

Package ‘geeVerse’

March 13, 2024

Type Package

Title A Comprehensive Analysis of High Dimensional Longitudinal Data

Version 0.1.0

Description To provide a comprehensive analysis of high dimensional longitudinal data, this package provides analysis for any combination of 1) simultaneous variable selection and estimation, 2) mean regression or quantile regression for heterogeneous data, 3) cross-sectional or longitudinal data, 4) balanced or imbalanced data, 5) moderate, high or even ultra-high dimensional data, via computationally efficient implementations of penalized generalized estimating equations.

License GPL-3

Encoding UTF-8

Imports mvtnorm, quantreg, Rcpp (>= 0.10.2), doParallel, foreach

Suggests methods

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.2.3

NeedsCompilation yes

Author Tianhai Zu [aut, cre],
Brittany Green [aut, ctb],
Yan Yu [aut, ctb]

Maintainer Tianhai Zu <zuti@mail.uc.edu>

Repository CRAN

Date/Publication 2024-03-13 11:10:02 UTC

R topics documented:

CVfit	2
generateData	3
PGEE	4
predict.qpgee	6
qpgee	7
qpgee_tune	8
Siga_cov	10

CVfit

*Cross-Validation for Generalized Estimating Equations (GEE)***Description**

This function performs k-fold cross-validation for model selection in the context of Generalized Estimating Equations (GEE). It is designed to evaluate the performance of different models specified by a range of lambda values, choosing the one that minimizes the cross-validation criterion.

Usage

```
CVfit(
  formula,
  id,
  data,
  family,
  scale.fix,
  scale.value,
  fold,
  lambda.vec,
  pindex,
  eps,
  maxiter,
  tol
)
```

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.
id	a vector which identifies the cluster/group for each observation.
data	an optional data frame containing the variables in the model.
family	a description of the error distribution and link function to be used in the model.
scale.fix	logical; if TRUE, the scale parameter is fixed to scale.value.
scale.value	the value of the scale parameter when scale.fix is TRUE.
fold	the number of folds to be used in the cross-validation.
lambda.vec	a vector of lambda values for which the cross-validation error will be calculated.
pindex	an optional numeric vector specifying a parameter index.
eps	the threshold for convergence criteria.
maxiter	the maximum number of iterations for the convergence of the algorithm.
tol	the tolerance level for the convergence of the algorithm.

Value

An object of class "CVfit", which is a list containing:

fold The number of folds used in the cross-validation.

lam.vect The vector of lambda values tested.

cv.vect The cross-validation error for each lambda.

lam.opt The lambda value that resulted in the minimum cross-validation error.

cv.min The minimum cross-validation error.

call The matched call.

generateData	<i>Generate Data for Simulation</i>
--------------	-------------------------------------

Description

This function generates simulated data including the predictor matrix 'X' and the response vector 'y', based on the specified parameters. The function allows for the simulation of data under different settings of correlation, distribution, and the number of observations and subjects.

Usage

```
generateData(n_sub, n_obs, p, beta0, rho, type = "ar", dis = "normal", ka)
```

Arguments

n_sub Integer, the number of subjects.

n_obs Integer or numeric vector, the number of observations per subject.

p Integer, the number of predictors.

beta0 Numeric vector, initial coefficients for the first few predictors.

rho Numeric, the correlation coefficient used in generating correlated errors.

type Character, the type of correlation structure (default is autoregressive).

dis Character, the distribution of errors ("normal" or "t").

ka 1 for heterogeneous errors and 0 for homogeneous errors.

Value

A list containing two elements: 'X', the matrix of predictors, and 'y', the response vector.

Examples

```
sim_data <- generateData(n_sub = 100, n_obs = rep(10, 100), p = 200,
                        beta0 = rep(1,7), rho = 0.6, type = "ar",
                        dis = "normal", ka = 1)
```

 PGEE

PGEE accelerated with RCpp

Description

A function to fit penalized generalized estimating equation model. This function was re-wrote partly with RCPP and RCPPEigen for better computation efficiency.

Usage

```
PGEE(
  formula,
  id,
  data,
  na.action = NULL,
  family = gaussian(link = "identity"),
  corstr = "independence",
  Mv = NULL,
  beta_int = NULL,
  R = NULL,
  scale.fix = TRUE,
  scale.value = 1,
  lambda,
  pindex = NULL,
  eps = 10^-6,
  maxiter = 30,
  tol = 10^-3,
  silent = TRUE
)
```

Arguments

formula	A formula expression response ~ predictors;
id	A vector for identifying subjects/clusters.
data	A data frame which stores the variables in formula with id variable.
na.action	A function to remove missing values from the data. Only na.omit is allowed here.
family	A family object: a list of functions and expressions for defining link and variance functions. Families supported in PGEE are binomial, gaussian, gamma and poisson. The links, which are not available in gee, is not available here. The default family is gaussian.
corstr	A character string, which specifies the type of correlation structure. Structures supported in PGEE are "AR-1", "exchangeable", "fixed", "independence", "stat_M_dep", "non_stat_M_dep", and "unstructured". The default corstr type is "independence".

Mv	If either "stat_M_dep", or "non_stat_M_dep" is specified in corstr, then this assigns a numeric value for Mv. Otherwise, the default value is NULL.
beta_int	User specified initial values for regression parameters. The default value is NULL.
R	If corstr = "fixed" is specified, then R is a square matrix of dimension maximum cluster size containing the user specified correlation. Otherwise, the default value is NULL.
scale.fix	A logical variable; if true, the scale parameter is fixed at the value of scale.value. The default value is TRUE.
scale.value	If scale.fix = TRUE, this assigns a numeric value to which the scale parameter should be fixed. The default value is 1.
lambda	A numerical value for the penalization parameter of the scad function, which is estimated via cross-validation.
pindex	An index vector showing the parameters which are not subject to penalization. The default value is NULL. However, in case of a model with intercept, the intercept parameter should be never penalized.
eps	A numerical value for the epsilon used in minorization-maximization algorithm. The default value is 10^{-6} .
maxiter	The number of iterations that is used in the estimation algorithm. The default value is 25.
tol	The tolerance level that is used in the estimation algorithm. The default value is 10^{-3} .
silent	A logical variable; if false, the regression parameter estimates at each iteration are printed. The default value is TRUE.

Value

a PGEE object, which includes: fitted coefficients - the fitted single index coefficients with unit norm and first component being non negative

Examples

```
#generate data
set.seed(2021)
sim_data <- generateData(n_sub = 100, n_obs = rep(10, 100), p = 100,
                        beta0 = rep(1,7), rho = 0.6, type = "ar",
                        dis = "normal", ka = 1)

X=sim_data$X
y=sim_data$y
id = rep(1:100, each = 10)
data = data.frame(X,y,id)

PGEE_fit = PGEE("y ~.-id-1",id = id, data = data,corstr = "exchangeable",lambda=0.01)
PGEE_fit$coefficients
```

`predict.qpgee`*Predict method for qpgee model objects*

Description

This function makes predictions from a "qpgee" model object. When 'newdata' is not provided, it returns predictions using the original data the model was fitted on. If 'newdata' is supplied (through '...'), it uses this new data for prediction.

Usage

```
## S3 method for class 'qpgee'  
predict(object, ...)
```

Arguments

<code>object</code>	A "qpgee" model object.
<code>...</code>	Additional arguments to the function. Can include 'newdata', a dataframe containing the new data to predict on. The structure of 'newdata' should match that of the data the model was originally fitted with, specifically in terms of the variables it contains. Additional arguments are ignored.

Value

If 'newdata' is not supplied, returns a vector of predictions based on the fitted values and handling of NAs specified in the model object. If 'newdata' is supplied, returns a vector of predictions for the new data.

Examples

```
# Example usage:  
sim_data <- generateData(n_sub = 100, n_obs = rep(10, 100), p = 100,  
                        beta0 = rep(1,7), rho = 0.6, type = "ar",  
                        dis = "normal", ka = 1)  
  
X=sim_data$X  
y=sim_data$y  
  
#fit qpgee  
qpgee.fit = qpgee(X,y,tau=0.5,nk=rep(10, 100))  
predict(qpgee.fit)
```

Description

This function implements Quantile Penalized Generalized Estimating Equations (QPGEE) for longitudinal data analysis. It estimates parameters using a penalized quantile regression approach within a GEE framework, allowing for different working correlation structures.

Usage

```
qpgee(
  x,
  y,
  tau = 0.5,
  nk = rep(1, length(y)),
  worktype = "CS",
  lambda = 0.1,
  betaint = NULL,
  f0 = NULL,
  max_it = 100,
  cutoff = 10^-1
)
```

Arguments

x	A matrix of predictors.
y	A numeric vector of response variables.
tau	The quantile to be estimated (default is 0.5, the median).
nk	A numeric vector indicating the number of observations per subject.
worktype	A string specifying the working correlation structure. Options include "CS" (Compound Symmetry), "AR" (Autoregressive), "Tri" (Tri-diagonal), and "Ind" (Independent).
lambda	The penalty parameter for regularization (default is 0.1).
betaint	Initial values for the beta coefficients. If NULL, non-longitudinal quantile regression is used for initialization.
f0	estimated conditional error distributions.
max_it	Maximum number of iterations (default is 100).
cutoff	Threshold for coefficient shrinkage (default is 0.1).

Value

A list containing the following components:

beta	Estimated beta coefficients.
g	Fitted values of the linear predictor.
R	Estimated working correlation matrix.
X_selected	Indices of selected predictors.
mcl	Mean check loss.
hbic	Hannan-Quinn Information Criterion value.
converge	Boolean indicating whether the algorithm converged.

Examples

```
# Example usage:
sim_data <- generateData(n_sub = 100, n_obs = rep(10, 100), p = 100,
                        beta0 = rep(1,7), rho = 0.6, type = "ar",
                        dis = "normal", ka = 1)

X=sim_data$X
y=sim_data$y

#fit qpgee
qpgee.fit = qpgee(X,y,tau=0.5,nk=rep(10, 100))
qpgee.fit$beta
```

qpgee_tune

Quantile Penalized Generalized Estimating Equations with Auto Selected Penalty level

Description

This function automatically select the penalty level by going through a list of lambdas, and select the best level of penalty with high-dimensional BIC (HBIC) or cross-validation (CV).

Usage

```
qpgee_tune(
  x,
  y,
  tau = 0.5,
  method = "HBIC",
  ncore = 1,
  nk = rep(1, length(y)),
  worktype = "CS",
  lambda = NULL,
```



```

  f0 = NULL,
  betaint = NULL,
  max_it = 100,
  cutoff = 10^-1
)

```

Arguments

x	A matrix of predictors.
y	A numeric vector of response variables.
tau	The quantile to be estimated (default is 0.5, the median).
method	The criterion to select level of penalty. Currently it only supports "HBIC".
ncore	A numeric value specifying how many core to use.
nk	A numeric vector indicating the number of observations per subject.
worktype	A string specifying the working correlation structure. Options include "CS" (Compound Symmetry), "AR" (Autoregressive), "Tri" (Tri-diagonal), and "Ind" (Independent).
lambda	A vector of penalty parameter for regularization. If not provided, a grid will be provided by this function.
f0	estimated conditional error distributions.
betaint	Initial values for the beta coefficients. If NULL, non-longitudinal quantile regression is used for initialization.
max_it	Maximum number of iterations (default is 100).
cutoff	Threshold for coefficient shrinkage (default is 0.1).

Value

A list containing the following components:

beta	Estimated beta coefficients.
g	Fitted values of the linear predictor.
R	Estimated working correlation matrix.
X_selected	Indices of selected predictors.
mcl	Mean check loss.
hbic	Hannan-Quinn Information Criterion value.
converge	Boolean indicating whether the algorithm converged.

Examples

```

# Example usage:

sim_data <- generateData(n_sub = 20, n_obs = rep(10, 20), p = 20,
  beta0 = rep(1,5), rho = 0.1, type = "ar",
  dis = "normal", ka = 1)

```

```
X=sim_data$X
y=sim_data$y

#fit qpgee with auto selected lambda
qpgee.fit = qpgee_tune(X,y,tau=0.5,nk=rep(10, 20),ncore=1)
qpgee.fit$beta
```

Siga_cov

Generate Covariance Matrix

Description

This function generates a covariance matrix based on the specified correlation structure. The function supports "compound symmetry" (cs) and "autoregressive" (ar) correlation structures, as well as an identity matrix as the default option when neither "cs" nor "ar" is specified.

Usage

```
Siga_cov(rho, type, nt)
```

Arguments

rho	Numeric, the correlation coefficient used for generating the covariance matrix. For "cs", it represents the common correlation between any two observations. For "ar", it represents the correlation between two consecutive observations, with the correlation decreasing for observations further apart.
type	Character, specifies the type of correlation structure for the covariance matrix. Options are "cs" for compound symmetry, "ar" for autoregressive, and any other input will result in an identity matrix.
nt	Integer, the dimension of the square covariance matrix (number of time points or observations).

Value

A square matrix of dimension 'nt' representing the specified covariance structure.

Index

CVfit, [2](#)

generateData, [3](#)

PGEE, [4](#)

predict.qpgee, [6](#)

qpgee, [7](#)

qpgee_tune, [8](#)

Siga_cov, [10](#)