y[41:72,] # extract test data

covplot

scree-ploting cov(first.set, second.set)

Description

Returns a scree-plot of the eigenvalues of cov(first.set, second.set) to select its first d largest eigenvectors.

Usage

covplot(first.set, second.set, mind=NULL)

Arguments

first.set numeric matrix (n * p), the first set of variables
second.set numeric matrix (n * r), the second set of variables
mind numeric, the number of the eigenvalues to show their cumulative percentages. The default is NULL, and then it is equal to min(p,r)

Value

eigenvalues the ordered eigenvalues of cov(X,Y)
cum.percent the cumulative percentages of the eigenvalues
num.evecs a vector of the numbers of the eigenvectors which forces the cumulative percentages bigger than 0.6, 0.7, 0.8, 0.9

Examples

data(cookie)
myseq<-seq(141,651,by=2)
X<-as.matrix(cookie[-c(23,61),myseq])
Y<-as.matrix(cookie[-c(23,61),701:704])
covplot(X, Y)
covplot(X, Y, mind=4)
**finalCCA**

*finalized CCA in seeded CCA*

**Description**

Returns the results of the finalized step in seeded CCA.

**Usage**

`finalCCA(first.ini.set, second.ini.set)`

**Arguments**

- `first.ini.set` numeric matrix (\(n \times d\)), the initially-CCAed first set of variables
- `second.ini.set` numeric matrix (\(n \times d\)), the initially-CCAed second set of variables

**Value**

- `xphi` the finalized canonical variates of the first set of variables
- `yphi` the finalized canonical variates of the second set of variables
- `xcoef` the estimated canonical coefficient matrix of the initially-CCAed first set of variables
- `ycoef` the estimated canonical coefficient matrix of the initially-CCAed second set of variables
- `eigen` the finalized canonical correlations

**Examples**

```r
########## data(cookie) ##########
data(cookie)
myseq <- seq(141, 651, by=2)
X <- as.matrix(cookie[-c(23,61), myseq])
Y <- as.matrix(cookie[-c(23,61), 701:784])
min.pr <- min( dim(X)[2], dim(Y)[2])
MX0 <- iniCCA(X, Y, u=4, num.d=min.pr)
ini.X <- X %*% MX0
finalCCA(ini.X, Y)

########## data(nutrimouse) ##########
data(nutrimouse)
Y<-as.matrix(nutrimouse$lipid)
X<-as.matrix(nutrimouse$gene)
MX0 <- iniCCA(X, Y, u=4, num.d=4)
MY0 <- iniCCA(Y, X, u=5, num.d=4)
ini.X <- X %*% MX0
ini.Y <- Y %*% MY0
finalCCA(ini.X, ini.Y)
```
iniCCA

**Description**

Returns the canonical coefficient matrices from the initialized step in seeded CCA. The initialized CCA is done only for the first set in its first argument. The "num.d" must be bigger than or equal to the dimension of the second set.

**Usage**

iniCCA(first.set, second.set, u, num.d)

**Arguments**

- **first.set**: numeric matrix (n * p), the first set of variables: this set of variables alone is reduced.
- **second.set**: numeric matrix (n * r), the second set of variables
- **u**: numeric, the terminating index of the projection
- **num.d**: numeric, the first "num.d" eigenvectors of cov(X,Y) to replace cov(X,Y), if min(p,r) relatively bigger than n. The "num.d" must be less than or equal to min(p,r).

**Value**

B the initialized CCAed X projected up to u

**Examples**

```r
########### data(cookie) ###########
data(cookie)
myseq<-seq(141,651,by=2)
X<-as.matrix(cookie[-c(23,61),myseq])
Y<-as.matrix(cookie[-c(23,61),701:704])
min.pr <- min( dim(X)[2], dim(Y)[2])
MX0 <- iniCCA(X, Y, u=4, num.d=min.pr)
ini.X <- X%*%MX0

########### data(nutrimouse) ###########
data(nutrimouse)
Y<-as.matrix(nutrimouse$lipid)
X<-as.matrix(nutrimouse$gene)
MX0 <- iniCCA(X, Y, u=4, num.d=4)
MY0 <- iniCCA(Y, X, u=5, num.d=4)
ini.X <- X %*% MX0
ini.Y <- Y %*% MY0
```
**Description**

The nutrimouse dataset comes from a nutrition study in the mouse. It was provided by Pascal Martin from the Toxicology and Pharmacology Laboratory (French National Institute for Agronomic Research).

**Usage**

data(nutrimouse)

**Format**

gene: data frame (40 * 120) with numerical variables
lipid: data frame (40 * 21) with numerical variables
diet: factor vector (40)
genotype: factor vector (40)

**Details**

Two sets of variables were measured on 40 mice:
expressions of 120 genes potentially involved in nutritional problems.
concentrations of 21 hepatic fatty acids: The 40 mice were distributed in a 2-factors experimental design (4 replicates).
Genotype (2-levels factor): wild-type and PPARalpha -/-
Diet (5-levels factor): Oils used for experimental diets preparation were corn and colza oils (50/50) for a reference diet (REF), hydrogenated coconut oil for a saturated fatty acid diet (COC), sunflower oil for an Omega6 fatty acid-rich diet (SUN), linseed oil for an Omega3-rich diet (LIN) and corn/colza/enriched fish oils for the FISH diet (43/43/14).

**References**


**Examples**

data(nutrimouse)
boxplot(nutrimouse$lipid)
**Pm**

*Projection of a seed matrix on to the column subspace of M with respect to Sx inner-product*

**Description**

The function returns a projection of a seed matrix on to the column subspace of M with respect to Sx inner-product.

**Usage**

```
Pm(M, Sx, seed)
```

**Arguments**

- **M**: numeric matrix (p * k), a basis matrix of the column space of M
- **Sx**: a inner-product matrix
- **seed**: seed matrix

**Examples**

```r
data(cookie)
myseq<-seq(141,651,by=2)
X<-as.matrix(cookie[-c(23,61),myseq])
Y<-as.matrix(cookie[-c(23,61),701:704])

## using cov(X,Y) as a seed matrix
seed <- cov(X,Y)
col.num <- dim(seed)[2]
M <- iniCCA(X, Y, u=4, num.d=col.num)
Sx <- cov(X)
Pm(M, Sx, seed)

## using the first 2 largest eigenvectors of cov(X,Y) as a seed matrix
seed2 <- svd(cov(X,Y))$u[,1:2]
M2 <- iniCCA(X, Y, u=4, num.d=2)
Pm(M, Sx, seed2)
```

**seedCCA**

*Seeded Canonical correlation analysis*

**Description**

seedCCA is to conduct CCA when the maximum of the dimensions of the two sets of variables is greater than the sample sizes.
Usage

seedCCA(first.set, second.set, u1=2, u2=2, case1=FALSE, num.d=4)

Arguments

- **first.set**: numeric matrix (n * p), the first set of variables
- **second.set**: numeric matrix (n * r), the second set of variables
- **u1**: the termination index of projections for the first set of variables
- **u2**: the termination index of projections for the second set of variables
- **case1**: logical If FALSE initialized CCA are conducted for both variables.
- **num.d**: numeric, the number of the "num.d" largest eigenvectors of cov(first.set, second.set) and cov(second.set, first.set), if case1=FALSE. The default value is equal to 4. This option does not work, if case1=TRUE.

Value

- **initialMX0**: the initialized canonical coefficient matrices of the first set of variables
- **initialMY0**: the initialized canonical coefficient matrices of the second set of variables
- **newX**: the initially-CCAed first set of variables
- **newY**: the initially-CCAed second set of variables
- **xcoef**: the estimated canonical coefficients for the first set of variables
- **ycoef**: the estimated canonical coefficients for the second set of variables
- **xcanvar**: the estimated canonical variates for the first set of variables
- **ycanvar**: the estimated canonical variates for the second set of variables
- **eigenvalue**: the two sets of canonical correlations

References


Examples

```r
# data(cookie) dataframe
data(cookie)
myseq<-seq(141,651,by=2)
X<-as.matrix(cookie[-c(23,61),myseq])
Y<-as.matrix(cookie[-c(23,61),701:704])
```
seeding

\[ \text{dim}(X) ; \text{dim}(Y) \]
\[ \text{selectu}(X, Y, \text{case1}=\text{TRUE}) \]
\[ \text{seedCCA}(X, Y, u1=2, \text{case1}=\text{TRUE}) \]

######## data(nutrimouse) ########
\[ \text{data(nutrimouse)} \]
\[ X<-\text{as.matrix(nutrimouse}$\text{gene} \]
\[ Y<-\text{as.matrix(nutrimouse}$\text{lipid} \]
\[ \text{dim}(X) ; \text{dim}(Y) \]
\[ \text{covplot}(X, Y, \text{mind}=10) \]
\[ \text{selectu}(X, Y, u=10, \text{num.d}=4) \]
\[ \text{seedCCA}(X, Y, u1=6, u2=5, \text{num.d}=4) \]

---

**seeding**

*Iterative projection of a seed matrix onto cov(X)*

**Description**

Returns iterative projection of a seed matrix on cov(X) up to k times.

**Usage**

\[
\text{seeding} (\text{seed}, \text{cov.x}, n, u=5) 
\]

**Arguments**

- **seed**: numeric matrix (p * d), a seed matrix
- **cov.x**: numeric matrix (p * p), cov(X)
- **n**: sample sizes
- **u**: maximum number of projection

**Value**

- **F.u_x**: n*Fu values of X
- **F.u_y**: n*Fu values of Y

**Examples**

\[
\text{data(cookie)} \\
\text{myseq<-seq(141,651,by=2)} \\
X<-\text{as.matrix(cookie}[-c(23,61),myseq]) \\
Y<-\text{as.matrix(cookie}[-c(23,61),701:704]) \\
\text{seed <- cov(X,Y)} \\
\text{cov.x <- cov(X)} \\
\text{seeding(seed, cov.x, n=dim(X)[1], u=4)}
\]
Function that guide the selection of the terminating index when using seedCCA function

Description

When you use the seedCCA function, you need to specify the terminating indexes, for the first and second sets of variables. The function provides the guidance for selecting the indexes for the first and second sets of variables. The function returns a plot of the increment in n*Fu against u. The plot contains a red line, which indicates that the increment is equal to 0.1. If "case1=TRUE", the n*F_u and screeplot for one to have a higher dimension between the first and second argument sets are reported. For example, suppose that the dimension of the first set is higher than the second set. Then selectu(X, Y, u=5, case1=TRUE) and selectu(Y, X, u=5, case1=TRUE) provide the equivalent scree plot for X, but the returned name of n*F_u for selectu(X, Y, u=5, case1=TRUE) is F.u1, but that for selectu(Y, X, u=5, case1=TRUE) is F.u2.

Usage

selectu(first.set, second.set, u=5, case1=FALSE, num.d=4)

Arguments

- first.set: numeric matrix (n * p), the first set of variables
- second.set: numeric matrix (n * r), the second set of variables
- u: numeric, the maximum number of u. The default value is equal to 5.
- case1: logical, if FALSE, initialized CCA are conducted for both variables. The default value is FALSE.
- num.d: numeric, the number of the "num.d" largest eigenvectors of cov(first.set, second.set), if case1=FALSE. The default value is equal to 4. This option does not work, if case1=TRUE.

Value

- F.u_x: n*Fu values of X
- F.u_y: n*Fu values of Y

Examples

```r
# data(cookie) #
data(cookie)
myseq<-seq(141,651,by=2)
X<-as.matrix(cookie[-c(23,61),myseq])
Y<-as.matrix(cookie[-c(23,61),701:704])
selectu(X, Y, case1=TRUE) ## nFu1 returned
selectu(Y, X, case1=TRUE) ## the same as selectu(X, Y, case1=TRUE), but nFu2 returned.
```
selectu(X, Y, u=10, case1=TRUE)

######## data(nutrimouse) ########
data(nutrimouse)
Y<-'as.matrix(nutrimouse$lipid)
X<-'as.matrix(nutrimouse$gene)
selectu(X, Y, u=10, num.d=4)
Index

cookie, 2

covplot, 3

finalCCA, 4

iniCCA, 5

nutrimouse, 6

Pm, 7

seedCCA, 7

seeding, 9

selectu, 10