Package 'superMDS'

February 20, 2015

Type Package

Title Implements the supervised multidimensional scaling (superMDS) proposal of Witten and Tibshirani (2011)

Version 1.0.2

Date 2013-01-02

Author Daniela M. Witten

Maintainer Daniela Witten <dwitten@u.washington.edu>

Description Witten and Tibshirani (2011) Supervised multidimensional scaling for visualization, classification, and bipartite ranking. Computational Statistics and Data Analysis 55(1): 789-801.

License GPL-2

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2013-12-02 18:29:00

R topics documented:

`superMDS-package` *Supervised multidimensional scaling for visualization, classification, and bipartite ranking*

Description

A method for implementing the supervised multidimensional scaling proposal of Witten and Tibshirani (2011)

Details

Package:	superMDS
Type:	Package
Version:	1.0.2
Date:	2013-01-02
License:	GPL-2
LazyLoad:	yes

Supervised multidimensional scaling (MDS) is a supervised version of least squares MDS. Suppose that we have a nxn dissimilarity matrix D and we want to find a set of n configuration points z1,...,zn, each a vector of length s, so that D is well-approximated by the Euclidean distances between the configuration points. Then least squares MDS can be used. However, suppose that we also have a vector of binary class labels associated with the dissimilarity matrix, yi = 1 or 2 for i=1,...,n. Then we might want configuration points whose Euclidean dsitances approximate D, and also that have the property that zis > zjs when yi > yj. This is the objective of supervised MDS. It leads to a method for visualizing observations, as well as a classification method. Details can be found in the paper below.

Author(s)

Daniela M. Witten

Maintainer: Daniela Witten <dwitten@u.washington.edu>

References

Witten and Tibshirani (2011) Supervised multidimensional scaling for visualization, classification, and bipartite ranking. Computational Statistics and Data Analysis 55(1): 789-801.

Examples

```
########### Generate some data  ############
n <- 30
p <- 10
x <- matrix(rnorm(n*p),ncol=p)
y <- c(rep(1,n/2),rep(2,n/2))
xte <- matrix(rnorm(n*p),ncol=p)
yte <- c(rep(1,n/2),rep(2,n/2))
x[y==1,1:(p)] <-  x[y==1,1:(p)] + .4
x[y==2,1:(p)] <-  x[y==2,1:(p)] - .4
xte[yte==1,1:(p)] <-  xte[yte==1,1:(p)] + .4
xte[yte==2,1:(p)] <-  xte[yte==2,1:(p)] - .4
# Done generating data  #

########### Perform SuperMDS ##############
out <- TrainSuperMDS(x=x,y=y,alpha=.4,S=2, silent=TRUE)
# A plot of the training configuration points #
par(mfrow=c(1,2))
plot(out$z, col=yte, main="Training Data", xlab="Dimension 1", ylab="Dimension 2")
testout <- TestSuperMDS(trout=out,xte=xte)
ytehat <- testout$ytehat
```

```
# A table showing the true vs predicted class labels #
print(table(ytehat,yte))
# A plot of the test configuration points #
plot(testout$zte, col=yte, main="Test Data", xlab="Dimension 1", ylab="Dimension 2")
```

`TestSuperMDS` *Given the configuration points of a set of training observations, and the dissimilarities between the training and test observations, compute the configuration points for a set of test observations.*

Description

Suppose that we have training data observations x (of dimension nxp) with an associated binary outcome vector y of length n, and that TrainSuperMDS has already been run on the training observations. Furthermore, we have test observations xte (of dimension mxp) for which we do not have an outcome vector. This function will predict the class of the test observations, and also to compute configuration points for the test observations.

Usage

```
TestSuperMDS(trout, xte = NULL, dtetr = NULL)
```

Arguments

`trout`	The output of a call to TrainSuperMDS on the training data. We assume that there were n training observations which were either passed into TrainSuperMDS as a nxn dissimilarity matrix, or as a nxp dat matrix.
`xte`	The test observations, a matrix of dimension mxp. If this is NULL then must pass in dtetr. Can pass in xte only if previously passed in x when TrainSuperMDS was called. Otherwise, pass in dtetr instead.
`dtetr`	A mxn data matrix with the dissimilarity between each test observation and each training observation; if NULL then must pass in xte.

Value

`ytehat`	Predicted class labels for test data.
`zte`	Predicted configuration points for test data; should be a matrix of dimension mxS where S is the dimension of training configuration points.

Author(s)

Daniela M Witten

References

Witten and Tibshirani (2011) Supervised multidimensional scaling for visualization, classificatoin, and bipartite ranking. CSDA.

See Also

`TrainSuperMDS`

Examples

```
# Try ?superMDS for examples.
```

`TrainSuperMDS` *Find a set of configuration points that agree with a dissimilarity matrix D and a vector of class labels y*

Description

Given a nxn dissimilarity matrix D and a n-vector of binary (1,2) class labels y, this function outputs a set of configuration points z1,...,zn, each a S-vector, such that the distances between the configuration points approximate the dissimilarity matrix D, AND such that zis >= zjs tends to occur when yi >= yj.

Usage

```
TrainSuperMDS(d = NULL, y, alpha = 0.5, S = 2, x = NULL, nstarts = 5, silent = FALSE)
```

Arguments

`d`	A nxn dissimilarity matrix. If NULL, then x, a nxp data matrix, must be input instead.
`y`	A n-vector of binary labels, in the form of 1's and 2's. For instance, c(1,1,1,2,2) could be input if D is a 5x5 matrix.
`alpha`	A scalar between 0 and 1. If alpha=0 then this is just least squares MDS, and if alpha=1 then it's completely supervised.
`S`	The number of dimensions of the configuration points z1,...,zn. Must be at least equal to 1.
`x`	A nxp data matrix, to be input only if D is NULL.
`nstarts`	The supervised MDS algorithm finds a local minimum for the objective. Here, specify the number of initial values to try. If nstarts>1 then the set of configuration points corresponding to the optimal (smallest) value of the objective will be reported.
`silent`	Set to TRUE in order to turn off printing output to screen.

Value

`z`	A nxS matrix of the configuration points obtained.
`crits`	The values of the criterion obtained at the iterations of the algorithm.
`stress`	The portion of the final criterion value that are due to the STRESS component of the objective function.
`super`	The portion of the final criterion value that are due to the SUPERVISED component of the objective function.

Author(s)

Daniela M Witten

References

Witten and Tibshirani (2011) Supervised multidimensional scaling for visualization, classification, and bipartite ranking. Computational Statistics and Data Analysis.

See Also

TestSuperMDS

Examples

```
# Try ?superMDS for examples
```

Index