

Package ‘SCGLR’

January 27, 2015

Type Package

Title Supervised Component Generalized Linear Regression (SCGLR)

Version 2.0.1

Date 2014-02-21

Author Mortier F., Trottier C., Cornu G., Bry X.

Maintainer “G. Cornu” <gcornu@cirad.fr>

Contact “F. Mortier” <fmortier@cirad.fr> and “G. Cornu”
<gcornu@cirad.fr> and “C. Trottier”
<catherine.trottier@univ-montp3.fr> and “X. Bry”
<xavier.bry@univ-montp2.fr>

Description SCGLR extends the Fisher Scoring Algorithm so as to combine PLS regression with GLM estimation in the multivariate context.

License CeCILL-2 | GPL-2

Depends R (>= 3.0.0)

Imports Matrix,Formula,expm,graphics,ggplot2,grid,ROCR

Suggests parallel,gridExtra,knitr

LazyData yes

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2014-10-14 16:30:27

R topics documented:

scglr-package	2
barplot.SCGLR	3
critConvergence	3
customize	4
genus	5

infoCriterion	6
Methods	7
multivariateFormula	7
multivariateGlm.fit	8
multivariatePredictGlm	10
pairs.SCGLR	11
plot.SCGLR	12
print.SCGLR	13
scglr	14
scglrCrossVal	16
summary.SCGLR	18
Index	19

 scglr-package

Supervised Component Generalized Linear Regression (SCGLR)

Description

SCGLR implements a new PLS regression approach in the multivariate generalized linear framework. The method allows the joint modeling of random variables from different exponential family distributions, searching for common PLS-type components. `scglr` and `scglrCrossVal` are the two main functions. The former constructs the components and performs the parameter estimation, while the latter selects the appropriate number of components by cross-validation. Dedicated plots, print, and summary functions are available. The package contains also an ecological dataset dealing with the abundance of multiple tree genera given a large number of geo-referenced environmental variables.

Author(s)

Mortier F., Trottier C., Cornu G., Bry X.

References

Bry X., Trottier C., Verron T. and Mortier F. (2013) Supervised Component Generalized Linear Regression using a PLS-extension of the Fisher scoring algorithm. *Journal of Multivariate Analysis*, 119, 47-60.#'

barplot.SCGLR	<i>Barplot of percent of overall X variance captured by component</i>
---------------	---

Description

Barplot of percent of overall X variance captured by component

Usage

```
## S3 method for class 'SCGLR'  
barplot(height, ...)
```

Arguments

height	object of class 'SCGLR', usually a result of running scglr .
...	optional arguments.

Value

an object of class ggplot.

See Also

For barplot application see examples in [plot.SCGLR](#).

critConvergence	<i>Auxiliary function for controlling SCGLR fitting</i>
-----------------	---

Description

Auxiliary function for scglr fitting used to construct a convergence control argument.

Usage

```
critConvergence(tol = 1e-06, maxit = 50)
```

Arguments

tol	positive convergence threshold.
maxit	integer, maximum number of iterations.

Value

a list containing elements named as the arguments.

Description

Parameters used to choose what to plot and how. These parameters are given to `plot.SCGLR` and `pairs.SCGLR`.

Details

Parameter name can be abbreviated (e.g. `pred.col` will be understood as `predictors.color`).

Options can be set globally using `options("plot.SCGLR")`. It will then provide default values that can be further overridden by giving explicit parameter value.

<i>parameter name</i>	<i>type (default value). Description.</i>
<code>title</code>	string (NULL). Main title of plot (override built-in).
<code>labels.auto</code>	logical (TRUE). Should covariate or predictor labels be aligned with arrows.
<code>labels.offset</code>	numeric (0.01). Offset by which labels should be moved from tip of arrows.
<code>labels.size</code>	numeric (1). Relative size for labels. Use it to globally alter label size.
<code>expand</code>	numeric (1). Expand factor for windows size. Use it for example to make room for clipped labels.
<code>threshold</code>	numeric. All covariates and/or predictors whose sum of square correlations with the two components exceeds this value.
<code>observations</code>	logical (FALSE). Should we draw observations.
<code>observations.size</code>	numeric (1). Point size.
<code>observations.color</code>	character ("black"). Point color.
<code>observations.alpha</code>	numeric (1). Point transparency.
<code>observations.factor</code>	logical (FALSE). Paint observations according to factor (specify factor).
<code>predictors</code>	logical or array of characters (FALSE). Should we draw predictors and optionally which one (TRUE) (string "red"). Base color used to draw predictors.
<code>predictors.color</code>	string ("red"). Base color used to draw predictors.
<code>predictors.alpha</code>	numeric (1). Overall transparency for predictors (0 is transparent, 1 is opaque).
<code>predictors.arrows</code>	logical (TRUE). Should we draw arrows for predictors.
<code>predictors.arrows.color</code>	string (predictors.color). Specific color for predictor arrows.
<code>predictors.arrows.alpha</code>	numeric (predictors.alpha). Transparency for predictor arrows.
<code>predictors.labels</code>	logical (TRUE). Should we draw labels for predictors.
<code>predictors.labels.color</code>	string (predictors.color). Specific color for predictor labels.
<code>predictors.labels.alpha</code>	numeric (predictors.alpha). Transparency for predictor labels.
<code>predictors.labels.size</code>	numeric (labels.size). Specific size for predictor labels.
<code>predictors.labels.auto</code>	logical (labels.auto). Should predictor labels be aligned with arrows.
<code>covariates</code>	logical or array of characters (TRUE). Should we draw covariates and optionally which one (TRUE) (string "black"). Base color used to draw covariates.
<code>covariates.color</code>	string ("black"). Base color used to draw covariates.
<code>covariates.alpha</code>	numeric (1). Overall transparency for covariates (0 is transparent, 1 is opaque).
<code>covariates.arrows</code>	logical (TRUE). Should we draw arrows for covariates.
<code>covariates.arrows.color</code>	string (covariates.color). Specific color for covariate arrows.
<code>covariates.arrows.alpha</code>	numeric (covariates.alpha). Transparency for covariate arrows.
<code>covariates.labels</code>	logical (TRUE). Should we draw labels for predictors.
<code>covariates.labels.color</code>	string (covariates.color). Specific color for predictor labels.
<code>covariates.labels.alpha</code>	numeric (covariates.alpha). Transparency for covariate labels.
<code>covariates.labels.size</code>	numeric (labels.size). Specific size for covariate labels.

covariates.labels.auto	logical (labels.auto). Should covariate labels be aligned with arrows.
factor	logical or character (FALSE). Should we draw a factor chosen among additional variables (TRUE).
factor.points	logical (TRUE). Should symbol be drawn for factors.
factor.points.size	numeric (4). Symbol size.
factor.points.shape	numeric (13). Point shape.
factor.labels	logical (TRUE). Should factor labels be drawn.
factor.labels.color	string ("black"). Color used to draw labels.
factor.labels.size	numeric (labels.size). Specific size for factor labels.

Examples

```
## Not run:
# setting parameters
plot(genus.scglr)
plot(genus.scglr, predictors=TRUE)
plot(genus.scglr, predictors=TRUE, pred.arrows=FALSE)

# setting global style
options(plot.SCGLR=list(predictors=TRUE, pred.arrows=FALSE))
plot(genus.scglr)

# setting custom style
myStyle <- list(predictors=TRUE, pred.arrows=FALSE)
plot(genus.scglr, style=myStyle)

## End(Not run)
```

genus

Sample dataset of abundance of genera in tropical moist forest

Description

Genus gives the abundance of 27 common tree genera in the tropical moist forest of the Congo-Basin and 40 geo-referenced environmental variables on one thousand 8 by 8 km plots (observations). Each plot's data was obtained by aggregating the data measured on a variable number of previously sampled 0.5 ha sub-plots. Geo-referenced environmental variables were used to describe the physical factors as well as vegetation characteristics. 14 physical factors were used pertaining the description of topography, geology and rainfall of each plot. Vegetation is characterized through 16-days enhanced vegetation index (EVI) data.

Format

gen1 to gen27	abundance of the 27 common genera.
altitude	above-sea level in meters.
pluvio_yr	mean annual rainfall.
forest	classified into seven classes.
pluvio_1 to pluvio_12	monthly rainfalls.

geology	5-level geological substrate.
evi_1 to evi_23	16-days enhanced vegetation indexes.
lon and lat	position of the plot centers.
surface	sampled area.

Note

The use of this dataset for publication must make reference to the CoForChange project.

Author(s)

CoForChange project

References

- S. Gourlet-Fleury et al. (2009–2014) CoForChange project: <http://www.coforchange.eu/>
 C. Garcia et al. (2013–2015) CoForTips project: <http://www.fordev.ethz.ch/research/active/CoForTips>

infoCriterion

Function that calculates cross-validation selection criteria

Description

Function that calculates cross-validation selection criteria

Usage

```
infoCriterion(ynew, pred, family, type, size = NULL, npar = 0)
```

Arguments

ynew	data matrix corresponding to the observations used as test sample.
pred	predicted value of the linear predictor obtained from Xnew and the estimated parameters.
family	a vector of the same length as the number of responses containing characters identifying the distribution families of the dependent variables. "bernoulli", "binomial", "poisson" or "gaussian" are allowed.
type	information criterion used. Likelihood, aic, bic, aicc or Mean Square Prediction Error (mspe) are defined. Area Under ROC Curve (auc) also defined for Bernoulli cases only.
size	describes the number of trials for the binomial dependent variables. A (number of statistical units * number of binomial dependent variables) matrix is expected.
npar	number of parameters used for penalisation.

Value

a matrix containing the criterion value for each dependent variable (row) and each number of components (column).

Methods	<i>Regularization criterion types</i>
---------	---------------------------------------

Description

- LPLS for PLS-type SCGLR
- SR Method iterative normed gradient (ING) for Structural Relevance

Usage

```
methodLPLS()
```

```
methodSR(phi = "vpi", l = 1, s = 1/2, maxiter = 1000, epsilon = 1e-06,
  bailout = 10)
```

Arguments

phi	character string describing structural relevance used in the regularization process. Allowed values are "vpi" for Variable Powered Inertia and "cv" for Component Variance. Default to "vpi".
l	is a numeric argument (>1) tuning the importance of variable bundle locality.
s	is a numeric argument (in [0,1]) tuning the strength of structural relevance with respect to goodness of fit.
maxiter	integer for maximum number of iterations of SR function
epsilon	positive convergence threshold
bailout	integer argument

multivariateFormula	<i>Formula construction</i>
---------------------	-----------------------------

Description

Helper function for building multivariate scglr formula.

NOTE: Interactions involving factors are not allowed for now. For interactions between two quantitative variables, use I(x*y) as usual.

Usage

```
multivariateFormula(namesY, namesX, namesAX = NULL)
```

Arguments

namesY	a vector of character containing the names of the dependent variables.
namesX	a vector of character containing the names of the covariates (X) involved in the components.
namesAX	a vector of character containing the names of the additional covariates.

Value

an object of class Formula.

multivariateGlm.fit *Multivariate generalized linear regression*

Description

multivariateGlm is used to fit multivariate generalized linear models specified by a symbolic formula together with the distributions of the responses. This function performs a simple GLM fit for each dependent variable with the associated distribution.

Usage

```
multivariateGlm.fit(Y, comp, family, offset, size)

multivariateGlm(formula, data, family, size = NULL, offset = NULL,
  subset = NULL)
```

Arguments

Y	matrix of dependent variables.
comp	matrix of covariates.
family	a vector of character giving the family distribution of each response.
offset	used for the poisson dependent variables. A vector or a matrix of size: number of observations * number of Poisson dependent variables is expected.
size	a matrix giving the number of trials for each Binomial dependent variable ncol(size) must be equal to the number of Binomial variables.
formula	an object of class Formula (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	the data frame to be modeled.
subset	an optional vector specifying a subset of observations to be used in the fitting process.

Value

the list, each item of which is the glm object associated with each response.

Examples

```

## Not run:
library(SCGLR)

# load sample data
data(genus)

# get variable names from dataset
n <- names(genus)
ny <- n[grep("^gen",n)] # Y <- names that begins with "gen"
nx <- n[-grep("^gen",n)] # X <- remaining names

# remove "geology" and "surface" from nx as surface
# is offset and we want to use geology as additional covariate
nx <- nx[!nx%in%c("geology","surface")]

# build multivariate formula
# we also add "lat*lon" as computed covariate
form <- multivariateFormula(ny,c(nx,"I(lat*lon)"),c("geology"))

# split genus dataset
sub <- sample(1:nrow(genus),100,replace=F)
sub_fit <- (1:nrow(genus))[-sub]

# define family
fam <- rep("poisson",length(ny))

# fit the model
genus.scglr <- scglr(formula=form, data=genus, family=fam, K=4,
  offset=genus$surface, subset=sub_fit)

# xnew, the design matrix associated to sub-sample used for prediction
# note rhs parameter is introduced to take into account that the
# covariate part of the formula is composed of two different sets
xnew <- model.matrix(form, data=genus[sub,], rhs=1:2)[,-1]

# prediction based on the scglr approach
pred.scglr <- multivariatePredictGlm(xnew,family=fam,
  beta=genus.scglr$beta, offset=genus$surface[sub])
cor.scglr <-diag(cor(pred.scglr,genus[sub,ny]))
plot(cor.scglr, col="red",ylim=c(-1,1))

# prediction based on classical poisson glm
genus.glm <- multivariateGlm(formula=form, data=genus, family=fam,
  offset=genus$surface, subset=sub_fit)
coefs <- sapply(genus.glm,coef)

# rhs parameter is introduced to take into account that the
# covariate part of the formula is composed of two different sets
pred.glm <- multivariatePredictGlm(xnew,family=fam,beta=coefs,
  offset=genus$surface[sub])
cor.glm <- diag(cor(pred.glm,genus[sub,ny]))

```

```
points(cor.glm, col="blue")

## End(Not run)
```

```
multivariatePredictGlm
```

Function that predicts the responses from the covariates for a new sample

Description

Function that predicts the responses from the covariates for a new sample

Usage

```
multivariatePredictGlm(Xnew, family, beta, offset = NULL)
```

Arguments

Xnew	a data frame containing the values of the covariates for the new sample.
family	a vector of character specifying the distributions of the responses.
beta	the matrix of coefficients estimated from the calibration sample.
offset	used for the poisson dependent variables. A vector or a matrix of size: number of observations * number of Poisson dependent variables is expected.

Value

a matrix of predicted values.

Examples

```
## Not run:
library(SCGLR)

# load sample data
data(genus)

# get variable names from dataset
n <- names(genus)
ny <- n[grepl("^gen",n)] # Y <- names that begins with "gen"
nx <- n[-grepl("^gen",n)] # X <- remaining names

# remove "geology" and "surface" from nx as surface is
# offset and we want to use geology as additional covariate
nx <- nx[!nx%in%c("geology","surface")]

# build multivariate formula
```

```

# we also add "lat*lon" as computed covariate
form <- multivariateFormula(ny,c(nx,"I(lat*lon)"),c("geology"))

# split genus dataset
sub <- sample(1:nrow(genus),100,replace=F)
sub_fit <- (1:nrow(genus))[-sub]

# define family
fam <- rep("poisson",length(ny))

# fit the model
genus.scglr <- scglr(formula=form, data=genus, family=fam, K=4,
  offset=genus$surface, subset=sub_fit)

# xnew, the design matrix associated to sub-sample used for prediction
# rhs parameters is introduced to take into account that the covariate
# of the formula is composed of two differents sets
xnew <- model.matrix(form, data=genus[sub,], rhs=1:2)[,-1]

# prediction based on the scglr approch
pred.scglr <- multivariatePredictGlm(xnew,family=fam,
  beta=genus.scglr$beta,offset=genus$surface[sub])
cor.scglr <-diag(cor(pred.scglr,genus[sub,ny]))
plot(cor.scglr, col="red",ylim=c(-1,1))

# prediction based on classical poisson glm
genus.glm <- multivariateGlm(formula=form, data=genus, family=fam,
  offset=genus$surface, subset=sub_fit)
coefs <- sapply(genus.glm,coef)

# rhs parameters is introduced to take into account that the covariate
# part of the formula is composed of two differents sets
pred.glm <- multivariatePredictGlm(xnew,family=fam,beta=coefs,
  offset=genus$surface[sub])
cor.glm <- diag(cor(pred.glm,genus[sub,ny]))

points(cor.glm, col="blue")

## End(Not run)

```

pairs.SCGLR

Pairwise scglr plot on components

Description

Pairwise scglr plot on components

Usage

```

## S3 method for class 'SCGLR'
pairs(x, ..., nrow = NULL, ncol = NULL, components = NULL)

```

Arguments

x	object of class 'SCGLR', usually a result of running <code>scglr</code> .
nrow	number of rows of the grid layout.
ncol	number of columns of the grid layout.
components	vector of integers selecting components to plot (default is all components).
...	optionally, further arguments forwarded to <code>link{plot.SCGLR}</code> .

See Also

For pairs application see examples in [plot.SCGLR](#)

plot.SCGLR

SCGLR generic plot

Description

SCGLR generic plot

Usage

```
## S3 method for class 'SCGLR'
plot(x, ..., style = getOption("plot.SCGLR"), plane = c(1,
  2))
```

Arguments

x	an object from SCGLR class.
style	named list of values used to customize the plot (see customize)
plane	a size-2 vector (or comma separated string) indicating which components are plotted (eg: <code>c(1,2)</code> or <code>"1,2"</code>).
...	optional arguments (see customize).

Value

an object of class `ggplot`.

Examples

```
## Not run:
library(SCGLR)

# load sample data
data(genus)

# get variable names from dataset
n <- names(genus)
```

```

ny <- n[grep("^gen",n)] # Y <- names that begins with "gen"
nx <- n[-grep("^gen",n)] # X <- remaining names

# remove "geology" and "surface" from nx
# as surface is offset and we want to use geology as additional covariate
nx <- nx[!nx%in%c("geology","surface")]

# build multivariate formula
# we also add "lat*lon" as computed covariate
form <- multivariateFormula(ny,c(nx,"I(lat*lon)"),c("geology"))

# define family
fam <- rep("poisson",length(ny))

genus.scglr <- scglr(formula=form,data = genus,family=fam, K=4,
  offset=genus$surface)

summary(genus.scglr)

barplot(genus.scglr)

plot(genus.scglr)

plot(genus.scglr, predictors=TRUE, factor=TRUE)

pairs(genus.scglr)

## End(Not run)

```

print.SCGLR

Print SCGLR object

Description

Prints inertia per component and deviance for each Y.

Usage

```

## S3 method for class 'SCGLR'
print(x, ...)

```

Arguments

x	object of class 'SCGLR', usually a result of running <code>scglr</code> .
...	Not used.

 scglr

Function that fits the scglr model

Description

Calculates the components to predict all the dependent variables.

Usage

```
scglr(formula, data, family, K = 1, size = NULL, offset = NULL,
      subset = NULL, na.action = na.omit, crit = list(),
      method = methodSR())
```

Arguments

formula	an object of class Formula (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	a data frame to be modeled.
family	a vector of character of the same length as the number of dependent variables: "bernoulli", "binomial", "poisson" or "gaussian" is allowed.
K	number of components, default is one.
size	describes the number of trials for the binomial dependent variables. A (number of statistical units * number of binomial dependent variables) matrix is expected.
offset	used for the poisson dependent variables. A vector or a matrix of size: number of observations * number of Poisson dependent variables is expected.
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 to na.omit.
crit	a list of two elements : maxit and tol, describing respectively the maximum number of iterations and the tolerance convergence criterion for the Fisher scoring algorithm. Default is set to 50 and 10e-6 respectively.
method	Regularization criterion type. Object of class "method.SCGLR" built by methodPLS for PLS-type approach or methodSR for Structural Relevance.

Value

an object of the SCGLR class.

The function [summary](#) (i.e., [summary.SCGLR](#)) can be used to obtain or print a summary of the results.

The generic accessor functions [coef](#) can be used to extract various useful features of the value returned by [scglr](#).

An object of class "SCGLR" is a list containing following components:

u	matrix of size (number of regressors * number of components), contains the component-loadings, i.e. the coefficients of the regressors in the linear combination giving each component.
comp	matrix of size (number of statistical units * number of components) having the components as column vectors.
compr	matrix of size (number of statistical units * number of components) having the standardized components as column vectors.
gamma	list of length number of dependant variables. Each element is a matrix of coefficients, standard errors, z-values and p-values.
beta	matrix of size (number of regressors + 1 (intercept) * number of dependent variables), contains the coefficients of the regression on the original regressors X.
lin.pred	data.frame of size (number of statistical units * number of dependent variables), the fitted linear predictor.
xFactors	data.frame containing the nominal regressors.
xNumeric	data.frame containing the quantitative regressors.
inertia	matrix of size (number of components * 2), contains the percentage and cumulative percentage of the overall regressors' variance, captured by each component.
deviance	vector of length (number of dependent variables), gives the deviance of each y_k 's GLM on the components.

References

Bry X., Trottier C., Verron T. and Mortier F. (2013) Supervised Component Generalized Linear Regression using a PLS-extension of the Fisher scoring algorithm. *Journal of Multivariate Analysis*, 119, 47-60.

Examples

```
## Not run:
library(SCGLR)

# load sample data
data(genus)

# get variable names from dataset
n <- names(genus)
ny <- n[grep("^gen",n)] # Y <- names that begins with "gen"
nx <- n[-grep("^gen",n)] # X <- remaining names

# remove "geology" and "surface" from nx
# as surface is offset and we want to use geology as additional covariate
nx <- nx[!nx%in%c("geology","surface")]

# build multivariate formula
# we also add "lat*lon" as computed covariate
form <- multivariateFormula(ny,c(nx,"I(lat*lon)"),c("geology"))

# define family
```

```
fam <- rep("poisson",length(ny))

genus.scglr <- scglr(formula=form,data = genus,family=fam, K=4,
  offset=genus$surface)

summary(genus.scglr)

## End(Not run)
```

scglrCrossVal	<i>Function that fits and selects the number of component by cross-validation.</i>
---------------	--

Description

Function that fits and selects the number of component by cross-validation.

Usage

```
scglrCrossVal(formula, data, family, K = 1, nfolds = 5, type = "mspe",
  size = NULL, offset = NULL, subset = NULL, na.action = na.omit,
  crit = list(), method = methodSR(), mc.cores = 1)
```

Arguments

formula	an object of class "Formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	the data frame to be modeled.
family	a vector of character of length q specifying the distributions of the responses. Bernoulli, binomial, poisson and gaussian are allowed.
K	number of components, default is one.
nfolds	number of folds, default is 5. Although nfolds can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets.
type	loss function to use for cross-validation. Currently six options are available depending on whether the responses are of the same distribution family. If the responses are all bernoulli distributed, then the prediction performance may be measured through the area under the ROC curve: type = "auc" In any other case one can choose among the following five options ("likelihood", "aic", "aicc", "bic", "mspe").
size	specifies the number of trials of the binomial variables included in the model. A (n*qb) matrix is expected for qb binomial variables.
offset	used for the poisson dependent variables. A vector or a matrix of size: number of observations * number of Poisson dependent variables is expected.
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 to the na.omit.
crit	a list of two elements : maxit and tol, describing respectively the maximum number of iterations and the tolerance convergence criterion for the Fisher scoring algorithm. Default is set to 50 and 10e-6 respectively.
method	Regularization criterion type. Object of class "method.SCGLR" built by methodPLS for PLS-type approach or methodSR for Structural Relevance.
mc.cores	max number of cores to use when using parallelization (Not available in windows yet and strongly discouraged if in interactive mode).

Value

a matrix containing the criterion values for each response (rows) and each number of components (columns).

References

Bry X., Trottier C., Verron T. and Mortier F. (2013) Supervised Component Generalized Linear Regression using a PLS-extension of the Fisher scoring algorithm. *Journal of Multivariate Analysis*, 119, 47-60.

Examples

```
## Not run:
library(SCGLR)

# load sample data
data(genus)

# get variable names from dataset
n <- names(genus)
ny <- n[grep("^gen",n)] # Y <- names that begins with "gen"
nx <- n[-grep("^gen",n)] # X <- remaining names

# remove "geology" and "surface" from nx
# as surface is offset and we want to use geology as additional covariate
nx <- nx[!nx%in%c("geology","surface")]

# build multivariate formula
# we also add "lat*lon" as computed covariate
form <- multivariateFormula(ny,c(nx,"I(lat*lon)"),c("geology"))

# define family
fam <- rep("poisson",length(ny))

# cross validation
genus.cv <- scglrCrossVal(formula=form, data=genus, family=fam, K=12,
  offset=genus$surface)

# find best K
mean.crit <- t(apply(genus.cv,1,function(x) x/mean(x)))
```

```

mean.crit <- apply(mean.crit,2,mean)
K.cv <- which.min(mean.crit)-1

#plot(mean.crit, type="l")

## End(Not run)

```

summary.SCGLR

Summarizing SCGLR fits

Description

Summary method for class "SCGLR".

Usage

```

## S3 method for class 'SCGLR'
summary(object, ...)

## S3 method for class 'summary.SCGLR'
print(x, digits = 3, cutoff = 1, ...)

```

Arguments

object	an object of class "SCGLR", usually a result of a call to <code>scglr</code> .
x	an object of class "summary.SCGLR", usually a result of a call to <code>summary.SCGLR</code> .
digits	the number of significant digits to use when printing.
cutoff	print coefficients with pvalue lower than or equal to cutoff (default to 1).
...	Not used.

Value

an object of class "summary.SCGLR".

inertia	inertia per component.
deviance	deviance for each Y_k .
rho	squared correlations with numerical covariates.
rho.pred	squared correlations with linear predictors.
coefficients	contains the coefficients of the regression on the components.
pvalue	contains the pvalues of the coefficients of the regression on the components.

Index

barplot.SCGLR, 3

coef, 14

critConvergence, 3

customize, 4, 12

genus, 5

ggplot, 12

infoCriterion, 6

methodLPLS, 14, 17

methodLPLS (Methods), 7

Methods, 7

methodSR, 14, 17

methodSR (Methods), 7

multivariateFormula, 7

multivariateGlm (multivariateGlm.fit), 8

multivariateGlm.fit, 8

multivariatePredictGlm, 10

pairs.SCGLR, 4, 11

plot.SCGLR, 3, 4, 12, 12

print.SCGLR, 13

print.summary.SCGLR (summary.SCGLR), 18

scglr, 2, 3, 12, 13, 14, 18

scglr-package, 2

scglrCrossVal, 2, 16

summary, 14

summary.SCGLR, 14, 18