Package ‘sda’
November 21, 2021

Version 1.3.8
Date 2021-11-21
Title Shrinkage Discriminant Analysis and CAT Score Variable Selection
Author Miika Ahdesmaki, Verena Zuber, Sebastian Gibb, and Korbinian Strimmer
Maintainer Korbinian Strimmer <strimmerlab@gmail.com>
Depends R (>= 3.0.2), entropy (>= 1.3.1), corpcor (>= 1.6.10), fdrtool
(>= 1.2.17)
Suggests crossval
Imports graphics, stats, utils
Enhances care
Description Provides an efficient framework for high-dimensional linear and diagonal discriminant analysis with variable selection. The classifier is trained using James-Stein-type shrinkage estimators and predictor variables are ranked using correlation-adjusted t-scores (CAT scores). Variable selection error is controlled using false non-discovery rates or higher criticism.
License GPL (>= 3)
URL https://strimmerlab.github.io/software/sda/
NeedsCompilation no
Repository CRAN
Date/Publication 2021-11-21 17:50:02 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>sda-package</td>
<td>2</td>
</tr>
<tr>
<td>catscore</td>
<td>3</td>
</tr>
<tr>
<td>centroids</td>
<td>5</td>
</tr>
<tr>
<td>khan2001</td>
<td>6</td>
</tr>
<tr>
<td>predict.sda</td>
<td>8</td>
</tr>
<tr>
<td>sda</td>
<td>9</td>
</tr>
<tr>
<td>sda.ranking</td>
<td>11</td>
</tr>
<tr>
<td>singh2002</td>
<td>14</td>
</tr>
</tbody>
</table>
**The sda Package**

**Description**

This package performs linear discriminant analysis (LDA) and diagonal discriminant analysis (DDA) with variable selection using correlation-adjusted t (CAT) scores.

The classifier is trained using James-Stein-type shrinkage estimators. Variable selection is based on ranking predictors by CAT scores (LDA) or t-scores (DDA). A cutoff is chosen by false non-discovery rate (FNDR) or higher criticism (HC) thresholding.

This approach is particularly suited for high-dimensional classification with correlation among predictors. For details see Zuber and Strimmer (2009) and Ahdesmäki and Strimmer (2010).

Typically the functions in this package are applied in three steps:

- Step 1: feature selection with `sda.ranking`,
- Step 2: training the classifier with `sda`, and
- Step 3: classification using `predict.sda`.

The accompanying web site (see below) provides example R scripts to illustrate the functionality of this package.

**Author(s)**

Miika Ahdesmäki, Verena Zuber and Korbinian Strimmer (https://strimmerlab.github.io/)

**References**


See website: https://strimmerlab.github.io/software/sda/

**See Also**

catscore, sda.ranking, sda, predict.sda.
catscore

Estimate CAT Scores and t-Scores

Description

catscore computes CAT scores (correlation-adjusted t-scores) between the group centroids and the pooled mean.

Usage

catscore(Xtrain, L, lambda, lambda.var, lambda.freqs, diagonal=FALSE, verbose=TRUE)

Arguments

- **Xtrain**: A matrix containing the training data set. Note that the rows correspond to observations and the columns to variables.
- **L**: A factor with the class labels of the training samples.
- **lambda**: Shrinkage intensity for the correlation matrix. If not specified it is estimated from the data. lambda=0 implies no shrinkage and lambda=1 complete shrinkage.
- **lambda.var**: Shrinkage intensity for the variances. If not specified it is estimated from the data. lambda.var=0 implies no shrinkage and lambda.var=1 complete shrinkage.
- **lambda.freqs**: Shrinkage intensity for the frequencies. If not specified it is estimated from the data. lambda.freqs=0 implies no shrinkage (i.e. empirical frequencies) and lambda.freqs=1 complete shrinkage (i.e. uniform frequencies).
- **diagonal**: for diagonal=FALSE (the default) CAT scores are computed; otherwise with diagonal=TRUE t-scores.
- **verbose**: Print out some info while computing.

Details

CAT scores generalize conventional t-scores to account for correlation among predictors (Zuber and Strimmer 2009). If there is no correlation then CAR scores reduce to t-scores. The squared CAR scores provide a decomposition of Hotelling’s T^2 statistic.

CAT scores for two classes are described in Zuber and Strimmer (2009), for the multi-class case see Ahdesmäki and Strimmer (2010).

The scale factors for t-scores and CAT-scores are computed from the estimated frequencies (for empirical scale factors set lambda.freqs=0).

Value

catscore returns a matrix containing the cat score (or t-score) between each group centroid and the pooled mean for each feature.
Author(s)

References

See Also
sda.ranking, carscore.

Examples
# load sda library
library("sda")

#########################################
# training data #
#########################################

# prostate cancer set
data(singh2002)

# training data
Xtrain = singh2002$x
Ytrain = singh2002$y
dim(Xtrain)

#########################################
# shrinkage t-score (DDA setting - no correlation) #
#########################################
tstat = catscore(Xtrain, Ytrain, diagonal=TRUE)
dim(tstat)
tstat[1:10,]

#########################################
# shrinkage CAT score (LDA setting - with correlation) #
#########################################
cat = catscore(Xtrain, Ytrain, diagonal=FALSE)
dim(cat)
cat[1:10,]
centroids

Group Centroids and (Pooled) Variances

Description

centroids computes group centroids, the pooled mean and pooled variance, and optionally the group specific variances.

Usage

centroids(x, L, lambda.var, lambda.freqs, var.groups=FALSE, centered.data=FALSE, verbose=TRUE)

Arguments

x A matrix containing the data set. Note that the rows are sample observations and the columns are variables.

L A factor with the group labels.

lambda.var Shrinkage intensity for the variances. If not specified it is estimated from the data, see details below. lambda.var=0 implies no shrinkage and lambda.var=1 complete shrinkage.

lambda.freqs Shrinkage intensity for the frequencies. If not specified it is estimated from the data. lambda.freqs=0 implies no shrinkage (i.e. empirical frequencies) and lambda.freqs=1 complete shrinkage (i.e. uniform frequencies).

var.groups Estimate group-specific variances.

centered.data Return column-centered data matrix.

verbose Provide some messages while computing.

Details

As estimator of the variance we employ var.shrink as described in Opgen-Rhein and Strimmer (2007). For the estimates of frequencies we rely on freqs.shrink as described in Hauser and Strimmer (2009). Note that the pooled mean is computed using the estimated frequencies.

Value

centroids returns a list with the following components:

samples a vector containing the samples sizes in each group,

freqs a vector containing the estimated frequency in each group,

means the group means and the pooled mean,

variances the group-specific and the pooled variances, and

centered.data a matrix containing the centered data.
Author(s)


See Also

var.shrink, powcor.shrink.

Examples

# load sda library
library("sda")

## prepare data set
data(iris) # good old iris data
X = as.matrix(iris[,1:4])
Y = iris[,5]

## estimate centroids and empirical pooled variances
centroids(X, Y, lambda.var=0)

## also compute group-specific variances
centroids(X, Y, var.groups=TRUE, lambda.var=0)

## use shrinkage estimator for the variances
centroids(X, Y, var.groups=TRUE)

## return centered data
xc = centroids(X, Y, centered.data=TRUE)$centered.data
apply(xc, 2, mean)

## useful, e.g., to compute the inverse pooled correlation matrix
powcor.shrink(xc, alpha=-1)

khan2001

_description_

Childhood Cancer Study of Khan et al. (2001)

Description

Gene expression data (2308 genes for 88 samples) from the microarray study of Khan et al. (2001).

Usage

data(khan2001)
Format

khan2001$x is a 88 x 2308 matrix containing the expression levels. Note that rows correspond to samples, and columns to genes. The row names are the original image IDs, and the column names the orginal probe labels.

khan2001$y is a factor containing the diagnosis for each sample ("BL", "EWS", "NB", "non-SRBCT", "RMS").

khan2001$descr provides some annotation for each gene.

Details

This data set contains measurements of the gene expression of 2308 genes for 88 observations: 29 cases of Ewing sarcoma (EWS), 11 cases of Burkitt lymphoma (BL), 18 cases of neuroblastoma (NB), 25 cases of rhabdomyosarcoma (RMS), and 5 other (non-SRBCT) samples.

Source

The data are described in Khan et al. (2001). Note that the values in khan.data$x are logarithmized (using natural log) for normalization.

References


Examples

# load sda library
library("sda")

# load full Khan et al (2001) data set
data(khan2001)
dim(khan2001$x) # 88 2308
hist(khan2001$x)
khan2001$y # 5 levels

# data set containing the SRBCT samples
get.srbct = function()
{
  data(khan2001)
  idx = which( khan2001$y == "non-SRBCT" )
  x = khan2001$x[-idx,]
  y = factor(khan2001$y[-idx])
  descr = khan2001$descr[-idx]

  list(x=x, y=y, descr=descr)
}
srbct = get.srbct()
dim(srbct$x) # 83 2308
hist(srbct$x)
predict.sda

srbct$y # 4 levels

Shrinkage Discriminant Analysis 3: Prediction Step

Description

predict.sda performs class prediction.

Usage

## S3 method for class 'sda'
predict(object, Xtest, verbose=TRUE, ...)

Arguments

object
An sda fit object obtained from the function sda.

Xtest
A matrix containing the test data set. Note that the rows correspond to observations and the columns to variables.

verbose
Report shrinkage intensities (sda) and number of used features (predict.sda).

...
Additional arguments for generic predict.

Value

predict.sda predicts class probabilities for each test sample and returns a list with two components:

class
a factor with the most probable class assignment for each test sample, and

posterior
a matrix containing the respective class posterior probabilities.

Author(s)


See Also

sda, sda.ranking.

Examples

# see the examples at the "sda" help page
Shrinkage Discriminant Analysis 2: Training Step

**Description**

`sda` trains a LDA or DDA classifier using James-Stein-type shrinkage estimation.

**Usage**

```r
sda(Xtrain, L, lambda, lambda.var, lambda.freqs, diagonal=FALSE, verbose=TRUE)
```

**Arguments**

- `Xtrain`: A matrix containing the training data set. Note that the rows correspond to observations and the columns to variables.
- `L`: A factor with the class labels of the training samples.
- `lambda`: Shrinkage intensity for the correlation matrix. If not specified it is estimated from the data. `lambda=0` implies no shrinkage and `lambda=1` complete shrinkage.
- `lambda.var`: Shrinkage intensity for the variances. If not specified it is estimated from the data. `lambda.var=0` implies no shrinkage and `lambda.var=1` complete shrinkage.
- `lambda.freqs`: Shrinkage intensity for the frequencies. If not specified it is estimated from the data. `lambda.freqs=0` implies no shrinkage (i.e. empirical frequencies) and `lambda.freqs=1` complete shrinkage (i.e. uniform frequencies).
- `diagonal`: Chooses between LDA (default, `diagonal=FALSE`) and DDA (`diagonal=TRUE`).
- `verbose`: Print out some info while computing.

**Details**

In order to train the LDA or DDA classifier, three separate shrinkage estimators are employed:

- class frequencies: the estimator `freqs.shrink` from Hausser and Strimmer (2008),
- variances: the estimator `var.shrink` from Opgen-Rhein and Strimmer (2007),
- correlations: the estimator `cor.shrink` from Schafer and Strimmer (2005).

Note that the three corresponding regularization parameters are obtained analytically without resorting to computer intensive resampling.

**Value**

`sda` trains the classifier and returns an `sda` object with the following components needed for the subsequent prediction:

- `regularization`: a vector containing the three estimated shrinkage intensities,
- `freqs`: the estimated class frequencies,
- `alpha`: vector containing the intercepts used for prediction,
- `beta`: matrix containing the coefficients used for prediction.
Author(s)


References


See Also

predict.sda, sda.ranking, freqs.shrink, var.shrink, invcor.shrink.

Examples

# load sda library
library("sda")

# training and test data

get.srbct = function()
{
  data(khan2001)
  idx = which(khan2001$y == "non-SRBCT")
  x = khan2001$x[-idx,]
  y = factor(khan2001$y[-idx])
  descr = khan2001$descr[-idx]

  list(x=x, y=y, descr=descr)
}
srbct = get.srbct()

# training data
Xtrain = srbct$x[1:63,]
Ytrain = srbct$y[1:63]
Xtest = srbct$x[64:83,]
Ytest = srbct$y[64:83]

# classification with correlation (shrinkage LDA)

sda.fit = sda(Xtrain, Ytrain)
ynew = predict(sda.fit, Xtest)$class # using all 2308 features
sum(ynew != Ytest)

# classification with diagonal covariance (shrinkage DDA) #
sda.ranking

Shrinkage Discriminant Analysis I: Predictor Ranking

Description

sda.ranking determines a ranking of predictors by computing CAT scores (correlation-adjusted t-scores) between the group centroids and the pooled mean.

plot.sda.ranking provides a graphical visualization of the top ranking features.

Usage

sda.ranking(Xtrain, L, lambda, lambda.var, lambda.freqs,
ranking.score=c("entropy", "avg", "max"),
diagonal=FALSE, fdr=TRUE, plot.fdr=FALSE, verbose=TRUE)
## S3 method for class 'sda.ranking'
plot(x, top=40, arrow.col="blue", zeroaxis.col="red",
ylab="Features", main, ...)

Arguments

Xtrain   A matrix containing the training data set. Note that the rows correspond to observations and the columns to variables.

L        A factor with the class labels of the training samples.

lambda   Shrinkage intensity for the correlation matrix. If not specified it is estimated from the data. lambda=0 implies no shrinkage and lambda=1 complete shrinkage.

lambda.var Shrinkage intensity for the variances. If not specified it is estimated from the data. lambda.var=0 implies no shrinkage and lambda.var=1 complete shrinkage.

lambda.freqs Shrinkage intensity for the frequencies. If not specified it is estimated from the data. lambda.freqs=0 implies no shrinkage (i.e. empirical frequencies) and lambda.freqs=1 complete shrinkage (i.e. uniform frequencies).

diagonal   Chooses between LDA (default, diagonal=FALSE) and DDA (diagonal=TRUE).

ranking.score how to compute the summary score for each variable from the CAT scores of all classes - see Details.
sda.ranking

fdr
plot.fdr
derive compute FDR values and HC scores for each feature.
Show plot with estimated FDR values.

verbose

x
The number of top-ranking features shown in the plot (default: 40).

top
arrow.col
Color of the arrows in the plot (default is "blue").
zeroaxis.col
Color for the center zero axis (default is "red").

ylab
Label written next to feature list (default is "Features").

main
Main title (if missing, "The", top, "Top Ranking Features" is used).

... Other options passed on to generic plot().

Details

For each predictor variable and centroid a shrinkage CAT scores of the mean versus the pooled mean is computed. If there are only two classes the CAT score vs. the pooled mean reduces to the CAT score between the two group means. Moreover, in the diagonal case (LDA) the (shrinkage) CAT score reduces to the (shrinkage) t-score.

The overall ranking of a feature is determine by computing a summary score from the CAT scores. This is controlled by the option ranking.score. The default setting (ranking.score="entropy") uses mutual information between the response and the respective predictors (ranking.score) for ranking. This is equivalent to a weighted sum of squared CAT scores across the classes. Another possibility is to employ the average of the squared CAT scores for ranking (as suggested in Ahdesmäki and Strimmer 2010) by setting ranking.score="avg". A third option is to use the maximum of the squared CAT scores across groups (similarly as in the PAM algorithm) via setting ranking.score="max". Note that in the case of two classes all three options are equivalent and lead to identical scores. Thus, the choice of ranking.score is important only in the multi-class setting. In the two-class case the features are simply ranked according to the (shrinkage) squared CAT-scores (or t-scores, if there is no correlation among predictors).

The current default approach is to use ranking by mutual information (i.e. relative entropy between full model vs. model without predictor) and to use shrinkage estimators of frequencies. In order to reproduce exactly the ranking computed by previous versions (1.1.0 to 1.3.0) of the sda package set the options ranking.score="avg" and lambda.freqs=0.

Calling sda.ranking is step 1 in a classification analysis with the sda package. Steps 2 and 3 are sda and predict.sda


Value

sda.ranking returns a matrix with the following columns:

idx original feature number
score sum of the squared CAT scores across groups - this determines the overall ranking of a feature
cat for each group and feature the cat score of the centroid versus the pooled mean
If fdr=TRUE then additionally local false discovery rate (FDR) values as well as higher criticism (HC) scores are computed for each feature (using fdrtool).

Author(s)

References

See Also
catscore, sda, predict.sda.

Examples

```r
# load sda library
library("sda")

############################
# training data #
############################

# prostate cancer set
data(singh2002)

# training data
Xtrain = singh2002$x
Ytrain = singh2002$y

############################
# feature ranking (diagonal covariance) #
############################

# ranking using t-scores (DDA)
ranking.DDA = sda.ranking(Xtrain, Ytrain, diagonal=TRUE)
ranking.DDA[1:10,]

# plot t-scores for the top 40 genes
plot(ranking.DDA, top=40)

# number of features with local FDR < 0.8
```

# (i.e. features useful for prediction)
sum(ranking.DDA[,"lfdr"] < 0.8)

# number of features with local FDR < 0.2
# (i.e. significant non-null features)
sum(ranking.DDA[,"lfdr"] < 0.2)

# optimal feature set according to HC score
plot(ranking.DDA[,c("HC","lfdr")], type="l")

which.max( ranking.DDA[1:1000,"HC"] )

# feature ranking (full covariance) #
# ranking using CAT-scores (LDA)
ranking.LDA = sda.ranking(Xtrain, Ytrain, diagonal=FALSE)
ranking.LDA[1:1000,]

# plot t-scores for the top 40 genes
plot(ranking.LDA, top=40)

# number of features with local FDR < 0.8
# (i.e. features useful for prediction)
sum(ranking.LDA[,"lfdr"] < 0.8)

# number of features with local FDR < 0.2
# (i.e. significant non-null features)
sum(ranking.LDA[,"lfdr"] < 0.2)

# optimal feature set according to HC score
plot(ranking.LDA[,c("HC","lfdr")], type="l")
which.max( ranking.LDA[1:1000,"HC"] )

--------------------------------------------------------------------------------

singh2002 Prostate Cancer Study of Singh et al. (2002)

Description

Gene expression data (6033 genes for 102 samples) from the microarray study of Singh et al. (2002).

Usage

data(singh2002)
**Format**

singh2002$x is a 102 x 6033 matrix containing the expression levels. The rows contain the samples and the columns the genes.

singh2002$y is a factor containing the diagnosis for each sample ("cancer" or "healthy").

**Details**

This data set contains measurements of the gene expression of 6033 genes for 102 observations: 52 prostate cancer patients and 50 healthy men.

**Source**

The data are described in Singh et al. (2001) and are provided in exactly the form as used by Efron (2008).

**References**


**Examples**

```r
# load sda library
library("sda")

# load Singh et al (2001) data set
data(singh2002)
dim(singh2002$x) # 102 6033
hist(singh2002$x)
singh2002$y # 2 levels
```
Index

* datasets
  khan2001, 6
  singh2002, 14

* multivariate
  catscore, 3
  centroids, 5
  predict.sda, 8
  sda, 9
  sda-package, 2
  sda.ranking, 11

carscore, 4
catscore, 2, 3, 13
centroids, 5
cor.shrink, 9

dfdtool, 13
freqs.shrink, 5, 9, 10

inv.cor.shrink, 10

khan2001, 6

log, 7

plot.sda.ranking (sda.ranking), 11
powcor.shrink, 6
predict.sda, 2, 8, 10, 12, 13

sda, 2, 8, 9, 12, 13
sda-package, 2
sda.ranking, 2, 4, 8, 10, 11
singh2002, 14

var.shrink, 5, 6, 9, 10