

# Package ‘sda’

April 19, 2009

**Version** 1.1.0

**Date** 2009-03-12

**Title** Shrinkage Discriminant Analysis and Feature Selection

**Author** Miika Ahdesmaki and Korbinian Strimmer

**Maintainer** Korbinian Strimmer <strimmer@uni-leipzig.de>

**Depends** R (>= 2.7.0), lattice, entropy (>= 1.1.3), corpcor (>= 1.5.2), fdrtool (>= 1.2.5)

## Suggests

**Description** This package provides an efficient framework for high-dimensional linear and diagonal discriminant analysis with feature selection. The classifier is trained using Stein-type shrinkage estimators and features are ranked using correlation-adjusted t-scores. Feature selection is controlled using false non-discovery rates or higher criticism scores.

**License** GPL (>= 3)

**URL** <http://strimmerlab.org/software/sda/>

**Repository** CRAN

**Date/Publication** 2009-03-12 20:07:08

## R topics documented:

sda-package . . . . .	2
centroids . . . . .	2
khan2001 . . . . .	4
predict.sda . . . . .	5
sda . . . . .	6
sda.ranking . . . . .	8
singh2002 . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

sda-package

*The sda package*

---

### Description

This package performs linear and diagonal discriminant analysis with variable selection.

The classifier is trained using James-Stein-type shrinkage estimators. Variable selection is based on a univariate test statistics (multi-class cat score) and subsequent false non-discovery rate (FNDR) testing. In addition, higher criticism (HC) feature scores are computed and can also be used for feature thresholding.

This approach is particularly suited for high-dimensional classification. For details see Ahdesmäki and Strimmer (2009).

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 and Korbinian Strimmer (<http://strimmerlab.org/>)

### References

See website: <http://strimmerlab.org/software/sda/>

### See Also

`sda.ranking`, `sda`, `predict.sda`.

---

centroids

*Group Centroids, (Pooled) Variances, and Powers of the Pooled Correlation Matrix*

---

### Description

`centroids` computes group centroids and optionally the pooled mean and pooled variance, the group specific variances, and powers of the pooled correlation matrix.

### Usage

```
centroids(x, L, mean.pooled=FALSE, var.pooled=TRUE, var.groups=FALSE,
  powcor.pooled=FALSE, alpha=1, shrink=FALSE, verbose=TRUE)
```

**Arguments**

<code>x</code>	A matrix containing the data set. Note that the rows are sample observations and the columns are variables.
<code>L</code>	A factor with the group labels.
<code>mean.pooled</code>	Estimate the pooled mean.
<code>var.pooled</code>	Estimate the pooled variances.
<code>var.groups</code>	Estimate all group-specific variances.
<code>powcor.pooled</code>	Estimate pooled correlation matrix (taken to the power of <code>alpha</code> ).
<code>alpha</code>	exponent for the pooled correlation matrix (default: <code>alpha=1</code> ).
<code>shrink</code>	Use empirical or shrinkage estimator.
<code>verbose</code>	Provide some messages while computing.

**Details**

If option `shrink=TRUE` then the shrinkage estimators `var.shrink` from Opgen-Rhein and Strimmer (2007) and `cor.shrink` from Schäfer and Strimmer (2005) are used.

**Value**

`centroids` returns a list with the following components:

<code>samples</code>	a vector containing the samples sizes in each group,
<code>means</code>	the empirical group means,
<code>mean.pooled</code>	the pooled empirical mean,
<code>var.pooled</code>	a vector containing the pooled variances,
<code>var.groups</code>	a matrix containing the group-specific variances, and
<code>powcor.pooled</code>	a matrix containing the pooled correlation matrix to the power of <code>alpha</code> (if all correlations are zero a vector containing only the is returned to save space).
<code>alpha</code>	exponent for the pooled correlation matrix.

**Author(s)**

Korbinian Strimmer (<http://strimmerlab.org>).

**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)

## show pooled mean
centroids(X, Y, mean.pooled=TRUE)

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

## and inverse pooled correlation
centroids(X, Y, var.groups=TRUE, powcor.pooled=TRUE, alpha=-1)

## use shrinkage estimator for variances and correlations
centroids(X, Y, var.groups=TRUE, powcor.pooled=TRUE, alpha=-1, shrink=TRUE)

```

---

khan2001

*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 original 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) and can be obtained from [http://cbbp.thep.lu.se/pub/Preprints/01/lu\\_tp\\_01\\_06\\_supp.html](http://cbbp.thep.lu.se/pub/Preprints/01/lu_tp_01_06_supp.html). Note that the values in `khan.data$x` are additionally logarithmized (using natural `log`) for normalization.

**References**

Khan et al. 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine* 7:673–679.

**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)
srbct$y # 4 levels
```

---

predict.sda

*Shrinkage Discriminant Analysis 3: Prediction Step*

---

**Description**

`predict.sda` performs class prediction.

**Usage**

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

**Arguments**

<code>object</code>	An <code>sda</code> fit object obtained from the function <code>sda</code> .
<code>Xtest</code>	A matrix containing the test data set. Note that the rows correspond to observations and the columns to variables.
<code>feature.idx</code>	A vector indicating which features to employ for prediction (if unspecified all features will be used).
<code>verbose</code>	Report shrinkage intensities ( <code>sda</code> ) and number of used features ( <code>predict.sda</code> ).
<code>...</code>	Additional arguments for generic <code>predict</code> .

**Value**

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

<code>class</code>	a factor with the most probable class assignment for each test sample, and
<code>posterior</code>	a matrix containing the respective class posterior probabilities.

**Author(s)**

Miika Ahdesmäki and Korbinian Strimmer (<http://strimmerlab.org>).

**See Also**

[sda](#), [sda.ranking](#).

**Examples**

```
# see the examples at the "sda" help page
```

---

sda

*Shrinkage Discriminant Analysis 2: Training Step*

---

**Description**

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

**Usage**

```
sda(Xtrain, L, diagonal=FALSE, verbose=TRUE)
```

**Arguments**

<code>Xtrain</code>	A matrix containing the training data set. Note that the rows correspond to observations and the columns to variables.
<code>L</code>	A factor with the class labels of the training samples.
<code>diagonal</code>	Chooses between LDA (default, <code>diagonal=FALSE</code> ) and DDA ( <code>diagonal=TRUE</code> ).
<code>verbose</code>	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 Schäfer 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,
prior               the estimated class frequencies,
predcoef           matrix containing the coefficients used for prediction, and
```

## Author(s)

Miika Ahdesmäki and Korbinian Strimmer (<http://strimmerlab.org>).

## References

Ahdesmäki, A., and K. Strimmer. 2009. Feature selection in "omics" prediction problems using cat scores and false non-discovery rate control. See <http://arxiv.org/abs/0903.2003> for publication details.

## See Also

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

## Examples

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

#####
# training and test data #
#####

# 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()

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

#####
# for complete example scripts illustrating classification with #
# feature selection visit http://strimmerlab.org/software/sda/ #
#####

```

---

sda.ranking

*Shrinkage Discriminant Analysis 1: Feature Ranking*


---

## Description

sda.ranking determines a ranking of features by computing cat 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, diagonal=FALSE, fdr=TRUE, plot.fdr=FALSE, verbose=TRUE)
## S3 method for class 'sda.ranking':
plot(x, top=40, ...)

```

**Arguments**

<code>Xtrain</code>	A matrix containing the training data set. Note that the rows correspond to observations and the columns to variables.
<code>L</code>	A factor with the class labels of the training samples.
<code>diagonal</code>	Chooses between LDA (default, <code>diagonal=FALSE</code> ) and DDA ( <code>diagonal=TRUE</code> ).
<code>fdr</code>	compute FDR values and HC scores for each feature.
<code>plot.fdr</code>	Show plot with estimated FDR values.
<code>verbose</code>	Print out some info while computing.
<code>x</code>	An "sda.ranking" object – this is produced by the <code>sda.ranking()</code> function.
<code>top</code>	The number of top-ranking features shown in the plot (default: 40).
<code>...</code>	Additional arguments for generic plot.

**Details**

For each feature and centroid a shrinkage cat scores of the mean versus the pooled mean is computed. The overall ranking of a feature is determined by the sum of the squared cat scores across all centroids. For the diagonal case (LDA) the cat score reduce to the t-score. Thus in the two-class diagonal case the feature are simply ranked according to the (shrinkage) t-scores.

Calling `sda.ranking` should be step 1 in a classification analysis. Steps 2 and 3 are [sda](#) and [predict.sda](#)

See Ahdesmäki and Strimmer (2009) for details. For the case of two classes see Zuber and Strimmer (2009).

**Value**

`sda.ranking` returns a matrix with the following columns:

<code>idx</code>	original feature number
<code>score</code>	sum of the squared cat scores - this determines the overall ranking
<code>cat</code>	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)**

Miiika Ahdesmäki and Korbinian Strimmer (<http://strimmerlab.org>).

**References**

Ahdesmäki, A., and K. Strimmer. 2009. Feature selection in "omics" prediction problems using cat scores and false non-discovery rate control. See <http://arxiv.org/abs/0903.2003> for publication details.

Zuber, V., and K. Strimmer. 2009. Gene ranking and biomarker discovery under correlation. See <http://arxiv.org/abs/0902.0751> for publication details.

**See Also**

[sda](#), [predict.sda](#).

**Examples**

```
# 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[,"HC"], 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[1:10,]

# 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[, "HC"], 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) - see <http://www-stat.stanford.edu/~ckirby/brad/papers/Ebaydata.R>.

### References

D. Singh et al. 2002. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* 1:203–209.

Efron, B. 2008. Empirical Bayes estimates for large-scale prediction problems. Technical Report, Stanford University.

**Examples**

```
# 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

## \*Topic **datasets**

khan2001, 4

singh2002, 11

## \*Topic **multivariate**

centroids, 2

predict.sda, 5

sda, 6

sda-package, 2

sda.ranking, 8

centroids, 2

cor.shrink, 3, 7

fdrtool, 9

freqs.shrink, 7

invcor.shrink, 7

khan2001, 4

log, 5

plot.sda.ranking(*sda.ranking*), 8

powcor.shrink, 3

predict.sda, 2, 5, 7, 9, 10

sda, 2, 6, 6, 9, 10

sda-package, 2

sda.ranking, 2, 6, 7, 8

singh2002, 11

var.shrink, 3, 7