Package 'PLSRegs'

October 18, 2023

Type Package

Title Implementing and predicting regression models with several PLS algorithms

Version 0.1.0

Depends R (>= 3.5.0), fds

Author Sude Gurer, Selay Sila Doner, Mehmet Emin Sahin

Maintainer
<sudegurer5934@hotmail.com> <selaydnr@gmail.com> <m.eminsahin5334@gmail.com>

Description This package provides functions for fitting and predicting regression models with Partial Least Squares (PLS) algorithms. It includes different variations of the PLS algorithms, such as SIMPLS and Non-Orthogonal Scores PLS. Cross-validation for optimal component selection and prediction for new data are supported.

License GPL-3

Encoding UTF-8

LazyLoad yes

Imports stats

ByteCompile TRUE

R topics documented:

bidiag1_reg *Bidiag1 Regression Function*

Description

This function performs two-way regression with optional cross-validation to determine the optimal number of components for the regression model.

Usage

```
bidiag1_reg(y, x, n_comp = NULL, nfold = 5, comp_max = 10, cv = TRUE)
```

Arguments

x	A matrix containing independent variables in its columnts.
y	A vector or matrix containing the observationsof dependent variable.
n_comp	An integer value defining the number of components to be used in the PLS regression model. The default value is NULL. If NULL, then, its optimum value is determine by an n-fold cross-validation algorithm.
nfold	An integer defining the number of folds used in the cross-validation. The default value is 5.
comp_max	An integer defining the maximum number of components. The default value is 10.
cv	Logical. If TRUE, cross-validation is performed; otherwise, FALSE. The default value is TRUE.

Details

The bidiag1_reg function is a regression function that utilizes the bidiagonalization process. It applies the bidiag1 function to perform bidiagonalization and then fits a model to the data. The function can either use a predetermined number of components for the regression (specified by the n_comp parameter), or n-fold cross-validation can be used to determine the optimum value of the PLS components.

Value

Returns a list object with the following components:

coef	A matrix containing the regression coefficients.
fitted.values	A column matrix containing fitted values.
residuals	A vector containing residuals.

Author(s)

Sude Gurer, Selay Sila Doner, Mehmet Emin Sahin

References

M. Andersson (2009), "A comparison of nine PLS1 algorithms", Journal of Chemometrics, 23(10), 518-529.

Examples

```
library(fds)
data("Octanespectrum")
data("Octanevalues")

y <- Octanevalues
x <- t(Octanespectrum$y)

n_train <- 40
n_test <- 20

train_index <- sample(1:(n_train+n_test), n_train)

y_train <- y[train_index]
y_test <- y[-train_index]

x_train <- x[train_index,]
x_test <- x[-train_index,]

model <- bidiag1_reg(y_train, x_train)
predictions <- predict_pls(model, x_test)

mean((y_test - predictions)^2)
```

---

bidiag2_reg *Bidiag2 Regression Function*

Description

This function performs two-way regression with optional cross-validation to determine the optimal number of components for the regression model.

Usage

```
bidiag2_reg(y, x, n_comp = NULL, nfold = 5, comp_max = 10, cv = TRUE)
```

Arguments

x	A matrix containing independent variables in its columnts.
y	A vector or matrix containing the observationsof dependent variable.
n_comp	An integer value defining the number of components to be used in the PLS regression model. The default value is NULL. If NULL, then, its optimum value is determine by an n-fold cross-validation algorithm.
nfold	An integer defining the number of folds used in the cross-validation. The default value is 5.
comp_max	An integer defining the maximum number of components. The default value is 10.
cv	A logical item. If TRUE, cross-validation is performed; otherwise, the value n_fold provided by the users is used to perform the regression model.

Details

The bidiag2_reg function is a regression function that utilizes the bidiagonalization process. It applies the bidiag1 function to perform bidiagonalization and then fits a model to the data. The function can either use a predetermined number of components for the regression (specified by the n_comp parameter), or n-fold cross-validation can be used to determine the optimum value of the PLS components.

Value

Returns a list object with the following components:

`coef`	A matrix containing the regression coefficients.
`fitted.values`	A column matrix containing fitted values.
`residuals`	A vector containing residuals.

Author(s)

Sude Gurer, Selay Sila Doner, Mehmet Emin Sahin

References

M. Andersson (2009), "A comparison of nine PLS1 algorithms", Journal of Chemometrics, 23(10), 518-529.

Examples

```
library(fds)
data("Octanespectrum")
data("Octanevalues")

y <- Octanevalues
x <- t(Octanespectrum$y)

n_train <- 40
n_test <- 20

train_index <- sample(1:(n_train+n_test), n_train)

y_train <- y[train_index]
y_test <- y[-train_index]

x_train <- x[train_index,]
x_test <- x[-train_index,]

model <- bidiag2_reg(y_train, x_train)
predictions <- predict_pls(model, x_test)

mean((y_test - predictions)^2)
```

directscorespls_reg *Directscores Regression Function*

Description

This function conducts Direct Scores Partial Least Squares Regression. It determines the optimal number of components through cross-validation (if enabled), computes regression coefficients, generates predictions, and provides error metrics.

Usage

```
directscorespls_reg(y, x, n_comp = NULL, nfold = 5, comp_max = 10, cv = TRUE)
```

Arguments

x	A Matrix of independent variables, where each column represents an independent variable.
y	A Vector or matrix of dependent variables.
n_comp	An integer value defining the number of components to be used in the PLS regression model. The default value is NULL. If NULL, then, its optimum value is determine by an n-fold cross-validation algorithm.
nfold	An integer defining the number of folds used in the cross-validation. The default value is 5.
comp_max	An integer defining the maximum number of components. The default value is 10.
cv	A logical item. If TRUE, cross-validation is performed; otherwise, the value n_fold provided by the users is used to perform the regression model.

Details

The function utilizes the Direct Scores Partial Least Squares (PLS) algorithm for regression analysis. It can either conduct cross-validation to determine the optimal number of components or utilize a fixed number of components specified by the user.

Value

Returns a list with the following components:

coef	A Matrix containing regression coefficients.
fitted.values	A Matrix containing predicted values.
residuals	A Vector containing residuals.

Author(s)

Sude Gurer, Selay Sila Doner, Mehmet Emin Sahin

References

M. Andersson (2009), "A comparison of nine PLS1 algorithms", Journal of Chemometrics, 23(10), 518-529.

Examples

```
library(fds)
data("Octanespectrum")
data("Octanevalues")

y <- Octanevalues
x <- t(Octanespectrum$y)

n_train <- 40
n_test <- 20

train_index <- sample(1:(n_train+n_test), n_train)

y_train <- y[train_index]
y_test <- y[-train_index]

x_train <- x[train_index,]
x_test <- x[-train_index,]

model <- directscorespls_reg(y_train, x_train)
predictions <- predict_pls(model, x_test)

mean((y_test - predictions)^2)
```

---

krylovpls_reg *Krylovpls Regression Function*

Description

KrylovPLS is a regression method utilized for analyzing and predicting the relationship between independent variables and the dependent variable. The function is capable of performing cross-validation and can utilize this method to determine the optimal number of components.

Usage

```
krylovpls_reg(y, x, n_comp = NULL, nfold = 5, comp_max = 10, cv = TRUE)
```

Arguments

x	A Matrix of independent variables.
y	A Vector or matrix of dependent variables.
n_comp	An integer value defining the number of components to be used in the PLS regression model. The default value is NULL. If NULL, then, its optimum value is determine by an n-fold cross-validation algorithm.
nfold	An integer defining the number of folds used in the cross-validation. The default value is 5.
comp_max	An integer defining the maximum number of components. The default value is 10.
cv	A logical item. If TRUE, cross-validation is performed; otherwise, the value n_fold provided by the users is used to perform the regression model.

Details

This function fits a PLS regression model to the provided predictors and response variable, using the specified number of components and the Krylov subspace method. If cv = TRUE, it uses k-fold cross-validation to select the optimal number of components.

Value

Returns a list with the following components:

`coef`	A Matrix containing regression coefficients.
`fitted.values`	A Matrix containing predicted values.
`residuals`	A Vector containing residuals.

Author(s)

Sude Gurer, Selay Sila Doner, Mehmet Emin Sahin

References

M. Andersson (2009), "A comparison of nine PLS1 algorithms", Journal of Chemometrics, 23(10), 518-529.

Examples

```
library(fds)
data("Octanespectrum")
data("Octanevalues")

y <- Octanevalues
x <- t(Octanespectrum$y)

n_train <- 40
n_test <- 20

train_index <- sample(1:(n_train+n_test), n_train)

y_train <- y[train_index]
y_test <- y[-train_index]

x_train <- x[train_index,]
x_test <- x[-train_index,]

model <- simpls_reg(y_train, x_train)
predictions <- predict_pls(model, x_test)

mean((y_test - predictions)^2)
```

`nonorth_scorespls_reg` *Non-Orthogonalscores Regression Function*

Description

NonorthScoresPLS is a regression method used to analyze the relationship between independent variables and dependent variables and make predictions.

Usage

```
nonorth_scorespls_reg(y, x, n_comp = NULL, nfold = 5, comp_max = 10, cv = TRUE)
```

Arguments

`x`	A Matrix of independent variables.
`y`	A Vector or matrix of dependent variables.
`n_comp`	An integer value defining the number of components to be used in the PLS regression model. The default value is NULL. If NULL, then, its optimum value is determine by an n-fold cross-validation algorithm.
`nfold`	An integer defining the number of folds used in the cross-validation. The default value is 5.
`comp_max`	An integer defining the maximum number of components. The default value is 10.
`cv`	A logical item. If TRUE, cross-validation is performed; otherwise, the value n_fold provided by the users is used to perform the regression model.

Details

The function fits a Partial Least Squares regression model using a non-orthogonal scoring algorithm, with the possibility of cross-validation to find the optimal number of PLS components. The function returns the regression coefficients, fitted values and residuals.

Value

Returns a list with the following components:

`coef`	A Matrix containing regression coefficients.
`fitted.values`	A Matrix containing predicted values.
`residuals`	A Vector containing residuals.

Author(s)

Sude Gurer, Selay Sila Doner, Mehmet Emin Sahin

References

M. Andersson (2009), "A comparison of nine PLS1 algorithms", Journal of Chemometrics, 23(10), 518-529.

Examples

```
library(fds)
data("Octanespectrum")
data("Octanevalues")

y <- Octanevalues
x <- t(Octanespectrum$y)

n_train <- 40
n_test <- 20

train_index <- sample(1:(n_train+n_test), n_train)

y_train <- y[train_index]
y_test <- y[-train_index]

x_train <- x[train_index,]
x_test <- x[-train_index,]

model <- nonorth_scorespls_reg(y_train, x_train)
predictions <- predict_pls(model, x_test)

mean((y_test - predictions)^2)
```

---

plsf_reg *Partial Regression Function*

Description

This function establishes a Partial Least Squares Fused Lasso (PLS-FL) regression model to analyze the relationship between dependent and independent variables and to generate predictions. PLS-FL selects variables considering the structural relationships among independent variables. The function supports cross-validation and can utilize this technique to determine the optimal number of components.

Usage

```
plsf_reg(y, x, n_comp = NULL, nfold = 5, comp_max = 10, cv = TRUE)
```

Arguments

x	A Matrix of independent variables.
y	A Vector or matrix of dependent variables.
n_comp	An integer value defining the number of components to be used in the PLS regression model. The default value is NULL. If NULL, then, its optimum value is determine by an n-fold cross-validation algorithm.
nfold	An integer defining the number of folds used in the cross-validation. The default value is 5.
comp_max	An integer defining the maximum number of components. The default value is 10.
cv	A logical item. If TRUE, cross-validation is performed; otherwise, the value n_fold provided by the users is used to perform the regression model.

Details

This function fits a PLS regression model to the provided predictors and response variable,using the specified or cross-validated optimal number of components. The output includes coefficients, fitted values, and residuals.

Value

Returns a list with the following components:

`coef`	A Matrix containing regression coefficients.
`fitted.values`	A Matrix containing predicted values.
`residuals`	A Vector containing residuals.

Author(s)

Sude Gurer, Selay Sila Doner, Mehmet Emin Sahin

References

M. Andersson (2009), "A comparison of nine PLS1 algorithms", Journal of Chemometrics, 23(10), 518-529.

Examples

```
library(fds)
data("Octanespectrum")
data("Octanevalues")

y <- Octanevalues
x <- t(Octanespectrum$y)

n_train <- 40
n_test <- 20

train_index <- sample(1:(n_train+n_test), n_train)

y_train <- y[train_index]
y_test <- y[-train_index]

x_train <- x[train_index,]
x_test <- x[-train_index,]

model <- plsf_reg(y_train, x_train)
predictions <- predict_pls(model, x_test)

mean((y_test - predictions)^2)
```

plspls_reg *PLS-PLS Regression Function*

Description

This function creates a Partial Least Squares (PLS) regression model to analyze and make predictions about the relationship between dependent and independent variables. PLS-PLS selects variables by considering the structural relationships between independent variables. The function is capable of cross-validation and can use this method to determine the optimal number of components.

Usage

```
plspls_reg(y, x, n_comp = NULL, nfold = 5, comp_max = 10, cv = TRUE)
```

Arguments

x	A Matrix of independent variables.
y	A Vector or matrix of dependent variables.
n_comp	An integer value defining the number of components to be used in the PLS regression model. The default value is NULL. If NULL, then, its optimum value is determine by an n-fold cross-validation algorithm.
nfold	An integer defining the number of folds used in the cross-validation. The default value is 5.
comp_max	An integer defining the maximum number of components. The default value is 10.
cv	Logical. If TRUE, cross-validation is performed; otherwise, FALSE. The default value is TRUE.

Details

The function performs PLS regression using a method of successive orthogonalization of column vectors. The predictor and response variables should be supplied as matrices, and the number of components to be used in the regression can be optionally specified. If not specified, the number of components is determined by cross-validation.

Value

Returns a list with the following components:

coef	A Matrix containing regression coefficients.
fitted.values	A Matrix containing predicted values.
residuals	A Vector containing residuals.

Author(s)

Sude Gurer, Selay Sila Doner, Mehmet Emin Sahin

References

M. Andersson (2009), "A comparison of nine PLS1 algorithms", Journal of Chemometrics, 23(10), 518-529.

Examples

```
library(fds)
data("Octanespectrum")
data("Octanevalues")

y <- Octanevalues
x <- t(Octanespectrum$y)

n_train <- 40
n_test <- 20

train_index <- sample(1:(n_train+n_test), n_train)

y_train <- y[train_index]
y_test <- y[-train_index]

x_train <- x[train_index,]
x_test <- x[-train_index,]

model <- bidiag1_reg(y_train, x_train)
predictions <- predict_pls(model, x_test)

mean((y_test - predictions)^2)
```

---

predict_pls *Predict Function for PLS Regression*

Description

This function is used to predict the response values for new observations using a fitted PLS regression model.

Usage

```
predict_pls(object, x_new)
```

Arguments

object	A list that represents the fitted PLS regression model. This list must include an element named 'coef' representing the regression coefficients.
x_new	A matrix of predictors for the new observations.

Details

The function multiplies the matrix of new predictors by the regression coefficients from the fitted model to produce the predicted response values.

Value

A vector of predicted response values for the new observations.

Author(s)

Sude Gurer, Selay Sila Doner, Mehmet Emin Sahin

`simpls_reg` *SIMPLS Regression Function*

Description

This function creates a Simultaneous Component Analysis for Partial Least Squares regression model to analyze and make predictions about the relationship between dependent and independent variables. SIMPLS selects variables by considering the structural relationships between independent variables. The function is capable of cross-validation and can use this method to determine the optimal number of components.

Usage

```
simpls_reg(y, x, n_comp = NULL, nfold = 5, comp_max = 10, cv = TRUE)
```

Arguments

`x`	A matrix containing independent variables in its columnts.
`y`	A vector or matrix containing the observationsof dependent variable.
`n_comp`	An integer value defining the number of components to be used in the PLS regression model. The default value is NULL. If NULL, then, its optimum value is determine by an n-fold cross-validation algorithm.
`nfold`	An integer defining the number of folds used in the cross-validation. The default value is 5.
`comp_max`	An integer defining the maximum number of components. The default value is 10.
`cv`	Logical. If TRUE, cross-validation is performed; otherwise, FALSE. The default value is TRUE.

Details

The function conducts a regression analysis utilizing the SIMPLS algorithm. When 'cv' is set to TRUE, it employs cross-validation to identify the optimal number of components. The function yields the PLS model coefficients, fitted values, and residuals as outputs.

Value

Returns a list with the following components:

`coef`	A Matrix containing regression coefficients.
`fitted.values`	A Matrix containing predicted values.
`residuals`	A Vector containing residuals.

Author(s)

Sude Gurer, Selay Sila Doner, Mehmet Emin Sahin

References

M. Andersson (2009), "A comparison of nine PLS1 algorithms", Journal of Chemometrics, 23(10), 518-529.

Examples

```
library(fds)
data("Octanespectrum")
data("Octanevalues")

y <- Octanevalues
x <- t(Octanespectrum$y)

n_train <- 40
n_test <- 20

train_index <- sample(1:(n_train+n_test), n_train)

y_train <- y[train_index]
y_test <- y[-train_index]

x_train <- x[train_index,]
x_test <- x[-train_index,]

model <- simpls_reg(y_train, x_train)
predictions <- predict_pls(model, x_test)

mean((y_test - predictions)^2)
```

Index