pdmclass

April 19, 2010

`pdmClass.cv` *Leave One Out Crossvalidation*

Description

This function performs a leave one out crossvalidation to estimate the accuracy of a classifier built using `pdmClass`.

Usage

```
pdmClass.cv(Y, X, method = c("pls", "pcr", "ridge"))
```

Arguments

`Y`	A vector of factors giving the class assignments for the samples to be used in the crossvalidation.
`X`	A matrix with samples in rows and observations in columns. Note that this is different than the usual paradigm for microarray data.
`method`	One of "pls", "pcr", "ridge", corresponding to partial least squares, principal components regression and ridge regression.

Details

This function performs a leave one out crossvalidation, which can be used to estimate the accuracy of a classifier. Each sample is removed in turn and a classifier is built using the remaining samples. The class of the removed sample is then predicted using the classifier. This is repeated for each sample, resulting in a vector of predicted class assignments for each sample in the original training set.

Although far from perfect, this method can be used to estimate the accuracy of a given classifier without splitting data into a training and testing set.

Value

A vector of factors giving the predicted class assignments for each of the samples in the training set. A confusion matrix can be constructed using `confusion`.

Author(s)

James W. MacDonald

References

http://www.sph.umich.edu/~ghoshd/COMPBIO/POPTSCORE

"Flexible Disriminant Analysis by Optimal Scoring" by Hastie, Tibshirani and Buja, 1994, JASA, 1255-1270.

"Penalized Discriminant Analysis" by Hastie, Buja and Tibshirani, Annals of Statistics, 1995 (in press).

Examples

```
library(fibroEset)
data(fibroEset)
y <- as.factor(pData(fibroEset)[,2])
x <- t(exprs(fibroEset))
tmp <- pdmClass.cv(y, x)
confusion(tmp, y)
```

`pdmClass` *Function to Classify Microarray Data using Penalized Discriminant Methods*

Description

This function is used to classify microarray data. Since the underlying model fit is based on penalized discriminant methods, there is no need for a pre-filtering step to reduce the number of genes.

Usage

```
pdmClass(formula , method = c("pls", "pcr", "ridge"), keep.fitted =
TRUE,  ...)
```

Arguments

`formula`	A symbolic description of the model to be fit. Details given below.
`method`	One of "pls", "pcr", "ridge", corresponding to partial least squares, principal components regression and ridge regression.
`keep.fitted`	Boolean. Should the fitted values be kept? Default is TRUE, as this is necessary for the plotting and predict functions.
`...`	Additional parameters to pass to `method` or `fda`. See `fda` for more information.

Details

The formula interface is identical to all other formula calls in R, namely Y ~ X, where Y is a numeric vector of class assignments and X is a matrix or data.frame containing the gene expression values. Note that unlike most microarray analyses, in this instance the columns of X are genes and rows are samples, so most calls will require something similar to Y ~ t(X).

Value

an object of class "fda". Use predict to extract discriminant variables, posterior probabilities or predicted class memberships. Other extractor functions are coef, and plot.

The object has the following components:

percent.explained
: the percent between-group variance explained by each dimension (relative to the total explained.)

values
: optimal scaling regresssion sum-of-squares for each dimension (see reference). The usual discriminant analysis eigenvalues are given by values / (1-values), which are used to define percent.explained.

means
: class means in the discriminant space. These are also scaled versions of the final theta's or class scores, and can be used in a subsequent call to fda (this only makes sense if some columns of theta are omitted—see the references).

theta.mod
: (internal) a class scoring matrix which allows predict to work properly.

dimension
: dimension of discriminant space.

prior
: class proportions for the training data.

fit
: fit object returned by method.

call
: the call that created this object (allowing it to be update-able)

confusion
: A 'confusion' matrix that shows how well the classifier works using the training data.

Author(s)

James W. MacDonald and Debashis Ghosh, based on fda in the mda package of Trevor Hastie and Robert Tibshirani, which was ported to R by Kurt Hornik, Brian D. Ripley, and Friedrich Leisch.

References

http://www.sph.umich.edu/~ghoshd/COMPBIO/POPTSCORE

"Flexible Disriminant Analysis by Optimal Scoring" by Hastie, Tibshirani and Buja, 1994, JASA, 1255-1270.

"Penalized Discriminant Analysis" by Hastie, Buja and Tibshirani, Annals of Statistics, 1995 (in press).

Examples

```
library(fibroEset)
data(fibroEset)
y <- as.factor(pData(fibroEset)[,2])
x <- t(exprs(fibroEset))
pdmClass(y ~ x)
```

pdmGenes *A Function to output the Top Ranked Genes from a Penalized Discriminant Classifier*

Description

After fitting a classifier, it is often desirable to output the most "interesting" genes for further validation. This function will output the top 'n' genes that discriminate between each class, along with an estimate of the stability of the observed rankings (see details for more information).

Usage

```
pdmGenes(formula = formula(data), method = c("pls", "pcr", "ridge"),
data = sys.frame(sys.parent()), weights, theta, dimension = J - 1,
eps = .Machine$double.eps, genelist = NULL, list.length = NULL, B = 100, ...)
```

Arguments

formula	A symbolic description of the model to be fit. Details given below.
method	One of "pls", "pcr", "ridge", corresponding to partial least squares, principal components regression and ridge regression.
data	An optional data.frame that contains the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which pdmClass is called. Note that unlike most microarray analyses, in this case rows are samples and columns are genes.
weights	An optional vector of sample weights. Defaults to 1.
theta	An optional matrix of class scores, typically with less than J - 1 columns.
dimension	The dimension of the solution, no greater than J - 1, where J is the number of classes. Defaults to J - 1.
eps	A threshold for excluding small discriminant variables. Defaults to .Machine$double.eps.
genelist	A vector of gene names, one per gene.
list.length	The number of 'top' genes to output.
B	The number of bootstrap samples to use for estimating stability. Defaults to 100. More than this may take an inordinate amount of time.
...	Additional parameters to pass to method.

Details

The formula interface is identical to all other formula calls in R, namely Y ~ X, where Y is a numeric vector of class assignments and X is a matrix or data.frame containing the gene expression values. Note that unlike most microarray analyses, in this instance the columns of X are genes and rows are samples, so most calls will require something similar to Y ~ t(X).

The dimension of the solution is typically J - 1, where J is the number of classes. The model fit uses contr.treatment contrasts, which means that all of the coefficients in the model are comparing the given class to a baseline class. Therefore, the genes listed are those that discriminate between a given class and the baseline. For instance, if there are three classes (characterized by a numeric vector of 1s, 2s, and 3s), then there will be two sets of 'top genes'. The first set will be those genes that discriminate between class 2 and class 1, whereas the second set will be the genes that discriminate between class 3 and class 1. The 'Y' vector will therefore need to be constructed to give the comparisons of interest.

Value

A list containing a `data.frame` for each comparison. The first column of each `data.frame` contains the gene names, and the second column contains the frequency that the gene was observed in the bootstrapped samples.

Author(s)

James W. MacDonald and Debashis Ghosh. Partial least squares and principal components regression based on code written by Mike Denham and contributed to StatLib. Model fit based on code from the `mda` package written by Trevor Hastie and Robert Tibshirani and ported to R by Kurt Hornik, Brian D. Ripley, and Friedrich Leisch.

References

http://www.sph.umich.edu/~ghoshd/COMPBIO/POPTSCORE

Examples

```
library(fibroEset)
data(fibroEset)
y <- as.factor(pData(fibroEset)[,2])
x <- t(exprs(fibroEset))
genes <- featureNames(fibroEset)
pdmGenes(y ~ x, genelist = genes, list.length = 25, B = 10)
```

`predict.pls` *Classify Observations using Penalized Discriminant Methods*

Description

These are functions that can be used to classify new samples (a test set) based on an existing classifier created using a training set.

Usage

```
## S3 method for class 'pls':
predict(object, x, ...)
## S3 method for class 'svd':
predict(object, x, ...)
```

Arguments

`object`	An object created by a call to `pdmClass`.
`x`	A matrix of new observations in which rows are samples and columns are genes. If not supplied, prediction will be performed on the original training set.
`...`	Other variables passed to predict.

Value

A vector of predicted class assignments.

Author(s)

Debashis Ghosh

References

http://www.sph.umich.edu/~ghoshd/COMPBIO/POPTSCORE

Examples

```
library(fibroEset)
data(fibroEset)
y <- as.numeric(pData(fibroEset)[,2])
x <- t(exprs(fibroEset))
genes <- featureNames(fibroEset)
tmp <- pdmClass(y ~ x)
predict(tmp)
```

Index