Package ‘SIDA’

June 5, 2020

Type Package

Title Sparse Integrative Discriminant Analysis for Multi-view Structured Data

Version 1.0

Author Sandra E. Safo, Eun Jeong Min, and Lillian Haine

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

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

Description The SIDA package implements the SIDA and SIDANet algorithms for joint association and classification studies. The algorithms consider the overall association between multi-view data, and the separation within each view when choosing discriminant vectors that are associated and optimally separate subjects. SIDANet incorporates prior structural information in joint association and classification studies.
It uses the normalized Laplacian of a graph to smooth coefficients of predictor variables, thus encouraging selection of predictors that are connected and
behave similarly.

License GPL (>=2.0)

Imports CVXR, doParallel, foreach, RSpectra,igraph, Matrix

Encoding UTF-8

LazyData true

NeedsCompilation no

Depends R (>= 3.5.0)

R topics documented:

CorrelationPlots *Correlation Plots*

Description

Plots for visualizing correlation between estimated discriminant vectors for pairwise data.

Usage

```
CorrelationPlots(Xtestdata=Xtestdata,Ytest=Ytest,hatalpha=hatalpha)
```

Arguments

Xtestdata	A list with each entry containing views of size $ntest \times p_d$, where $d = 1, ..., D$.Rows are samples and columns are variables. Can use testing or training data
Ytest	$ntest \times 1$ vector of class membership.
hatalpha	A list of estimated sparse discriminant vectors for each view.

Details

The function will return correlation plot(s).

Value

NULL	output of plot

References

Sandra E. Safo, Eun Jeong Min, and Lillian Haine (2019) , *Sparse Linear Discriminant Analysis for Multi-view Structured Data, submitted*

See Also

cvSIDA,sidatunerange, DiscriminantPlots

Examples

```
library(SIDA)
data(DataExample)
##---- call sida algorithm to estimate discriminant vectors, and predict on testing data

Xdata=DataExample[[1]]
Y=DataExample[[2]]
Xtestdata=DataExample[[3]]
Ytest=DataExample[[4]]


#call sidatunerange to get range of tuning paramater
ngrid=10
mytunerange=sidatunerange(Xdata,Y,ngrid,standardize=TRUE,weight=0.5,withCov=FALSE)

# an example with Tau set as the lower bound
Tau=c(mytunerange$Tauvec[[1]][1], mytunerange$Tauvec[[2]][1])
```

```
#call sida
mysida=sida(Xdata,Y,Tau,withCov=FALSE,Xtestdata=Xtestdata,Ytest=Ytest,AssignClassMethod='Joint',
plotIt=TRUE, standardize=TRUE,maxiteration=20,weight=0.5,thresh= 1e-03)

test.error=mysida$sidaerror

test.correlation=mysida$sidacorrelation

#estimated discriminant vectors and predicted class
hatalpha=mysida$hatalpha

predictedClass=mysida$PredictedClass


##----plot discriminant and correlation plots
#---------Discriminant plot
mydisplot=DiscriminantPlots(Xtestdata,Ytest,mysida$hatalpha)

mycorrplot=CorrelationPlots(Xtestdata,Ytest,mysida$hatalpha)
```

`cvSIDA` *Cross validation for Sparse Integrative Discriminant Analysis for Multi-view Data*

Description

Peforms nfolds cross validation to select optimal tuning parameters for sida based on training data, which are then used with the training or testing data to predict class membership. Allows for inclusion of covariates which are not penalized. If you want to apply optimal tuning parameters to testing data, you may also use sida.

Usage

```
cvSIDA(Xdata=Xdata,Y=Y,withCov=FALSE,plotIt=FALSE, Xtestdata=NULL,Ytest=NULL,
       isParallel=TRUE, ncores=NULL,gridMethod='RandomSearch',
       AssignClassMethod='Joint',nfolds=5,ngrid=8,standardize=TRUE,
       maxiteration=20,weight=0.5,thresh=1e-03)
```

Arguments

`Xdata`	A list with each entry containing training views of size $n \times p_d$, where $d = 1, ..., D$. Rows are samples and columns are variables. If covariates are available, they should be included as a separate view, and set as the last dataset. For binary or categorical covariates (assumes no ordering), we suggest the use of indicator variables.
`Y`	$n \times 1$ vector of class membership.
`withCov`	TRUE or FALSE if covariates are available. If TRUE, please set all covariates as one dataset and should be the last dataset. For binary and categorical variables, use indicator matrices/vectors. Default is FALSE.

`plotIt` TRUE or FALSE. If TRUE, produces discriminants and correlation plots. Default is FALSE.

`Xtestdata` A list with each entry containing testing views of size $ntest \times p_d$, where $d = 1, ..., D$. Rows are samples and columns are variables. The order of the list should be the same as the order for the training data, Xdata. Use if you want to predict on a testing dataset. If no Xtestdata, set to NULL.

`Ytest` $ntest \times 1$ vector of test class membership. If no testing data provided, set to NULL.

`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.

`gridMethod` GridSearch or RandomSearch. Optimize tuning parameters over full grid or random grid. Default is RandomSearch.

`AssignClassMethod`
Classification method. Either Joint or Separate. Joint uses all discriminant vectors from D datasets to predict class membership. Separate predicts class membership separately for each dataset. Default is Joint

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

`ngrid` Number of grid points for tuning parameters. Default is 8 for each view if $D = 2$. If $D > 2$, default is 5.

`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.

`weight` Balances separation and association. Default is 0.5.

`thresh` threshold for convergence. Default is 0.001.

Details

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

Value

`sidaerror` estimated classication error. If testing data provided, this will be test classification error, otherwise, training error

`sidacorrelation`
sum of pairwise RV coefficients. Normalized to be within 0 and 1, inclusive.

`hatalpha` A list of estimated sparse discriminant vectors for each view.

`PredictedClass` Predicted class. If AssignClassMethod='Separate', this will be a $ntest \times D$ matrix, with each column the predicted class for each data.

`optTau` Optimal tuning parameters for each view, not including covariates, if available.

`gridValues` Grid values used for searching optimal tuning paramters.

`AssignClassMethod`
Classification method used. Joint or Separate.

`gridMethod` Grid method used. Either GridSearch or RandomSearch

References

Sandra E. Safo, Eun Jeong Min, and Lillian Haine (2019) , *Sparse Linear Discriminant Analysis for Multi-view Structured Data, submitted*

See Also

sida,CorrelationPlots,DiscriminantPlots

Examples

```
library(SIDA)
##---- read in sample data
data(DataExample)

Xdata=DataExample[[1]]
Y=DataExample[[2]]
Xtestdata=DataExample[[3]]
Ytest=DataExample[[4]]


##---- call cross validation

mycv=cvSIDA(Xdata,Y,withCov=FALSE,plotIt=FALSE, Xtestdata=Xtestdata,Ytest=Ytest,
            isParallel=TRUE,ncores=NULL,gridMethod='RandomSearch',
            AssignClassMethod='Joint',nfolds=5,ngrid=8,standardize=TRUE,
            maxiteration=20, weight=0.5,thresh=1e-03)

#check output
test.error=mycv$sidaerror

test.correlation=mycv$sidacorrelation

optTau=mycv$optTau

hatalpha=mycv$hatalpha

#---------Discriminant plot
mydisplot=DiscriminantPlots(Xtestdata,Ytest,mycv$hatalpha)

mycorrplot=CorrelationPlots(Xtestdata,Ytest,mycv$hatalpha)
```

---

cvSIDANet — *Cross validation for Sparse Integrative Discriminant Analysis for Multi-view Structured (Network) Data*

Description

Peforms nfolds cross validation to select optimal tuning parameters for sidanet based on training data, which are then used with the training or testing data to predict class membership. Allows for inclusion of covariates which are not penalized. If you want to apply optimal tuning parameters to testing data, you may also use sidanet.

Usage

```
cvSIDANet(Xdata=Xdata,Y=Y,myedges=myedges,myedgeweight=myedgeweight,withCov=FALSE,
        plotIt=FALSE,Xtestdata=NULL,Ytest=NULL,isParallel=TRUE,ncores=NULL,
       gridMethod='RandomSearch', AssignClassMethod='Joint', nfolds=5,ngrid=8,
        standardize=TRUE,maxiteration=20, weight=0.5,thresh=1e-03,eta=0.5)
```

Arguments

`Xdata` A list with each entry containing training views of size $n \times p_d$, where $d = 1, ..., D$. Rows are samples and columns are variables. If covariates are available, they should be included as a separate view, and set as the last dataset. For binary or categorical covariates (assumes no ordering), we suggest the use of indicator variables.

`Y` $n \times 1$ vector of class membership.

`myedges` A list with each entry containing a $M_d \times 2$ matrix of edge information for each view. If a view has no edge information, set to 0; this will default to SIDA. If covariates are available as a view (Dth view), the edge information should be set to 0.

`myedgeweight` A list with each entry containing a $M_d \times 1$ vector of weight information for each view. If a view has no weight information, set to 0; this will use the Laplacian of an unweighted graph. If covariates are available as a view (Dth view), the weight information should be set to 0.

`withCov` TRUE or FALSE if covariates are available. If TRUE, please set all covariates as one dataset and should be the last dataset. For binary and categorical variables, use indicator matrices/vectors. Default is FALSE.

`plotIt` TRUE or FALSE. If TRUE, produces discriminants and correlation plots. Default is FALSE.

`Xtestdata` A list with each entry containing testing views of size $ntest \times p_d$, where $d = 1, ..., D$. Rows are samples and columns are variables. The order of the list should be the same as the order for the training data, Xdata. Use if you want to predict on a testing dataset. If no Xtestdata, set to NULL.

`Ytest` $ntest \times 1$ vector of test class membership. If no testing data provided, set to NULL.

`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.

`gridMethod` GridSearch or RandomSearch. Optimize tuning parameters over full grid or random grid. Default is RandomSearch.

`AssignClassMethod`
Classification method. Either Joint or Separate. Joint uses all discriminant vectors from D datasets to predict class membership. Separate predicts class membership separately for each dataset. Default is Joint

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

`ngrid` Number of grid points for tuning parameters. Default is 8 for each view if $D = 2$. If $D > 2$, default is 5.

`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.

`weight` Balances separation and association. Default is 0.5.

`thresh` Threshold for convergence. Default is 0.001.

`eta` Balances the selection of network, and variables within network. Default is 0.5.

Details

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

Value

`sidaerror` Estimated classication error. If testing data provided, this will be test classification error, otherwise, training error

`sidacorrelation` Sum of pairwise RV coefficients. Normalized to be within 0 and 1, inclusive.

`hatalpha` A list of estimated sparse discriminant vectors for each view.

`PredictedClass` Predicted class. If AssignClassMethod='Separate', this will be a $ntest \times D$ matrix, with each column the predicted class for each data.

`optTau` Optimal tuning parameters for each view, not including covariates, if available.

`gridValues` Grid values used for searching optimal tuning paramters.

`AssignClassMethod` Classification method used. Joint or Separate.

`gridMethod` Grid method used. Either GridSearch or RandomSearch

References

Sandra E. Safo, Eun Jeong Min, and Lillian Haine (2019) , *Sparse Linear Discriminant Analysis for Multi-view Structured Data, submitted*

See Also

sidanet,CorrelationPlots,DiscriminantPlots

Examples

```
library(SIDA)
##---- read in sample data
data(SIDANetDataExample)

##---- call cross validation

#example with two views having edge weights

Xdata=SIDANetDataExample[[1]]
Y=SIDANetDataExample[[2]]
Xtestdata=SIDANetDataExample[[3]]
Ytest=SIDANetDataExample[[4]]
myedges=SIDANetDataExample[[5]]
myedgeweight=SIDANetDataExample[[6]]

mycv=cvSIDANet(Xdata,Y,myedges,myedgeweight,withCov=FALSE,plotIt=FALSE,Xtestdata=Xtestdata,
```

```
        Ytest=Ytest,isParallel=TRUE,ncores=NULL,gridMethod='RandomSearch',
        AssignClassMethod='Joint',nfolds=5,ngrid=8,standardize=TRUE,
        maxiteration=20, weight=0.5,thresh=1e-03,eta=0.5)


#check output
test.error=mycv$sidaneterror

test.correlation=mycv$sidanetcorrelation

optTau=mycv$optTau

hatalpha=mycv$hatalpha

#---------Discriminant plot
mydisplot=DiscriminantPlots(Xtestdata,Ytest,mycv$hatalpha)

mycorrplot=CorrelationPlots(Xtestdata,Ytest,mycv$hatalpha)
```

DataExample *Data example for SIDA*

Description

Simulated data to demonstrate the use of SIDA.

Usage

```
data(DataExample)
```

Format

A list with 4 elements

Xdata A list with each entry containing two views of training data with dimension 160×2000 each. Rows are samples and columns are variables.

Y 160×1 vector of training class membership. There are three classes each with size 80.

Xtestdata A list with each entry containing two views of testing data with dimension 320×2000 each. Rows are samples and columns are variables.

Ytest 320×1 vector of testing class membership. There are three classes each with size 160.

References

Sandra E. Safo, Eun Jeong Min, and Lillian Haine (2019) , *Sparse Linear Discriminant Analysis for Multi-view Structured Data*, *submitted*

Examples

```
# see cvSIDA or sida examples
```

DiscriminantPlots *Discriminant Plots*

Description

Plots discriminant vectors for visualizing class separation

Usage

```
DiscriminantPlots(Xtestdata=Xtestdata,Ytest=Ytest,hatalpha=hatalpha)
```

Arguments

Xtestdata	A list with each entry containing views of size $ntest \times p_d$, where $d = 1, ..., D$. Rows are samples and columns are variables. Can use testing or training data.
Ytest	$ntest \times 1$ vector of class membership.
hatalpha	A list of estimated sparse discriminant vectors for each view.

Details

The function will return discriminant plots.

Value

NULL

References

Sandra E. Safo, Eun Jeong Min, and Lillian Haine (2019) , *Sparse Linear Discriminant Analysis for Multi-view Structured Data, submitted*

See Also

cvSIDA,sidatunerange, CorrelationPlots

Examples

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

##---- call sida algorithm to estimate discriminant vectors, and predict on testing data

Xdata=DataExample[[1]]
Y=DataExample[[2]]
Xtestdata=DataExample[[3]]
Ytest=DataExample[[4]]


#call sidatunerange to get range of tuning paramater
ngrid=10
mytunerange=sidatunerange(Xdata,Y,ngrid,standardize=TRUE,weight=0.5,withCov=FALSE)
```

```
# an example with Tau set as the lower bound
Tau=c(mytunerange$Tauvec[[1]][1], mytunerange$Tauvec[[2]][1])

mysida=sida(Xdata,Y,Tau,withCov=FALSE,Xtestdata=Xtestdata,Ytest=Ytest,
            AssignClassMethod='Joint', plotIt=TRUE, standardize=TRUE,
            maxiteration=20,weight=0.5,thresh= 1e-03)

test.error=mysida$sidaerror

test.correlation=mysida$sidacorrelation

hatalpha=mysida$hatalpha

predictedClass=mysida$PredictedClass


##----plot discriminant and correlation plots
#---------Discriminant plot
mydisplot=DiscriminantPlots(Xtestdata,Ytest,mysida$hatalpha)

mycorrplot=CorrelationPlots(Xtestdata,Ytest,mysida$hatalpha)
```

`sida` *Sparse Integrative Discriminant Analysis for Multi-view Data*

Description

Performs sparse integrative disdcriminant analysis of multi-view data to 1) obtain discriminant vectors that are associated and optimally separate subjects into different classes 2) estimate misclassification rate, and total correlation coefficient. Allows for the inclusion of other covariates which are not penalized in the algorithm. It is recommended to use cvSIDA to choose best tuning parameter.

Usage

```
sida(Xdata=Xdata,Y=Y,Tau=Tau,withCov=FALSE,Xtestdata=Xtestdata,Ytest=Ytest,
    AssignClassMethod='Joint',plotIt=FALSE, standardize=TRUE,
    maxiteration=20,weight=0.5,thresh= 1e-03)
```

Arguments

Xdata	A list with each entry containing training views of size $n \times p_d$, where $d = 1, ..., D$ views. Rows are samples and columns are variables. If covariates are available, they should be included as a separate view, and set as the last dataset. For binary or categorical covariates (assumes no ordering), we suggest the use of indicator variables.
Y	$n \times 1$ vector of class membership.
Tau	$d \times 1$ vector of tuning parameter. It is recommended to use sidatunerange to obtain lower and upper bounds for the tuning parameters since too large a tuning parameter will result in a trivial solution vector (all zeros) and too small may result in non-sparse vectors.

`withCov` TRUE or FALSE if covariates are available. If TRUE, please set all covariates as one dataset and should be the last dataset. For binary and categorical variables, use indicator matrices/vectors. Default is FALSE.

`Xtestdata` A list with each entry containing testing views of size $ntest \times p_d$, where $d = 1, ..., D$. Rows are samples and columns are variables. The order of the list should be the same as the order for the training data, Xdata. Use if you want to predict on a testing dataset. If no Xtestdata, set to NULL.

`Ytest` $ntest \times 1$ vector of test class membership. If no testing data provided, set to NULL.

`AssignClassMethod` Classification method. Either Joint or Separate. Joint uses all discriminant vectors from D datasets to predict class membership. Separate predicts class membership separately for each dataset. Default is Joint

`plotIt` TRUE or FALSE. If TRUE, produces discriminants and correlation plots. Default is FALSE

`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.

`weight` Balances separation and association. Default is 0.5.

`thresh` Threshold for convergence. Default is 0.001.

Details

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

Value

`sidaerror` Estimated classication error. If testing data provided, this will be test classification error, otherwise, training error

`sidacorrelation` Sum of pairwise RV coefficients. Normalized to be within 0 and 1, inclusive.

`hatalpha` A list of estimated sparse discriminant vectors for each view.

`PredictedClass` Predicted class. If AssignClassMethod='Separate', this will be a $ntest \times D$ matrix, with each column the predicted class for each data.

References

Sandra E. Safo, Eun Jeong Min, and Lillian Haine (2019), *Sparse Linear Discriminant Analysis for Multi-view Structured Data, submitted*

See Also

cvSIDA, sidatunerange, CorrelationPlots, DiscriminantPlots

Examples

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

```
Xdata=DataExample[[1]]
Y=DataExample[[2]]
Xtestdata=DataExample[[3]]
Ytest=DataExample[[4]]


##---- call sida algorithm to estimate discriminant vectors, and predict on testing data

#call sidatunerange to get range of tuning paramater
ngrid=10
mytunerange=sidatunerange(Xdata,Y,ngrid,standardize=TRUE,weight=0.5,withCov=FALSE)

# an example with Tau set as the lower bound
Tau=c(mytunerange$Tauvec[[1]][1], mytunerange$Tauvec[[2]][1])

mysida=sida(Xdata,Y,Tau,withCov=FALSE,Xtestdata=Xtestdata,Ytest=Ytest,
            AssignClassMethod='Joint',plotIt=TRUE, standardize=TRUE,
            maxiteration=20,weight=0.5,thresh= 1e-03)

test.error=mysida$sidaerror

test.correlation=mysida$sidacorrelation

hatalpha=mysida$hatalpha

predictedClass=mysida$PredictedClass


##----plot discriminant and correlation plots

#---------Discriminant plot
mydisplot=DiscriminantPlots(Xtestdata,Ytest,mysida$hatalpha)

mycorrplot=CorrelationPlots(Xtestdata,Ytest,mysida$hatalpha)
```

`sidaclassify` *Classification approach for Sparse Integrative Discriminant Analysis*

Description

Performs classification using nearest centroid on separate or combined estimated discriminant vectors, and predicts class membership.

Usage

```
sidaclassify(hatalpha=hatalpha,Xtestdata=Xtestdata,Xdata=Xdata,Y=Y,
             AssignClassMethod='Joint', standardize=TRUE)
```

Arguments

hatalpha	A list of estimated sparse discriminant vectors for each view. This may be obtained from sida or cvSIDA

`Xtestdata` A list with each entry containing testing views of size $ntest \times p_d$, where $d = 1, ..., D$ views. Rows are samples and columns are variables. The order of the list should be the same as the order for the training data, Xdata. If covariates are available, they should be included as a separate view, and set as the last dataset. For binary or categorical covariates (assumes no ordering), we suggest the use of indicator variables. If you want to obtain training error, set as Xdata.

`Xdata` A list with each entry containing training views of size $n \times p_d$, where $d = 1, ..., D$ views. Rows are samples and columns are variables. If covariates are available, they should be included as a separate view, and set as the last dataset. For binary or categorical covariates (assumes no ordering), we suggest the use of indicator variables.

`Y` $n \times 1$ vector of class membership. Same size as the number of training samples.

`AssignClassMethod` Classification method. Either Joint or Separate. Joint uses all discriminant vectors from D datasets to predict class membership. Separate predicts class membership separately for each dataset. Default is Joint

`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 an R object, showing the predicted class and the classification method. To see the results, use the “$" operator.

Value

`PredictedClass` Predicted class. If AssignClassMethod='Separate', this will be a $ntest \times D$ matrix, with each column the predicted class for each data.

`AssignClassMethod` Classification method used. Joint or Separate.

References

Sandra E. Safo, Eun Jeong Min, and Lillian Haine (2019) , *Sparse Linear Discriminant Analysis for Multi-view Structured Data, submitted*

See Also

cvSIDA, sida, cvSIDANet, sidanet

Examples

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

Xdata=DataExample[[1]]
Y=DataExample[[2]]
Xtestdata=DataExample[[3]]
Ytest=DataExample[[4]]


#call sidatunerange to get range of tuning paramater
ngrid=10
```

```
mytunerange=sidatunerange(Xdata,Y,ngrid,standardize=TRUE,weight=0.5,withCov=FALSE)

# an example with Tau set as the lower bound
Tau=c(mytunerange$Tauvec[[1]][1], mytunerange$Tauvec[[2]][1])

mysida=sida(Xdata,Y,Tau,withCov=FALSE,Xtestdata=Xtestdata,Ytest=Ytest)


#classification with combined estimated vectors
mysida.classify.Joint=sidaclassify(mysida$hatalpha,Xtestdata,Xdata,Y,
                        AssignClassMethod='Joint')

mysida.PredClass.Joint=mysida.classify.Joint$PredictedClass

#classification with separate estimated vectors
mysida.classify.Separate=sidaclassify(mysida$hatalpha,Xtestdata,Xdata,Y,
                        AssignClassMethod='Separate')

mysida.PredClass.Separate=mysida.classify.Separate$PredictedClass
```

---

sidanet *Sparse Integrative Discriminant Analysis for Multi-view Structured (Network) Data*

Description

Performs sparse integrative disdcriminant analysis of multi-view structured (network) data to 1) obtain discriminant vectors that are associated and optimally separate subjects into different classes 2) estimate misclassification rate, and total correlation coefficient. The Laplacian of the underlying graph is used to smooth the discriminant vectors to encourage variables within a view that are connected to have a similar effect. Allows for the inclusion of other covariates which are not penalized in the algorithm. It is recommended to use cvSIDANet to choose best tuning parameter.

Usage

```
sidanet(Xdata=Xdata,Y=Y,myedges=myedges,myedgeweight=myedgeweight,
        Tau=Tau,withCov=FALSE,Xtestdata=NULL,Ytest=NULL,
        AssignClassMethod='Joint',plotIt=FALSE, standardize=TRUE,
        maxiteration=20,weight=0.5,thresh= 1e-03,eta=0.5,
        mynormLaplacianG=NULL)
```

Arguments

Xdata	A list with each entry containing training views of size $n \times p_d$, where $d = 1, ..., D$. Rows are samples and columns are variables. If covariates are available, they should be included as a separate view, and set as the last dataset. For binary or categorical covariates (assumes no ordering), we suggest the use of indicator variables.
Y	$n \times 1$ vector of class membership.

`myedges` A list with each entry containing a $M_d \times 2$ matrix of edge information for each view. If a view has no edge information, set to 0; this will default to SIDA. If covariates are available as a view (Dth view), the edge information should be set to 0.

`myedgeweight` A list with each entry containing a $M_d \times 1$ vector of weight information for each view. If a view has no weight information,set to 0; this will use the Laplacian of an unweighted graph. If covariates are available as a view (Dth view), the weight information should be set to 0.

`Tau` $d \times 1$ vector of tuning parameter. It is recommended to use sidatunerange to obtain lower and upper bounds for the tuning parameters since too large a tuning parameter will result in a trivial solution vector (all zeros) and too small may result in non-sparse vectors.

`withCov` TRUE or FALSE if covariates are available. If TRUE, please set all covariates as one dataset and should be the last dataset. For binary and categorical variables, use indicator matrices/vectors. Default is FALSE.

`Xtestdata` A list with each entry containing testing views of size $ntest \times p_d$, where $d = 1, ..., D$. Rows are samples and columns are variables. The order of the list should be the same as the order for the training data, Xdata. Use if you want to predict on a testing dataset. If no Xtestdata, set to NULL.

`Ytest` $ntest \times 1$ vector of test class membership. If no testing data provided, set to NULL.

`AssignClassMethod` Classification method. Either Joint or Separate. Joint uses all discriminant vectors from D datasets to predict class membership. Separate predicts class membership separately for each dataset. Default is Joint

`plotIt` TRUE or FALSE. If TRUE, produces discriminants and correlation plots. Default is FALSE

`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.

`weight` Balances separation and association. Default is 0.5.

`thresh` Threshold for convergence. Default is 0.001.

`eta` Balances the selection of network, and variables within network. Default is 0.5.

`mynormLaplacianG` The normalized Laplacian of a graph. Set to NULL and this would be estimated using edge matrix and edge weights.

Details

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

Value

`sidaneterror` Estimated classication error. If testing data provided, this will be test classification error, otherwise, training error

`sidanetcorrelation` Sum of pairwise RV coefficients. Normalized to be within 0 and 1, inclusive.

`hatalpha` A list of estimated sparse discriminant vectors for each view.

`PredictedClass` Predicted class. If AssignClassMethod='Separate', this will be a $ntest \times D$ matrix, with each column the predicted class for each data.

References

Sandra E. Safo, Eun Jeong Min, and Lillian Haine (2019) , *Sparse Linear Discriminant Analysis for Multi-view Structured Data, submitted*

See Also

cvSIDANet,sidatunerange, CorrelationPlots,DiscriminantPlots

Examples

```
library(SIDA)
##---- read in data
data(SIDANetDataExample)
##---- call sidanet algorithm to estimate discriminant vectors, and predict on testing data

#call sidanettunerange to get range of tuning paramater

Xdata=SIDANetDataExample[[1]]
Y=SIDANetDataExample[[2]]
Xtestdata=SIDANetDataExample[[3]]
Ytest=SIDANetDataExample[[4]]
myedges=SIDANetDataExample[[5]]
myedgeweight=SIDANetDataExample[[6]]


ngrid=10
mytunerange=sidanettunerange(Xdata,Y,ngrid,standardize=TRUE,weight=0.5,eta=0.5,
                myedges,myedgeweight)

# an example with Tau set as the lower bound
Tau=c(mytunerange$Tauvec[[1]][1], mytunerange$Tauvec[[2]][1])

#example with two views having edge weights
mysidanet=sidanet(Xdata,Y,myedges,myedgeweight,Tau,Xtestdata=Xtestdata,Ytest=Ytest)


test.error=mysidanet$sidaneterror

test.correlation=mysidanet$sidanetcorrelation

hatalpha=mysidanet$hatalpha

predictedClass=mysidanet$PredictedClass


##----plot discriminant and correlation plots

#---------Discriminant plot
mydisplot=DiscriminantPlots(Xtestdata,Ytest,mysidanet$hatalpha)

mycorrplot=CorrelationPlots(Xtestdata,Ytest,mysidanet$hatalpha)
```

`SIDANetDataExample` *Data example for SIDANet*

Description

Simulated data to demonstrate the use of SIDANet.

Usage

```
data(SIDANetDataExample)
```

Format

A list with 6 elements:

XdataNet A list with each entry containing two views of training data with dimension 240×1000 each. Rows are samples and columns are variables.

YNet 240×1 vector of training class membership. There are three classes each with size 80.

XtestdataNet A list with each entry containing two views of testing data with dimension 480×1000 each. Rows are samples and columns are variables.

YtestNet 480×1 vector of testing class membership. There are three classes each with size 160.

myedges A list with each entry containing a 36×2 matrix of edge information for each view. Assumes variable 1 is connected to variables 2 to 10, variable 11 is connected to variables 12 to 20, variable 21 is connected to variables 22 to 30 and variable 31 is connected to variables 32 to 40. All remaining variables are singletons.

myedgeweight A list with each entry containing edgeweight. In this example, views 1 and 2 have edge weights so the Laplacian of a weighted graph will be used.

References

Sandra E. Safo, Eun Jeong Min, and Lillian Haine (2019) , *Sparse Linear Discriminant Analysis for Multi-view Structured Data, submitted*

Examples

```
# see cvSIDANet or sidanet examples
```

`sidanettunerange` *Tuning paramter grid values for sidanet*

Description

Sidanet function to provide tuning parameter grid values for each view, not including covariates, if available. It is recommended to use this to get lower and upper bounds of tuning parameters for each view that can be used in sidanet. This function is called by cvSIDANet to select optimal tuning parameters.

Usage

```
sidanettunerange(Xdata=Xdata,Y=Y,ngrid=8,standardize=TRUE,weight=0.5,eta=0.5,
                 myedges=myedges,myedgeweight=myedgeweight,withCov=FALSE)
```

Arguments

Xdata	A list with each entry containing training views of size $n \times p_d$, where $d = 1, ..., D$ views. Rows are samples and columns are variables. If covariates are available, they should be included as a separate view, and set as the last dataset. For binary or categorical covariates (assumes no ordering), we suggest the use of indicator variables.
Y	$n \times 1$ vector of class membership. Same size as the number of training samples.
ngrid	Number of grid points for tuning parameters.
standardize	TRUE or FALSE. If TRUE, data will be normalized to have mean zero and variance one for each variable. Default is TRUE.
weight	Balances separation and association. Default is 0.5.
eta	Balances the selection of network, and variables within network. Default is 0.5.
myedges	A list with each entry containing a $M_d \times 2$ matrix of edge information for each view. If a view has no edge information, set to 0; this will default to SIDA. If covariates are available as a view (Dth view), the edge information should be set to 0.
myedgeweight	A list with each entry containing a $M_d \times 1$ vector of weight information for each view. If a view has no weight information, set to 0; this will use the Laplacian of an unweighted graph. If covariates are available as a view (Dth view), the weight information should be set to 0.
withCov	TRUE or FALSE if covariates are available. If TRUE, set all covariates as one dataset and should be the last dataset. For binary and categorical variables, use indicator matrices/vectors. Default is FALSE.

Details

The function will return an R object with grid values for each data, not including covariates, if available. To see the results, use the “$" operator.

Value

Tauvec	Grid values for each data, not including covariates, if available.

References

Sandra E. Safo, Eun Jeong Min, and Lillian Haine (2019) , *Sparse Linear Discriminant Analysis for Multi-view Structured Data*, *submitted*

See Also

sidanet

Examples

```
# see examples in sidanet
```

sidatunerange *Tuning paramter grid values for sida*

Description

Sida function to provide tuning parameter grid values for each view, not including covariates, if available. It is recommended to use this to get lower and upper bounds of tuning parameters for each view that can be used in sida. This function is called by cvSIDA to select optimal tuning parameters.

Usage

```
sidatunerange(Xdata=Xdata,Y=Y,ngrid=8,standardize=TRUE,weight=0.5,withCov=FALSE)
```

Arguments

Xdata	A list with each entry containing each $n \times p_d$ training view, where $d = 1, ..., D$ views. Rows are samples and columns are variables. If covariates are available, they should be included as a separate view, and set as the last dataset. For binary or categorical covariates (assumes no ordering), we suggest the use of indicator variables.
Y	$n \times 1$ vector of class membership. Same size as the number of training samples.
ngrid	Number of grid points for tuning parameters.
standardize	TRUE or FALSE. If TRUE, data will be normalized to have mean zero and variance one for each variable. Default is TRUE.
weight	Balances separation and association. Default is 0.5.
withCov	TRUE or FALSE if covariates are available. If TRUE, set all covariates as one dataset and should be the last dataset. For binary and categorical variables, use indicator matrices/vectors. Default is FALSE.

Details

The function will return an R object with grid values for each data, not including covariates, if available. To see the results, use the “$" operator.

Value

Tauvec	grid values for each data, not including covariates, if available.

References

Sandra E. Safo, Eun Jeong Min, and Lillian Haine (2019) , *Sparse Linear Discriminant Analysis for Multi-view Structured Data*, *submitted*

See Also

sida

Examples

```
## see examples in sida
```

Index