Package 'plstools'

November 27, 2012

Type Package

Title PLS and GPLS models based on NIPALS algorithm

Version 1.0-8

Date 2008-09-18

Author Pierre BADY <pierre.bady@unil.ch>

Maintainer Pierre BADY <pierre.bady@unil.ch>

Depends R (>= 2.15.0),ade4

Suggests ade4, boot, pls, gpls

Description PLS regression of first generation and diagnostic tools

License GPL version 2 or newer

LazyLoad yes

R topics documented:

plstools-package *PLS regression based on NIPALS algoritm*

Description

This package contains PLS regression functions based on NIPALS algoritm and some tools to evaluate the goodness of fit and the model quality.

Details

Package:	plstools
Type:	Package
Version:	1.0
Date:	2008-09-18
License:	GPL version 2 or newer
LazyLoad:	yes
Depends:	R (>= 2.7.1),ade4
Suggests:	ade4, boot

The function 'pls' use the function 'delete.intercept' from the package pls.

Author(s)

Pierre Bady <pierre.bady@unil.ch>
Maintainer: Pierre Bady <pierre.bady@unil.ch>

References

Wold H. (1966) Estimation of principal components and related models by iterative least squares. In P. Krishnaiah, editors.Multivariate Analysis, Academic Press, 391<96>420.
Wold H. (1975) Modeling in Complex Situations with Soft Information, Third World Congress of Econometric Society, August 21-26, Toronto, Canada.
Wold S., Martens H. & Wold H. (1983) <ab> The multivariate calibration problem in chemistry solved by the PLS method<bb>, in Proc. Conf. Matrix Pencils, Ruhe, A. & K<e5>gstr<f8>m, B. (Eds), March 1982, Lecture Notes in Mathematics, Springer Verlag, Heidelberg, pp. 286- 293.
Tenenhaus M.(1998) La Regression PLS. Theorie et pratique. Technip, Paris.

See Also

pls, gpls, ade4, boot

Examples

```
require(pls)
data(yarn)
yarn.pls <- plsr(density ~ NIR, 6, data = yarn, validation = "CV")
plstest1 <- plsone(density ~ NIR, nf=6, data = yarn,scale=FALSE)
cor(plstest1$th,yarn.pls$scores)
```

```
cor(fitted(plstest1),yarn.pls$fitted.values[,,6])
plot(fitted(plstest1),yarn.pls$fitted.values[,,6],pch=20,panel.first=c(grid()),cex=2)
abline(0,1,col="red")
abline(0,1,col="red",lwd=2)
plstest2 <- update(plstest1,nf=2)
plstest2
plstest1
summary(plstest1)
summary(plstest2)
plot(plstest1,which.par=1:2)
plot(plstest1,which.par=c(1:2,6))
```

---

gplsone *GPLS model based on NIPALS algorithm*

Description

Generalized Partial least squares regresssion based on NIPALS algorithm (IWPLS).

Usage

```
gplsone(formula, data,subset,weights=NULL,na.action,offset,
  family=gaussian,nf = 2,method="gplsone.fit",scale=FALSE,
  tol=1e-9,deps=1e-20,nitermax=100, ...)
gplsone.fit(y,X,nf=2,family,weights=NULL,tol=1e-9,deps=1e-20,
  nitermax=100,scale=FALSE,...)
cgplsone.fit(y,X,nf=2,family,weights=NULL,tol=1e-9,deps=1e-20,
  nitermax=100,scale=FALSE,...)
## S3 method for class 'gplsone'
residuals(object,...)
```

Arguments

object	an object of class inheriting from 'gplsone'.
formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under Details.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula)
family	a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See "family" for details of family functions.)
weights	an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector.
subset	an optional vector specifying a subset of observations to be used in the fitting process.

na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful. Value na.pass is used to take in account the missing values.
offset	this can be used to specify an a priori known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length either one or equal to the number of cases. One or more offset terms can be included in the formula instead or as well, and if both are specified their sum is used (See model.offset).
y	a numeric vector corresponding to the response
X	a numeric data.frame corresponding to explanatory variables (see model.matrix)
nf	an integer indicating the number of kept components
scale	a logical value. If scale is TRUE, X is scaled by dividing each variable by its sample standard deviation (by deafult, scale=TRUE).
method	the method to be used in fitting the model. The default method "plsone.fit" uses partial least squares (PLS). The only current alternative is "model.frame" which returns the model frame and does no fitting.
tol	positive convergence tolerance (by default, tol=1e-9).
deps	positive tolerance (used in the computation of convergence value,by default deps=1e-20).
nitermax	integer giving the maximal number of IWPLS iterations (by default, nitermax=100).
...	further arguments passed to or from other methods.

Details

The function 'plsone' is regression methods based on partail least square.
The default method 'plsone.fit' uses NIPALS algorithm.
The Coordinates ch,th, ph and ch correspond to Slopes equation in linear least squares regression without intercept term and can be compute with missing data.
+ additional details on the computation of the terms.
The function 'gplsone' is base on two functions: scalew (from the library ade4) and delete.intercept (from the library pls). For more details on these both methods, you can directly consult their documentations.
The functions 'cgplsone.fit' is based on C code and the function 'gpls' is programmed in R.

Value

The function 'gplsone' returns an object of class inheriting from 'plsone'.

glm	an object 'glm' based on the PLS components (th)
bh	an numeric vector which contains coefficients associated with the explanatory variables
y	a numeric vector corresponding to the response
x	a numeric data.frame corresponding to explanatory variables
ch	a numeric vector containing regression coefficients of y on the components 'th'
th	an object 'data.frame' containing the conponents 'th'.
uh	an object 'data.frame' containing normalised residuals for h > 1.

ph	an object 'data.frame' containing regression coefficient of the h_{th} columns of X_{h-1} on t(h).
wh	an object 'data.frame' containing regression coefficient of the h_{th} columns of X_{h-1} on y(h-1).
whx	an object 'data.frame' containing regression coefficent of th on explanatory variables.
????	

nf	an integer indicating the number of kept components.
formula	the initial formula supplied.
method	the name of the fitter function used, currently always "plsone.fit".
call	generally match.call()

Note

The function 'pls' use the function 'delete.intercept' from the package pls.

References

Marx 1996

See Also

gpls, delete.intercept, glm

Examples

```
require(pls)
data(yarn)
```

---

plot.gplsone *Plot Diagnostics for an gplsone Object*

Description

in construction

Usage

```
## S3 method for class 'gplsone'
plot(x,xax=1,yax=2,...)
```

Arguments

x	an object of class inheriting from 'gplsone'
xax	the column number for the x-axis
yax	the column number for the y-axis
...	further arguments passed to or from other methods.

Details

The selected plots are drawn on a graphics device.
1: Observed vs Expected values
2: Normal QQ-plot of residuals
3: Representation of Components th (in 2 dimensions, by default xax=1 and yax=2).
4: Prediction vs the h^{th} component (by default xax=1).
6: Representation of Weights (in 2 dimensions, by default xax=1 and yax=2).
9: Representation of correlations with the conponents (in 2 dimensions, by default xax=1 and yax=2)

Value

x is invisibly returned.

References

Tenenhaus M.(1998) La Regression PLS. Theorie et pratique. Technip, Paris.

See Also

gplsone, qqnorm, s.arrow, s.label, plot.glm

Examples

```
require(pls)
data(yarn)
```

`plot.plsone` *Plot Diagnostics for an plsone Object*

Description

Nine plots (selectable by which.plot) are currently available: Observed vs Expected values, Normal QQ-plot of residuals, Representation of components th, Prediction vs the h^{th} component, Representation of DModXN, Representation of Weights, Representation of DModYN, Representationof Hotellin T2, Representation of correlations with the conponents.

Usage

```
## S3 method for class 'plsone'
plot(x, xax = 1, yax = 2, mfrow = NULL, which.plot =1:9,...)
```

Arguments

x	an object of class inheriting from 'plsone'
xax	the column number for the x-axis
yax	the column number for the y-axis
mfrow	a vector of the form 'c(nr,nc)', otherwise computed by as special own function n2mfrow
which.plot	a numeric vector containing the numbers of The selected plots (see details)
...	further arguments passed to or from other methods.

Details

The selected plots are drawn on a graphics device.
1: Observed vs Expected values
2: Normal QQ-plot of residuals
3: Representation of Components th (in 2 dimensions, by default xax=1 and yax=2).
4: Prediction vs the h^{th} component (by default xax=1).
5: Representation of DModXN
6: Representation of Weights (in 2 dimensions, by default xax=1 and yax=2).
7: representation of DModYN
8: Representationof Hotellin T2
9: Representation of correlations with the conponents (in 2 dimensions, by default xax=1 and yax=2)

Value

x is invisibly returned.

References

Tenenhaus M.(1998) La Regression PLS. Theorie et pratique. Technip, Paris.

See Also

plsone, qqnorm, s.arrow, s.label, plot.lm

Examples

```
require(pls)
data(yarn)
plstest <- plsone(density ~ NIR, nf=6, data = yarn,scale=FALSE)
plot(plstest)
plot(plstest,which.plot=1:2)
plot(plstest,which.plot=c(1:2,6))
```

plsone *PLS regression based on NIPALS algorithm*

Description

Partial least squares regresssion based on NIPALS algorithm.

Usage

```
plsone(formula, data,subset,na.action,nf = 2,scale=TRUE,method="plsone.fit",...)
plsone.fit(y, X, nf = 2,scale=TRUE, ...)
cplsone.fit(y, X, nf = 2,scale=TRUE, ...)
## S3 method for class 'plsone'
print(x, ...)
## S3 method for class 'plsone'
coef(object,type="components", ...)
## S3 method for class 'plsone'
coefficients(object, ...)
## S3 method for class 'plsone'
residuals(object, ...)
## S3 method for class 'plsone'
fitted(object, ...)
## S3 method for class 'plsone'
formula(x, ...)
```

Arguments

x	an object of class inheriting from 'plsone'
object	an object of class inheriting from 'plsone'
formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under Details.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula)
subset	an optional vector specifying a subset of observations to be used in the fitting process.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful. Value na.pass is used to take in account the missing values.
y	a numeric vector corresponding to the response
X	a numeric data.frame corresponding to explanatory variables (see model.matrix)
nf	an integer indicating the number of kept components
scale	a logical value. If scale is TRUE, X is scaled by dividing each variable by its sample standard deviation (by deafult, scale=TRUE).

method	the method to be used in fitting the model. The default method "plsone.fit" uses partial least squares (PLS). The only current alternative is "model.frame" which returns the model frame and does no fitting.
type	the type of coefficient. The default "components" retruns the coefficients of regression of y on the components. The alternative "varaibles" gives coefficients based on initial explanatory variables
...	further arguments passed to or from other methods.

Details

The function 'plsone' is regression methods based on partail least square.
The default method 'plsone.fit' uses NIPALS algorithm.
The Coordinates ch,th, ph and ch correspond to Slopes equation in linear least squares regression without intercept term and can be compute with missing data.
+ additional details on the computation of the terms.
+ The library propose two functuion to fit model: plsone.fit programmed only in R and cplsone.fit based on C programs.
The function plsone is base on the function delete.intercept from the library 'pls'.

Value

The function 'plsone' returns an object of class inheriting from 'plsone'.

lm	an object 'lm' based on the PLS components (th)
bh	an numeric vector which contains coefficients associated with the explanatory variables
y	a numeric vector corresponding to the response
x	a numeric data.frame corresponding to explanatory variables
ch	a numeric vector containing regression coefficients of y on the components 'th'
th	an object 'data.frame' containing the conponents 'th'.
uh	an object 'data.frame' containing normalised residuals for h > 1.
ph	an object 'data.frame' containing regression coefficient of the h_{th} columns of X_{h-1} on t(h).
wh	an object 'data.frame' containing regression coefficient of the h_{th} columns of X_{h-1} on y(h-1).
whx	an object 'data.frame' containing regression coefficent of th on explanatory variables.
Q2	a numeric vector conatining the criteria Q2 given by: $Q2 = 1 - PRESS_h / RSS_{h-1}$.
nf	an integer indicating the number of kept components.
formula	the initial formula supplied.
method	the name of the fitter function used, currently always "plsone.fit". The C version is called "cplsone.fit".
call	generally match.call()

Note

The function 'pls' use the function 'delete.intercept' from the package pls.

References

Wold H. (1966) Estimation of principal components and related models by iterative least squares. In P. Krishnaiah, editors.Multivariate Analysis, Academic Press, 391<96>420.
Wold H. (1975) Modeling in Complex Situations with Soft Information, Third World Congress of Econometric Society, August 21-26, Toronto, Canada.
Wold S., Martens H. & Wold H. (1983) <ab> The multivariate calibration problem in chemistry solved by the PLS method<bb>, in Proc. Conf. Matrix Pencils, Ruhe, A. & K<e5>gstr<f8>m, B. (Eds), March 1982, Lecture Notes in Mathematics, Springer Verlag, Heidelberg, pp. 286- 293.
Tenenhaus M.(1998) La Regression PLS. Theorie et pratique. Technip, Paris.

See Also

plsr, delete.intercept, mvr, lm

Examples

```
require(pls)
data(yarn)
yarn.pls <- plsr(density ~ NIR, 6, data = yarn, validation = "CV")
plstest1 <- plsone(density ~ NIR, nf=6, data = yarn,scale=FALSE)
cor(plstest1$th,yarn.pls$scores)
cor(fitted(plstest1),yarn.pls$fitted.values[,,6])
plot(fitted(plstest1),yarn.pls$fitted.values[,,6],pch=20,panel.first=c(grid()),cex=2)
abline(0,1,col="red",lwd=2)
# update fonction
plstest2 <- update(plstest1,nf=2)
plstest2
plstest1
# with missing data
data(yarn)
yarn$NIR[sample(1:(268*28),300)] <- rep(NA,300)
plsna1 <- plsone(density ~ NIR, nf=6, data = yarn,scale=FALSE,na.action=na.pass)
plot(fitted(plstest1),fitted(plsna1),pch=20,panel.first=c(grid()),cex=2)
abline(0,1,col="red",lwd=2)
```

---

predict.gplsone *Predict method for Generalized Partial Least Squares Fits*

Description

The function 'predict.gplsone'retruns predicted values, confidence and tolerance intervals.

Usage

```
## S3 method for class 'gplsone'
predict(object,newdata=NULL,type="link",se.fit=FALSE,interval="none",level=0.95,...)
```

Arguments

object	Object of class inheriting from 'plsone'
newdata	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
type	the type of prediction required. The default is on the scale of the linear predictors; the alternative "response" is on the scale of the response variable. Thus for a default binomial model the default predictions are of log-odds (probabilities on logit scale) and type = "response" gives the predicted probabilities. The "terms" option returns a matrix giving the fitted values of each term in the model formula on the linear predictor scale. The value of this argument can be abbreviated.
se.fit	A switch indicating if standard errors are required.
interval	none, confidence, prediction
level	Tolerance/confidence level
...	further arguments passed to or from other methods.

Details

add formula for tolerance and confidence interval

Value

The function 'predict.plsone'retruns ...

comp1	Description of 'comp1'
comp2	Description of 'comp2'

...

References

Tenenhaus M. (1998) La Regression PLS. Theorie et pratique. Technip, Paris.

See Also

gplsone, predict.glm

Examples

```
require(pls)
data(yarn)
```

predict.plsone *Predict method for Partial Least Squares Fits*

Description

The function 'predict.plsone'retruns predicted values, confidence and tolerance intervals.

Usage

```
## S3 method for class 'plsone'
predict(object, newdata, se.fit = FALSE,interval="none",level = 0.95,...)
```

Arguments

object	Object of class inheriting from 'plsone'
newdata	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
se.fit	A switch indicating if standard errors are required.
interval	none, confidence, prediction
level	Tolerance/confidence level
...	further arguments passed to or from other methods.

Details

add formula for tolerance and confidence interval

Value

The function 'predict.plsone'retruns ...

comp1	Description of 'comp1'
comp2	Description of 'comp2'

...

References

Tenenhaus M. (1998) La Regression PLS. Theorie et pratique. Technip, Paris.

See Also

plsone, predict.lm

Examples

```
require(pls)
data(yarn)
yarn.pls <- plsr(density ~ NIR, 6, data = yarn, validation = "CV")
plstest1 <- plsone(density ~ NIR, nf=6, data = yarn,scale=FALSE)
plot(predict(plstest1)$fitted,plstest1$y)
```

summary.gplsone *Summarizing Generalized Partial Least Squares model*

Description

This function provides additional elements to evaluate the goodness of fit and the quality of the model (object of class inheriting from 'gplsone').

Usage

```
## S3 method for class 'gplsone'
summary(object, ...)
## S3 method for class 'summary.gplsone'
print(x, ...)
```

Arguments

object	an object of class inheriting from 'gplsone'
x	an object of class inheriting from 'summary.gplsone'
...	further arguments passed to or from other methods.

Details

decrire le calcul des elements ...

+ additional details on the computation of each elemnts
+ At the moment this function is very slow to compute the diagnostic elements.

Value

The function 'summary.gplsone' retruns an list of class 'summary.plsone' containing several elements associated with the model diagnostic:

bh	an numeric vector which contains coefficients associated with the explanatory variables
n	observation number
p	number of explanatory varaibles.
nf	an integer indicating the number of kept components (h).

References

Marx 1996

See Also

gplsone,summary.glm

Examples

```
require(pls)
data(yarn)
```

`summary.plsone` *Summarizing Partial Least Squares regression*

Description

This function provides additional elements to evaluate the goodness of fit and the quality of the model (object of class inheriting from 'plsone').

Usage

```
## S3 method for class 'plsone'
summary(object, ...)
## S3 method for class 'summary.plsone'
print(x,digits = max(3, getOption("digits") - 3),...)
```

Arguments

`object`	an object of class inheriting from 'plsone'.
`x`	an object of class inheriting from 'summary.plsone'.
`digits`	the number of significant digits to use when printing.
`...`	further arguments passed to or from other methods.

Details

decrire le calcul des elements ...

+ additional details on the computation of each elemnts
+ At the moment this function is very slow to compute the diagnostic elements.

FT2 = nf * (n * n - 1) * qf(0.95, nf, n - nf)/(n * (n - nf))
DcritX and DcritY =

Value

The function 'summary.plsone' retruns an list of class 'summary.plsone' containing several elements associated with the model diagnostic:

`rx2`	a matrix containing the R2 between x and th
`ry2`	a matrix containing the R2 between y and th
`T2`	a numeric vector containing Hotelling's T2 for each observation.
`Ts2`	a numeric vector containing adapted T2 for each observation (statistic used in SIMCA-P).

FT2	a numerical vector containing threshold to detected outlier with Hotelling's T2 (F-value).
rxt2	a matrix containing the correlations between x and th
ryt2	a matrix containing the correlations between y and th
DModX	a numeric vector containing distance from the i^{th} observation to the model in the space of the explanatory variables
DModXN	a numeric vector containing normaliszed values of DModX.
DcritX	a numerical vector containing threshold to detected the observations badly rebuilt by the model.
DModY	a numeric vector containing residuals of the regression of y on the 'nf' components.
DModYN	a numeric vector containing normaliszed values of DModY.
DcritY	a numerical vector containing threshold to detected the observations badly rebuilt by the model.
coefficients	a numeric vector containing regression coefficients of y on the components 'th'
r.squared	a numerical value corresponding to the R squared of linear model ('fraction of variance explained by the model',see 'summary.lm').
adj.r.squared	a numerical value corresponding to the adjusted R squared of linear model
bh	an numeric vector which contains coefficients associated with the explanatory variables
n	observation number
p	number of explanatory varaibles.
nf	an integer indicating the number of kept components (h).

References

Tenenhaus M. (1998) La Regression PLS. Theorie et pratique. Technip, Paris.

See Also

plsone,summary.lm

Examples

```
require(pls)
data(yarn)
plstest1 <- plsone(density ~ NIR, nf=6, data = yarn,scale=FALSE)
summary(plstest1)
summary(plstest1$lm)
```

Index