

Package ‘SDDA’

April 17, 2009

Date 2009-2-9

Title Stepwise Diagonal Discriminant Analysis

Version 1.0-4

Author CSIRO Bioinformatics, Glenn Stone

Description Stepwise Diagonal Discriminant Analysis - a fast algorithm for building multivariate classifiers

Maintainer David Clifford <David.Clifford@csiro.au>

License file LICENSE

Depends MASS

Repository CRAN

Date/Publication 2009-02-09 08:06:29

R topics documented:

plotdiag	2
predict.sdda	2
print.sdda	3
sdda	3
summary.sdda	5
which.genes	5
xvalidate	6
Index	7

plotdiag *Diagnostic Plots for SDDA*

Description

Diagnostic Plots for SDDA to check the choice of the number of variables.

Usage

```
plotdiag(obj)
```

Arguments

obj Output from sdda

Author(s)

Glenn Stone

predict.sdda *Predict functionality for stepwise diagonal discriminant analysis*

Description

Produces predicted classes or class probabilities for an sdda rule.

Usage

```
predict.sdda(object, newdata, ...)
```

Arguments

object A fitted sdda rule object.

newdata A dataset from which we wish to make predictions. Unlike many predict methods this is required, as sdda does not store the original data due to potential memory problems.

... parameters passed to the based dlda or dqda predict method. Specify type to be "class" or "prob" to return the predicted class or the predicted probabilities for each class, by default the predicted class is returned

Value

The predicted class (by default), or the probabilities of being from each class

Author(s)

Glenn Stone

```
print.sdda
```

Print SDDA output to screen

Description

Summary statistics for an SDDA object are printed to the screen including the number of samples, the number of variables, the number of groups and their names etc. Pass plot=TRUE to get plot diagnostics also.

Usage

```
print.sdda(x, plot=FALSE, ...)
```

Arguments

x	output from SDDA
plot	Should we plot some diagnostics?
...	additional parameters as needed

Author(s)

David Clifford

```
sdda
```

Stepwise Diagonal Discriminant Analysis

Description

Uses a forwards stepwise strategy to add variables one at a time to a diagonal discriminant rule.

Usage

```
sdda(X, y, priors, start = rep(FALSE, ncol(X)), never = rep(FALSE, ncol(X)), method=
```

Arguments

X	Training data matrix - rows are observations, columns are variables.
y	A factor of true class labels, or a numeric vector with values 1, 2, 3, ... G where G is the number of classes.
priors	Prior probabilities for the different classes, if left unspecified these default to equal probability to belong to each group
start	Logical vector indicating which variables (if any) should start in the rule.
never	Logical vector indicating which variables (if any) should never be in the rule.
method	Choose linear or quadratic discriminant analysis, "lda" or "qda"
...	Other terms as appropriate

Details

Uses a forwards stepwise strategy to add variables one at a time to a diagonal discriminant rule. At each step the variable that most decreases the (leave-one-out) cross-validated error rate is added. Adding stops when the cross-validated error rate cannot be decreased.

Value

means	The means for each variable within each group
vars	Under DLDA (default) this is a vector of the variances of each variable, under DQDA this is a matrix of variances for each variable within each group
counts	The number of samples in each group
priors	The prior probability of being in each group
S	Vector of flags of the variables in the rule, including <code>start</code>
ecrit	X-validated error rate criterion. Used for diagnostics, see plotdiag
pcrit	X-validated likelihood criterion. Used for diagnostics, see plotdiag

Author(s)

Glenn Stone

See Also

[xvalidate](#), [predict.sdda](#), [plotdiag](#)

Examples

```
data(iris)
Y <- iris[,5]
noise <- matrix(rnorm(50*nrow(iris)),nrow=nrow(iris))
colnames(noise) = paste("noise",1:ncol(noise),sep="")
X <- cbind(as.matrix(iris[,1:4]),noise)
dim(X)

s1 <- sdda(X,Y)

which.genes(s1)
plotdiag(s1)
summary(s1)

## Probabilities for each class
predict(s1,newdata=X,type="prob")
rs <- predict(s1,X)
table(Y,rs)

## Leave-one-out crossvalidation
xv <- xvalidate(X,Y,trace=TRUE)
table(Y,xv)
```

summary.sdda *Summarise SDDA output*

Description

Summarise output from SDDA

Usage

```
summary.sdda(object, ...)
```

Arguments

object Output from sdda
... Additional parameters as needed

Author(s)

David Clifford

which.genes *Print list of genes chosen by SDDA*

Description

This function processes the sdda output to list the genes chosen.

Usage

```
which.genes(obj)
```

Arguments

obj output from sdda

Author(s)

Glenn Stone

xvalidate

Produces cross validated fitted values for sdda.

Description

Ten-fold cross-validation, or leave one out cross validation (default).

Usage

```
xvalidate(X, y, method=sdda, fold = NULL, trace = FALSE, ...)
```

Arguments

X	real matrix with n rows (samples) and p columns (variables or genes)
y	vector of class labels
method	sdda is the only option in this version of the library
fold	an integer specifying the number of folds to use in the cross validation. Default value NULL gives leave one out cross validation.
trace	a logical variable with value TRUE if information about cross validation progress to be output and value FALSE if no output required.
...	Other terms as appropriate

Details

The data is divided into fold groups. The fitted value for each group is obtained by using a model built from the data for the remaining groups. For sdda the usual cross validated class labels are produced.

Value

An n by 1 vector of cross validated fitted values.

Note

Missing values in X and y are not allowed

Author(s)

Glenn Stone

Index

*Topic **models**

- plotdiag, 1
- predict.sdda, 2
- print.sdda, 2
- sdda, 3
- summary.sdda, 5
- which.genes, 5
- xvalidate, 6

*Topic **multivariate**

- plotdiag, 1
- predict.sdda, 2
- print.sdda, 2
- sdda, 3
- summary.sdda, 5
- which.genes, 5
- xvalidate, 6

*Topic **regression**

- xvalidate, 6

- plotdiag, 1, 4
- predict.sdda, 2, 4
- print.sdda, 2

- sdda, 3
- summary.sdda, 5

- which.genes, 5

- xvalidate, 4, 6