Package 'cvq2'

February 19, 2015

Type Package

Title Calculate the predictive squared correlation coefficient

Version 1.2.0

Date 2013-10-10

Author Torsten Thalheim

Maintainer Torsten Thalheim <torstenthalheim@gmx.de>

Description The external prediction capability of quantitative structure-activity relationship (QSAR) models is often quantified using the predictive squared correlation coefficient. This value can be calculated with an external data set or by cross validation.

Depends methods, stats

License GPL-3

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2013-10-15 08:42:38

R topics documented:

cvq2-package *Calculate the predictive squared correlation coefficient.*

Description

This package compares observation with their predictions calculated by model M. It calculates the predictive squared correlation coefficient, q^2, in comparison to the well known conventional squared correlation coefficient, r^2.

Details

Package:	cvq2
Type:	Package
Version:	1.2.0
Date:	2013-10-10
Depends:	methods, stats
License:	GPL v3
LazyLoad:	yes

This package needs either a description of parameters and observations (I) or a data set that already contains the observations and their related predictions (II). In case of (I), a linear model M is generated on the fly. Afterwards, its calibration performance can be compared with its prediction power.

If the input data consist of observations and precidctions only (II), the package can be used to compute either the calibration performance or the prediction power.

If model M is generated on the fly (I), the procedure is as follows: The input data set consists of parameters $x_1, x_2, \ldots, x_n$ which describe an observation y. A linear regression (glm) of this data set yields to M. Thus the conventional squared correlation coefficient, r^2, can be calculated:

$$r^2 = 1 - \frac{\sum_{i=1}^{N} \left(y_i^{fit} - y_i\right)^2}{\sum_{i=1}^{N} \left(y_i - y_{mean}\right)^2} \equiv 1 - \frac{RSS}{SS}$$

The denominator complies with the **R**esidual **S**um of **S**quares *RSS*, the difference between the fitted values y_i^{fit} predicted by M and the observations y_i. The numerator is the **S**um of **S**quares, *SS*, and refers to the difference between the observations y_i and their mean y_{mean}.
To compare the calibration of M with its prediction power, M is applied to an external data set. External it is called, because these data have not been used during the linear regression to generate M. The comparison of the predictions y_i^{pred} with the observation y_i yields to the predictive squared

correlation coefficient, q^2:

$$q^2 = 1 - \frac{\sum_{i=1}^{N} \left(y_i^{pred} - y_i\right)^2}{\sum_{i=1}^{N} \left(y_i - y_{mean}\right)^2} \equiv 1 - \frac{PRESS}{SS}$$

The **PRE**dictive residual **S**um of **S**quares *PRESS* is the difference between the predictions y_i^{pred} and the observations y_i. The **S**um of **S**quares *SS* refers to the difference between the observations y_i and their mean y_{mean}.

In case that no external data set is available, one can perform a cross-validation to evaluate the prediction performance. The cross-validation splits the model data set (N elements) into a training set ($N - k$ elements) and a test set (k elements). Each training set yields to an individual model *M'*, which is used to predict the missing k value(s). Each model *M'* is slightly different to *M*. Thereby any observed value y_i is predicted once and the comparison between the observation and the prediction ($y_i^{pred(N-k)}$) yields to q_{cv}^2:

$$q_{cv}^2 = 1 - \frac{\sum_{i=1}^{N} \left(y_i^{pred(N-k)} - y_i\right)^2}{\sum_{i=1}^{N} \left(y_i - y_{mean}^{N-k,i}\right)^2}$$

The arithmetic mean used in this equation, $y_{mean}^{N-k,i}$, is individually for any test set and calculated for the observed values comprised in the training set.

If $k > 1$, the compilation of training and test set may have impact on the calculation of the predictive squared correlation coefficient. To overcome biasing, one can repeat this calculation with various compilations of training and test set. Thus, any observed value is predicted several times, according to the number of runs performed.

Remark, if the prediction performance is evaluated with cross-validation, the calculation of the predictive squared correlation coefficient, q^2, is more accurate than the calculation of the conventional squared correlation coefficient, r^2.

In addition to r^2 and q^2 the root-mean-square-error *rmse* is calculated to measure the accuracy of model *M*:

$$rmse = \sqrt{\frac{\sum_{i=1}^{N} \left(y_i^{pred} - y_i\right)^2}{N - \nu}}$$

The *rmse* ist the difference between a model's prediction (y_i^{pred}) and the actual observation (y_i) and can be applied for both, calibration and prediction power. It depends on the number of observations *N* and the method used to generate the model *M*. The *rmse* tends to overestimate *M*. According to Friedrich Bessel's suggestion [Upton and Cook 2008], this overestimation can be resolved while regarding the degrees of freedom, ν. Thus in case of cross-validation, $\nu = 1$ is recommended to calculate the *rmse* in relation to the prediction power. The degrees of freedom, ν, for the calculation of *rmse* regarding the prediction power can be set as parameter for `cvq2()`, `looq2()` and `q2()`. In opposite $\nu = 0$ is fixed while calculating the *rmse* in relation to the model calibration.

In case, the input is a comparison of observed and predicted values only (II), r^2 respective q^2 as well as their *rmse* are calculated immediately for these data. Neither a model *M* is generated nor a cross-validation is applied.

Note

The package development started few years ago in the Ecological Chemistry Department during my time at the Helmholtz Centre for Environmental Research in Leipzig. Thereby it is based on Schüürmann et al. 2008: External validation and prediction employing the predictive squared correlation coefficient - test set activity mean vs training set activity mean.

Author(s)

Torsten Thalheim <torstenthalheim@gmx.de>

References

1. Cramer RD III. 1980. BC(DEF) Parameters. 2. An Empirical Structure-Based Scheme for the Prediction of Some Physical Properties. *J. Am. Chem. Soc.* **102:** 1849-1859.
2. Cramer RD III, Bunce JD, Patterson DE, Frank IE. 1988. Crossvalidation, Bootstrapping, and Partial Least Squares Compared with Multiple Linear Regression in Conventional QSAR Studies. *Quant. Struct.-Act. Relat.* **1988:** 18-25.
3. Organisation for Economic Co-operation and Development. 2007. Guidance document on the validation of (quantitative) structure-activity relationship [(Q)SAR] models. *OECD Series on Testing and Assessment 69.* OECD Document ENV/JM/MONO(2007)2, pp 55 (paragraph no. 198) and 65 (Table 5.7).
4. Schüürmann G, Ebert R-U, Chen J, Wang B, Kühne R. 2008. External validation and prediction employing the predictive squared correlation coefficient - test set activity mean vs training set activity mean. *J. Chem. Inf. Model.* **48:** 2140-2145.
5. Tropsha A, Gramatica P, Gombar VK. 2003. The Importance of Being Earnest: Validation is the Absolute Essential for Successful Application and Interpretation of QSPR Models. *QSAR Comb. Sci.* **22:** 69-77.
6. Upton G, Cook I. 2008. Oxford Dictionary of Statistics *Oxford University Press* **ISBN 978-0-19-954145-4** entry for "Variance (data)".

Examples

```
library(cvq2)

data(cvq2.sample.A)
result <- cvq2( cvq2.sample.A, y ~ x1 + x2 )
result

data(cvq2.sample.B)
result <- cvq2( cvq2.sample.B, y ~ x, nFold = 3 )
result

data(cvq2.sample.B)
result <- cvq2( cvq2.sample.B, y ~ x, nFold = 3, nRun = 5 )
result

data(cvq2.sample.A)
data(cvq2.sample.A_pred)
```

```
result <- q2( cvq2.sample.A, cvq2.sample.A_pred, y ~ x1 + x2 )
result

data(cvq2.sample.C)
result <- calibPow( cvq2.sample.C )
result

data(cvq2.sample.D)
result <- predPow( cvq2.sample.D, obs_mean="observed_mean" )
result
```

`cvq2-class` *Class* `"cvq2"`

Description

The class `"cvq2"` extends class `"q2"` and is used to store information about the model calibration, its prediction performance and the cross-validation applied to determine the prediction performance.

Objects from the Class

Objects can be created by calls of the form `new("cvq2", ...)`.

Slots

`result` Contains three lists (`fit`, `pred`, `cv`) regarding the results from linear regression (model calibration, `fit`) and cross-validation (prediction power, `pred` and `cv`) for the given model.

`output` A list of parameters like number formats, output restrictions or output targets.

Linear regression and prediction result list: These lists are inherted from the parent class `q2`. Differences caused by cross-validation appear in the prediction result list for:

`data` For each observation the model parameters used for the prediction are stored additionally as well as the arithmetic mean of the training set

`nTrainingSet` The number of elements in *one* training set ($N - k$) plus an eventually variation.

`nTestSet` The number of elements in *one* test set (k) minus an eventually variation.

Cross-validation result list:

`testSetSizeVaries` *True*, if some test sets consist of $k - 1$ elements.

`nFold` `modelData` is randomly split into *n* equal sized (according to `testSetSizeVaries`) test sets for each individual run.

`nRun` The number of runs each value is predicted.

Extends

Class `"q2"`, directly.

Methods

show Returns a comprehensive overview about the model calibration and the prediction performance.

Author(s)

Torsten Thalheim <torstenthalheim@gmx.de>

Examples

```
showClass("cvq2")
```

`cvq2.sample.A` *Small data set to demonstrate the difference between the conventional and the predictive squared correlation coefficient.*

Description

Contains a small data set with four observations, the observation y depends on two parameters (x_1, x_2).

Usage

```
data(cvq2.sample.A)
```

Format

A data frame with four observations. Each row contains two parameters and the observed value.

`x1` parameter 1

`x2` parameter 2

`y` observation

Details

This data set can be used to demonstrate the differences between the model calibration and the prediction power. The prediction power can be determined either with cross-validation or the application of the model to the data set `cvq2.sample.A_pred`.

Note

This data set contains one outlier (row #2). If the prediction power is determined with cross-validation, this outlier leads to a considerably decreased prediction power, q^2_{cv}, in comparison to the model calibration, r^2. For this data set, one can perform a Leave-One-Out cross-validation only.

Source

Generic data set, created for this purpose only.

See Also

cvq2, q2

---

cvq2.sample.A_pred *Prediction set for model set* cvq2.sample.A.

Description

This data set can be used to determine the prediction power of the model *M* generated with cvq2.sample.A. The four observations *y* depend on two parameters (x_1, x_2).

Usage

```
data(cvq2.sample.A_pred)
```

Format

A data frame with four observations. Each row contains two parameters and the observed value.

x1 parameter 1

x2 parameter 2

y observation

Details

This data set fits very good to the model *M* derived from cvq2.sample.A. The prediction power q^2 of *M* for cvq2.sample.A_pred is as high as its calibration power, r^2.

Source

Generic data set, created for this purpose only.

See Also

cvq2, q2

`cvq2.sample.B` *Small data set to demonstrate the difference between the conventional and the predictive squared correlation coefficient while performing a cross-validation.*

Description

Contains a small data set with six observations, the observed value y depends on the parameter x.

Usage

```
data(cvq2.sample.B)
```

Format

A data frame with six observations and one parameter per observation.

`x` parameter

`y` observation

Details

The sample can be used to demonstrate the various settings of `cvq2`. The cross-validation applied to determine q^2 can be performed either as Leave-One-Out (`nFold` $= N = 6$) or as k-fold (`nFold` $= \{2, 3\}$).
In case `nFold` $= \{2, 3\}$ `modelData` is randomly split into `nFold` disjunct and equal sized test sets. Furthermore one has the opportunity to repeat the cross-validation, while each run (`nRun` $= \{2, 3, \ldots, x\}$) has an individual test set compilation.
The prediction power, q^2_{cv}, calculated for this data set is considerably smaller than the model calibration, r^2, promises.

Source

Generic data set, created for this purpose only.

See Also

`cvq2`, `q2`

cvq2.sample.C *Small data set to demonstrate the statistic methods.*

Description

Contains a small data set with four observations and four predictions.

Usage

```
data(cvq2.sample.C)
```

Format

A data frame with four observations and four predictions.

`observed` observation

`predicted` prediction

Source

Generic data set, created for this purpose only.

See Also

`cvq2`, `predPow`

cvq2.sample.D *Small data set to demonstrate the statistic methods.*

Description

Contains a small data set with four observations, four predictions and the arithmetic mean of the observed values used for each prediction.

Usage

```
data(cvq2.sample.D)
```

Format

A data frame with four observations, four predictions and four different arithmetic means $y_{mean}^{N-k,i}$ (see cvq2-package).

`observed` observation

`predicted` prediction

`observed_mean` mean of the observed values used during the prediction

Source

Generic data set, created for this purpose only.

See Also

cvq2, predPow

---

predPow *Statistical analysis of a model results compared to observations.*

Description

Determines the model calibration or its prediction power. The statistical analysis is done with the observed values and their related prediction only, as no data about the model used to calculate the prediction is available.

Usage

```
calibPow(data, obs = "observed", pred = "predicted",
nu = 0, round = 4, extOut = FALSE, extOutFile = NULL)
predPow(data, obs = "observed", pred = "predicted",
obs_mean = NULL, nu = 0, round = 4, extOut = FALSE,
extOutFile = NULL)
```

Arguments

data	A data frame that contains at least two columns containing the observations and their predictions. The data frame can be extended e.g. by a column that specifies the individual mean of the observed values $y_{mean}^{N-k,i}$.
obs	The name of the column that contains the observations
pred	The name of the column that contains the predictions
obs_mean	The mean of the observations $y_{mean}^{N-k,i}$. Can be either a string that names the actual column or the column itself
nu	The degrees of freedom used for *rmse* calculation, **DEFAULT: 0**
round	The rounding value used in the output, **DEFAULT: 4**
extOut	Extended output, **DEFAULT: FALSE** If extOutFile is not specified, write to stdout()
extOutFile	Write extended output into file (*implies* extOut = TRUE), **DEFAULT: NULL**

Details

`data` contains the observation and the its predictions calculated with model M.

`calibPow()`:
Alias: `calibrationPower()`
The calibration power of model M is calculated with `data`.

`predPow()`:
Alias: `predictionPower()`
The prediction power of model M is calculated with `data`.

Value

Returns an object of class "q2". It contains information about the model calibration or its prediction performance.

Author(s)

Torsten Thalheim <torstenthalheim@gmx.de>

See Also

cvq2

Examples

```
require(methods)
require(stats)
library(cvq2)

data(cvq2.sample.C)
result <- calibPow( cvq2.sample.C )
result

data(cvq2.sample.D)
result <- predPow( cvq2.sample.D, obs_mean="observed_mean" )
result
```

---

q2 *Model prediction power calculation.*

Description

Determines the prediction power of model M. Therefore M is applied to an external data set and its observations are compared to the model predictions. If an external data set is not available, the prediction power is calculated while performing a cross-validation to the model data set.

Usage

```
looq2( modelData, formula = NULL, nu = 1, round = 4,
extOut = FALSE, extOutFile = NULL )

cvq2( modelData, formula = NULL, nFold = N, nRun = 1, nu = 1,
round = 4, extOut = FALSE, extOutFile = NULL )

q2( modelData, predictData, formula = NULL, nu = 0, round = 4,
extOut = FALSE, extOutFile = NULL )
```

Arguments

`modelData`	The model data set consists of parameters $x_1, x_2, \ldots, x_n$ and an observation y
`predictData`	The prediction data set consists of parameters $x_1, x_2, \ldots, x_n$ and an observation y
`formula`	The formula used to predict the observation: $y \sim x_1 + x_2 + \ldots + x_n$ **DEFAULT: NULL** If NULL, a generic formula is derived from the data set, assuming that the last column contains observations whereas the others contain parameters $x_1, x_2, \ldots, x_n$
`nFold`	The data set `modelData` is randomly partitioned into *nFold* equal sized subsets (test sets) during each run, **DEFAULT: *N***, $2 <= nFold <= N$
`nRun`	Number of iterations, the cross-validation is repeated for this data set. This corresponds to the number of individual predictions per observation, $1 <= nRun$, **DEFAULT: 1** Must be 1, if $nFold = N$.
`nu`	The degrees of freedom used in *rmse* calculation in relation to the prediction power, **DEFAULT: 1** (`looq2()`, `cvq2()`), **0** (else)
`round`	The rounding value used in the output, **DEFAULT: 4**
`extOut`	Extended output, **DEFAULT: FALSE** If `extOutFile` is not specified, write to stdout()
`extOutFile`	Write extended output into file (*implies* `extOut = TRUE`), **DEFAULT: NULL**

Details

The calibration of model *M* with `modelData` is done with a linear regression.

`q2()`:
Alias: `qsq()`, `qsquare()`
The model described by `modelData` is used to predict the observations of `predictData`.
These predictions are used to calculate the predictive squared correlation coefficient, q^2.

`cvq2()`:
Alias: `cvqsq()`, `cvqsquare()`
A cross-validation is performed for `modelData`, whereas `modelData` (*N* elements) is split into `nFold` disjunct and equal sized test sets. Each test set consists of *k* elements:

$$k = \left\lceil \frac{N}{nFold} \right\rceil$$

In case $\frac{N}{nFold}$ is a decimal number, some test sets consist of $k-1$ elements. The remaining $N-k$ elements are merged together as training set for this test set and describe the model *M'*. This model is used to predict the observations in the test set. Note, that *M'* is slighlty different to model *M*, which is a result of the missing *k* values.
Each observation from `modelData` is predicted once. The difference between the prediction and the observation within the test sets is used to calculate the **PRE**dictive residual **S**um of **S**quares (*PRESS*). Furthermore for any training set the mean of the observed values in this training set, $y_{mean}^{N-k,i}$, is calculated. *PRESS* and $y_{mean}^{N-k,i}$ are required to calculate the predictive squared correlation coefficient, q_{cv}^2.
In case $k > 1$ one can repeat the cross-validation to overcome biasing. Therefore in each iteration ($\texttt{nRun} = \{1, 2, \ldots, x\}$) the test sets are compiled individually by random. Within one iteration, each observation is predicted once. If $\texttt{nFold} = N$, *one* iteration is necessary only.

`looq2()`:
Same procedure as `cvq2()` (see above), but implicit $\texttt{nFold} = N$ to perform a Leave-One-Out cross-validation. For Leave-One-Out cross-validation *one* iteration (`nRun = 1`) is necessary only.

Value

`q2()`:
The method `q2` returns an object of class `"q2"`. It contains information about the model calibration and its prediction performance on the external data set, `predictData`.

`cvq2(), looq2()`:
The methods `cvq2` and `looq2` return an object of class `"cvq2"`. It contains information about the model calibration and its prediction performance as well as data about the cross-validation applied to `modelData`.

Author(s)

Torsten Thalheim <torstenthalheim@gmx.de>

Examples

```
require(methods)
require(stats)
library(cvq2)

data(cvq2.sample.A)
result <- cvq2( cvq2.sample.A )
result

data(cvq2.sample.B)
result <- cvq2( cvq2.sample.B, y ~ x, nFold = 3 )
result

data(cvq2.sample.B)
result <- cvq2( cvq2.sample.B, y ~ x, nFold = 3, nRun = 5 )
result

data(cvq2.sample.A)
```

```
result <- looq2( cvq2.sample.A, y ~ x1 + x2 )
result

data(cvq2.sample.A)
data(cvq2.sample.A_pred)
result <- q2( cvq2.sample.A, cvq2.sample.A, y ~ x1 + x2 )
result
```

`q2-class` *Class* `"q2"`

Description

The class `"q2"` is used to store information about the calibration of model *M* and its prediction performance. To determine the prediction power, *M* is applied to an external, independent data set.

Objects from the Class

Objects can be created by calls of the form `new("q2", ...)`.

Slots

`result` Contains two lists (`fit`, `pred`) regarding the results from linear regression (model calibration, `fit`) and the application of the model to a validation set (prediction power, `pred`)

`output` A list of parameters like number formats, output restrictions or output targets

Model calibration: This part contains the measurements regarding the model calibration of the linear model *M*.

`data` The observations and the linear fitted predictions by model *M*

`data.col` The explanation of data's column names

`model` The linear model *M*

`n` The number of elements in the data set

`observed_mean` The arithmetic mean of the observations

`r2` The conventional squared correlation coefficient

`rmse` The root mean square error with regard to the degree's of freedom ν

`nu` The degree's of freedom

Prediction performance: This part contains the measurements regarding the prediction power of model *M* which ia applied to an external data set.

`data` Contains the observations and their predictions by *M*

`data.col` The explanation of data's column names

`nTrainingSet` The number of elements in the model set $(N - k)$

`nTestSet` The number of elements in the prediction set (k)

`q2` The predictive squared correlation coefficient

`rmse` The root mean square with regard to the degree's of freedom ν

`nu` The degree's of freedom

Methods

show Returns a comprehensive overview about the model calibration and the prediction performance.

Author(s)

Torsten Thalheim <torstenthalheim@gmx.de>

Examples

```
showClass("q2")
```

Index