Package ‘sparseLDA’

October 14, 2022

Version 0.1-9
Date 2016-09-22
Title Sparse Discriminant Analysis
Author Line Clemmensen <lhc@imm.dtu.dk>, contributions by Max Kuhn
Maintainer Max Kuhn <mxkuhn@gmail.com>
Imports elasticnet, MASS, mda
Depends R (>= 2.10)
Description Performs sparse linear discriminant analysis for Gaussians and mixture of Gaussian models.
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2016-09-22 17:10:01

R topics documented:

normalize .................................................. 2
normalizetest ............................................. 3
penicilliumYES .......................................... 4
predict.sda ................................................ 5
sda ........................................................ 6
smda ....................................................... 8

Index 11
normalize  

Normalize training data

Description

Normalize a vector or matrix to zero mean and unit length columns.

Usage

normalize(X)

Arguments

X  a matrix with the training data with observations down the rows and variables in the columns.

Details

The function can e.g. be used for the training data in sda or smda.

Value

Returns a list with the following attributes:

Xc  The normalized data.
mx  Mean of columns of X.
vx  Length of columns of X.
Id  Logical vector indicating which variables are included in X. If some of the columns have zero length they are omitted.

Author(s)

Line Clemmensen

References


See Also

normalizetest, sda, smda
Examples

## Data
X<-matrix(sample(seq(3),12,replace=TRUE),nrow=3)

## Normalize data
Nm<-normalize(X)
print(Nm$Xc)

## See if any variables have been removed
which(!Nm$Id)

descrip

---

normalizeetest Normalize test data

Description

Normalize test data using output from the normalize() of the training data

Usage

normalizeetest(Xtst,Xn)

Arguments

- **Xtst**: a matrix with the test data with observations down the rows and variables in the columns.
- **Xn**: List with the output from normalize(Xtr) of the training data.

Details

The function can e.g. be used to normalize the testing data in sda or smda.

Value

Returns the normalized test data

- **Xtst**: The normalized data.

Author(s)

Line Clemmensen

References

### Description

The data set `penicilliumYES` has 36 rows and 3754 columns. The variables are 1st order statistics from multi-spectral images of three species of *Penicillium* fungi: *Melanoconidium*, *Polonicum*, and *Venetum*. These are the data used in the Clemmemsen et al "Sparse Discriminant Analysis" paper.

### Usage

```r
data(penicilliumYES)
```

### Format

This data set contains the following matrices:

- **X**: A matrix with 36 columns and 3754 rows. The training and test data. The first 12 rows are *P. Melanoconidium* species, rows 13-24 are *P. Polonicum* species, and the last 12 rows are *P. Venetum* species. The samples are ordered so that each pair of three is from the same isolate.

- **Y**: A matrix of dummy variables for the training data.

- **Z**: Z matrix of probabilities for the subclasses of the training data.

### Details

The X matrix is not normalized.

### Source

http://www.imm.dtu.dk/~lhc

---

#### Data

```r
Xtr<-matrix(sample(seq(3),12,replace=TRUE),nrow=3)
Xtst<-matrix(sample(seq(3),12,replace=TRUE),nrow=3)

## Normalize training data
Nm<-normalize(Xtr)

## Normalize test data
Xtst<-normalizetest(Xtst,Nm)
```
References


---

**predict.sda**

*Predict method for Sparse Discriminant Methods*

**Description**

Prediction functions for `link{sda}` and `link{smda}`.

**Usage**

```r
## S3 method for class 'sda'
predict(object, newdata = NULL, ...)
## S3 method for class 'smda'
predict(object, newdata = NULL, ...)
```

**Arguments**

- `object` an object of class `link{sda}` or `link{smda}`
- `newdata` a matrix or data frame of predictors
- `...` arguments passed to `link[MASS]{predict.lda}`

**Details**

The current implementation for mixture discriminant models current predicts the subclass probabilities.

**Value**

A list with components:

- `class` The classification (a factor)
- `posterior` posterior probabilities for the classes (or subclasses for `link{smda}`)
- `x` the scores
Sparse discriminant analysis

Description

Performs sparse linear discriminant analysis. Using an alternating minimization algorithm to minimize the SDA criterion.

Usage

sda(x, ...)

## Default S3 method:
sda(x, y, lambda = 1e-6, stop = -p, maxIte = 100, Q = K-1, trace = FALSE, tol = 1e-6, ...)

Arguments

- **x**: A matrix of the training data with observations down the rows and variables in the columns.
- **y**: A matrix initializing the dummy variables representing the groups.
- **lambda**: The weight on the L2-norm for elastic net regression. Default: 1e-6.
- **stop**: If STOP is negative, its absolute value corresponds to the desired number of variables. If STOP is positive, it corresponds to an upper bound on the L1-norm of the b coefficients. There is a one to one correspondence between stop and t. The default is -p (-the number of variables).
- **maxIte**: Maximum number of iterations. Default: 100.
- **Q**: Number of components. Maximum and default is K-1 (the number of classes less one).
- **trace**: If TRUE, prints out its progress. Default: FALSE.
- **tol**: Tolerance for the stopping criterion (change in RSS). Default is 1e-6.
- ... additional arguments

Details

The function finds sparse directions for linear classification.

Value

Returns a list with the following attributes:

- **beta**: The loadings of the sparse discriminative directions.
- **theta**: The optimal scores.
- **rss**: A vector of the Residual Sum of Squares at each iteration.
- **varNames**: Names on included variables
Author(s)

Line Clemmensen, modified by Trevor Hastie

References


See Also

normalize, normalizetest, smda

Examples

```r
## load data
data(penicilliumYES)

X <- penicilliumYES$X
Y <- penicilliumYES$Y
colnames(Y) <- c("P. Melanoconidium", "P. Polonicum", "P. Venetum")

## test samples
Iout<-c(3,6,9,12)
Iout<-c(Iout,Iout+12,Iout+24)

## training data
Xtr<-X[-Iout,]
k<-3
m<-dim(Xtr)[1]

## Normalize data
Xc<-normalize(Xtr)
Xn<-Xc$Xc
p<-dim(Xn)[2]

## Perform SDA with one non-zero loading for each discriminative
## direction with Y as matrix input
out <- sda(Xn, Y,
    lambda = 1e-6,
    stop = -1,
    maxIte = 25,
    trace = TRUE)

## predict training samples
train <- predict(out, Xn)

## testing
Xtst<-X[Iout,]
Xtst<-normalizetest(Xtst,Xc)
```
## smda

**Sparse mixture discriminant analysis**

### Description

Performs sparse linear discriminant analysis for mixture of gaussians models.

### Usage

```r
smda(x, ...)  
```  
### Arguments

- `x` A matrix of the training data with observations down the rows and variables in the columns.
- `y` A matrix initializing the dummy variables representing the groups.
- `Z` An optional matrix initializing the probabilities representing the groups.
- `Rj` K length vector containing the number of subclasses in each of the K classes.
- `stop` If STOP is negative, its absolute value corresponds to the desired number of variables. If STOP is positive, it corresponds to an upper bound on the L1-norm of the b coefficients. There is a one to one correspondence between stop and t.
- `maxIte` Maximum number of iterations. Default: 50.
- `Q` The number of components to include. Maximum and default is R-1 (total number of subclasses less one).
- `trace` If TRUE, prints out its progress. Default: FALSE.
- `tol` Tolerance for the stopping criterion (change in RSS). Default: 1e-4
- `...` additional arguments
Details

The function finds sparse directions for linear classification of mixture of gaussians models.

Value

Returns a list with the following attributes:

- `call`: The call
- `beta`: The loadings of the sparse discriminative directions.
- `theta`: The optimal scores.
- `Z`: Updated subclass probabilities.
- `Rj`: A vector of the number of subclasses per class
- `rss`: A vector of the Residual Sum of Squares at each iteration.

Author(s)

Line Clemmensen

References


See Also

normalize, normalizetest, sda

Examples

```r
# load data
data(penicilliumYES)
X <- penicilliumYES$X
Y <- penicilliumYES$Y
Z <- penicilliumYES$Z

## test samples
Iout <- c(3, 6, 9, 12)
Iout <- c(Iout, Iout+12, Iout+24)

## training data
Xtr <- X[-Iout,]
k <- 3
n <- dim(Xtr)[1]
Rj <- rep(4, 3)

## Normalize data
Xc <- normalize(Xtr)
Xn <- Xc$Xc
p <- dim(Xn)[2]
```
## perform SMDA with one non-zero loading for each discriminative direction

## Not run:
```r
smdaFit <- smda(x = Xn,
                 y = Y,
                 Z = Z,
                 Rj = Rj,
                 lambda = 1e-6,
                 stop = -5,
                 maxIte = 10,
                 tol = 1e-2)
```

# testing
```r
Xtst <- X[Iout,]
Xtst <- normalizetest(Xtst, Xc)

test <- predict(smdaFit, Xtst)
```

## End(Not run)
Index

* classif
  penicilliumYES, 4
  sda, 6
  smda, 8

* datasets
  penicilliumYES, 4

* multivariate
  normalize, 2
  normalizetest, 3
  penicilliumYES, 4
  predict.sda, 5
  sda, 6
  smda, 8
  normalize, 2, 4, 7, 9
  normalizetest, 2, 3, 7, 9
  penicilliumYES, 4
  predict.sda, 5
  predict.smda (predict.sda), 5
  sda, 2, 4, 6, 9
  smda, 2, 4, 7, 8