Package ‘SELPCCA’

June 7, 2020

Type Package

Title Sparse Canonical Correlation Analysis for Associating Mutiple High Dimensional Data

Version 1.0

Author Haoyu Chen and Sandra E Safo

Maintainer Sandra E. Safo <ssafo@umn.edu>

Url https://www.sandraesafo.com/software

Description
Sparse canonical correlation analysis method to associate two high dimensional data types. The algorithm obtains linear combinations of subsets of variables for each data type that contribute to overall dependency structure between the data types.

License GPL (>=2.0)

Imports CVXR, doParallel, foreach

Encoding UTF-8

LazyData true

NeedsCompilation no

Depends R (>= 3.5.0)

R topics documented:

`cvselpscca` *Cross validation for Sparse Canonical Correlation Analysis*

Description

Peforms nfolds cross validation to select optimal tuning parameters for SELPCCA based on training data. If you want to apply optimal tuning parameters to testing data, you may also use multiplescca.

Usage

```
cvselpscca(Xdata1=Xdata1,Xdata2=Xdata2,ncancorr=ncancorr,CovStructure="Iden",
          isParallel=TRUE,ncores=NULL,nfolds=5,ngrid=10,
          standardize=TRUE,thresh=0.0001,maxiteration=20)
```

Arguments

`Xdata1` A matrix of size $n \times p$ for first dataset. Rows are samples and columns are variables.

`Xdata2` A matrix of size $n \times q$ for second dataset. Rows are samples and columns are variables.

`ncancorr` Number of canonical correlation vectors. Default is 1.

`CovStructure` Covariance structure to use in estimating sparse canonical correlation vectors. Either "Iden" or "Ridge". Iden assumes the covariance matrix for each dataset is identity. Ridge uses the sample covariance for each dataset. See reference article for more details.

`isParallel` TRUE or FALSE for parallel computing. Default is TRUE.

`ncores` Number of cores to be used for parallel computing. Only used if isParallel=TRUE. If isParallel=TRUE and ncores=NULL, defaults to half the size of the number of system cores.

`nfolds` Number of cross validation folds. Default is 5.

`ngrid` Number of grid points for tuning parameters. Default is 10 for each dataset.

`standardize` TRUE or FALSE. If TRUE, data will be normalized to have mean zero and variance one for each variable. Default is TRUE.

`maxiteration` Maximum iteration for the algorithm if not converged. Default is 20.

`thresh` Threshold for convergence. Default is 0.0001.

Details

The function will return several R objects, which can be assigned to a variable. To see the results, use the “$" operator.

Value

`hatalpha` Estimated sparse canonical correlation vectors for first dataset.

`hatbeta` Estimated sparse canonical correlation vectors for second dataset.

`CovStructure` Covariance structure used in estimating sparse canonical correlation vectors. Either "Iden" or "Ridge".

`optTau` Optimal tuning parameters for each dataset.

`maxcorr` Estimated canonical correlation coefficient.

`tunerange` Grid values for each dataset used for searching optimal tuning paramters.

References

Sandra E. Safo, Jeongyoun Ahn, Yongho Jeon, and Sungkyu Jung (2018) , *Sparse Generalized Eigenvalue Problem with Application to Canonical Correlation Analysis for Integrative Analysis of Methylation and Gene Expression Data. Biometrics*

See Also

multiplescca

Examples

```
library(SELPCCA)
##---- read in data
data(DataExample)

Xdata1=DataExample[[1]]
Xdata2=DataExample[[2]]


##---- call cross validation to estimate first canonical correlation vectors
ncancorr=1
mycv=cvselpscca(Xdata1=Xdata1,Xdata2=Xdata2,ncancorr=ncancorr,CovStructure="Iden",
                isParallel=FALSE,ncores=NULL,nfolds=5,ngrid=10,
                standardize=TRUE,thresh=0.0001,maxiteration=20)

#check output
train.correlation=mycv$maxcorr

optTau=mycv$optTau

hatalpha=mycv$hatalpha

hatbeta=mycv$hatbeta

#obtain correlation plot using training data
scoresX1=Xdata1%*% hatalpha
scoresX2=Xdata2%*% hatbeta
plot(scoresX1, scoresX2,lwd=3,
       ,xlab=paste(
         "First Canonical correlation variate for dataset", 1),
         ylab=paste("First Canonical correlation variate for dataset", 2),
      main=paste("Correlation plot for datasets",1, "and" ,2, ",", "\u03C1 =", mycv$maxcorr))


#obtain correlation plot using testing data

Xtestdata1=DataExample[[3]]
Xtestdata2=DataExample[[4]]
scoresX1=Xtestdata1%*%hatalpha
scoresX2=Xtestdata2%*%hatbeta
mytestcorr=round(abs(cor(Xtestdata1%*%hatalpha,Xtestdata2%*%hatbeta)),3)

plot(scoresX1, scoresX2,lwd=3,xlab=paste(
         "First Canonical correlation variate for dataset", 1),
         ylab=paste("First Canonical correlation variate for dataset", 2),
      main=paste("Correlation plot for datasets",1, "and" ,2, ",", "\u03C1 =", mytestcorr))
```

---

cvtunerange *Tuning parameter range*

Description

Obtain upper and lower bounds of tuning parameters for each canonical correlation vector. It is recommended to use cvselpscca to choose optimal tuning paramters for each dataset.

Usage

```
cvtunerange(Xdata1=Xdata1,Xdata2=Xdata2,ncancorr=ncancorr,
            CovStructure="Iden",standardize=TRUE)
```

Arguments

`Xdata1` A matrix of size $n \times p$ for first dataset. Rows are samples and columns are variables.

`Xdata2` A matrix of size $n \times q$ for second dataset. Rows are samples and columns are variables.

`ncancorr` Number of canonical correlation vectors. Default is one.

`CovStructure` Covariance structure to use in estimating sparse canonical correlation vectors. Either "Iden" or "Ridge". Iden assumes the covariance matrix for each dataset is identity. Ridge uses the sample covariance for each dataset. See reference article for more details.

`standardize` TRUE or FALSE. If TRUE, data will be normalized to have mean zero and variance one for each variable. Default is TRUE.

Details

The function will return tuning ranges for sparse estimation of canonical correlation vectors. To see the results, use the “$" operator.

Value

`TauX1range` A $ncancorr \times 2$ matrix of upper and lower bounds of tuning parameters for each canonical correlation vector for first dataset.

`TauX2range` A $ncancorr \times 2$ matrix Upper and lower bounds of tuning parameters for each canonical correlation vector for second dataset.

References

Sandra E. Safo, Jeongyoun Ahn, Yongho Jeon, and Sungkyu Jung (2018) , *Sparse Generalized Eigenvalue Problem with Application to Canonical Correlation Analysis for Integrative Analysis of Methylation and Gene Expression Data. Biometrics*

See Also

`cvselpscca`,`multiplescca`

Examples

```
#see example in multiplescca
```

DataExample *Simulated data with one true canonical correlation vectors.*

Description

Simulated data with one true canonical correlation vectors for first and second datasets. The first 20 and 15 variables are nonzero (i.e., signal variables) in the first canonical correlation vectors for the first and second datasets respectively.

Usage

```
data(DataExample)
```

Format

A list with 7 elements

Xdata1 A matrix of size 80×200 for first dataset. Rows are samples and columns are variables.

Xdata2 A matrix of size 80×150 for second dataset. Rows are samples and columns are variables.

Xtestdata1 A matrix of size 400×200 for first dataset. Rows are samples and columns are variables.

Xtestdata2 A matrix of size 400×150 for second dataset. Rows are samples and columns are variables.

TrueAlpha The first canonical correlation vector for Xdata1.

TrueBeta The first canonical correlation vector for Xdata2.

TrueCorr The first canonical correlation coefficient.

References

Sandra E. Safo, Jeongyoun Ahn, Yongho Jeon, and Sungkyu Jung (2018) , *Sparse Generalized Eigenvalue Problem with Application to Canonical Correlation Analysis for Integrative Analysis of Methylation and Gene Expression Data. Biometrics*

Examples

```
#see example in multiplescca or cvselpscca
```

`multiplescca` *Sparse canonical correlation vectors for fixed tuning paramters.*

Description

Obtain sparse canonical correlation vectors for fixed tuning parameters. It is recommended to use cvselpscca to choose optimal tuning paramters for each dataset, or use cvtunerange for range of tuning parameters.

Usage

```
multiplescca(Xdata1=Xdata1,Xdata2=Xdata2,ncancorr=ncancorr,Tau=Tau,
          CovStructure="Iden”,standardize=TRUE,maxiteration=20, thresh=0.0001)
```

Arguments

`Xdata1` A matrix of size $n \times p$ for first dataset. Rows are samples and columns are variables.

`Xdata2` A matrix of size $n \times q$ for second dataset. Rows are samples and columns are variables.

`ncancorr` Number of canonical correlation vectors. Default is one.

`Tau` A vector or matrix of fixed tuning parameters for each dataset.

`CovStructure` Covariance structure to use in estimating sparse canonical correlation vectors. Either "Iden" or "Ridge". Iden assumes the covariance matrix for each dataset is identity. Ridge uses the sample covariance for each dataset. See reference article for more details.

`standardize` TRUE or FALSE. If TRUE, data will be normalized to have mean zero and variance one for each variable. Default is TRUE.

`maxiteration` Maximum iteration for the algorithm if not converged. Default is 20.

`thresh` Threshold for convergence. Default is 0.0001.

Details

The function will return three R objects, which can be assigned to a variable. To see the results, use the “$" operator.

Value

`hatalpha` Estimated sparse canonical correlation vectors for first dataset.

`hatbeta` Estimated sparse canonical correlation vectors for second dataset.

`maxcorr` Estimated correlation from canonical correlation vectors.

References

Sandra E. Safo, Jeongyoun Ahn, Yongho Jeon, and Sungkyu Jung (2018) , *Sparse Generalized Eigenvalue Problem with Application to Canonical Correlation Analysis for Integrative Analysis of Methylation and Gene Expression Data. Biometrics*

See Also

cvselpscca,cvtunerange

Examples

```
library(SELPCCA)
##---- read in data
data(DataExample)

Xdata1=DataExample[[1]]
Xdata2=DataExample[[2]]

##---- estimate first canonical correlation vectors
ncancorr=1

#use cvtunerange for range of tuning parameters
mytunerange=cvtunerange(Xdata1=Xdata1,Xdata2=Xdata2,ncancorr=ncancorr,
                        CovStructure="Iden",standardize=TRUE)
print(mytunerange)

#Fix Tau for first and second datasets as 1.1 and 1.0 respectively
Tau=matrix(c(1,1.2,1),nrow=1)
mysparsevectors=multiplescca(Xdata1=Xdata1,Xdata2=Xdata2,ncancorr=ncancorr,
                            Tau=Tau, CovStructure="Iden",standardize=TRUE,
                            maxiteration=20, thresh=0.0001)


#example with two canonical correlation vectors
#use cvselpscca to obtain optimal tuning parameters
mycv=cvselpscca(Xdata1=Xdata1,Xdata2=Xdata2,ncancorr=ncancorr,
                CovStructure="Iden",isParallel=FALSE,ncores=NULL,nfolds=5,
                ngrid=10, standardize=TRUE,thresh=0.0001,maxiteration=20)


Tau=mycv$optTau
mysparsevectors=multiplescca(Xdata1=Xdata1,Xdata2=Xdata2,ncancorr=ncancorr,
                Tau=Tau, CovStructure="Iden",standardize=TRUE,maxiteration=20,
                thresh=0.0001)
```

Index