

Package: geeVerse (via r-universe)

October 10, 2024

Type Package

Title A Comprehensive Analysis of High Dimensional Longitudinal Data

Version 0.2.1

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

LazyData true

Depends R (>= 3.5.0)

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

Suggests methods

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.2.3

Repository <https://zzz1990771.r-universe.dev>

RemoteUrl <https://github.com/zzz1990771/geeverse>

RemoteRef HEAD

RemoteSha 4d5d3650ea96b9a90a95ef084faed7e4c360b140

Contents

CVfit	2
generateData	3
PGEE	4
predict.qpgee	6
qpgee	7

qpgee.est	8
Siga_cov	10
simuGene	10
yeastG1	11

Index	13
--------------	-----------

CVfit	<i>Cross-Validation for Generalized Estimating Equations (GEE)</i>
-------	--

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,
  ncore = 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.
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.
ncore	if greater than 1, the function will use parallel computation.

Details

Note that this is a re-implemented version with parallel computing.

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(nsub, nobs, p, beta0, rho, correlation = "AR1", dis = "normal", ka)
```

Arguments

nsub	Integer, the number of subjects.
nobs	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.
correlation	Character, the correlation 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(nsub = 100, nobs = rep(10, 100), p = 200,
                        beta0 = c(rep(1,7),rep(0,93)), rho = 0.6, correlation = "AR1",
                        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.


```

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

object	A "qpgee" model object.
...	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(nsub = 100, nobs = rep(10, 100), p = 100,
                        beta0 = c(rep(1,7),rep(0,93)), rho = 0.6, correlation = "AR1",
                        dis = "normal", ka = 1)

X=sim_data$X
y=sim_data$y

#fit qpgee
qpgee.fit = qpgee(X,y,tau=0.5,nobs=rep(10, 100))
predict(qpgee.fit)

```

qpgee	<i>Quantile Penalized Generalized Estimating Equations with Auto Selected Penalty level</i>
-------	---

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(
  x,
  y,
  tau = 0.5,
  method = "HBIC",
  ncore = 1,
  nobs = rep(1, length(y)),
  correlation = "exchangeable",
  lambda = NULL,
  intercept = FALSE,
  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.
nobs	A numeric vector indicating the number of observations per subject.
correlation	A string specifying the working correlation structure. Options include "exchangeable" (Exchangeable), "AR1" (Autoregressive), "Tri" (Tri-diagonal), and "exchangeable" (Independent).
lambda	A vector of penalty parameter for regularization. If not provided, a grid will be provided by this function.
intercept	Whether to include an intercept when estimating.
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.
mc1	Mean check loss.
hbic	Hannan-Quinn Information Criterion value.
converge	Boolean indicating whether the algorithm converged.

Examples

```
# Example usage:

sim_data <- generateData(nsub = 20, nobs = rep(10, 20), p = 20,
                        c(rep(1,5),rep(0,15)), rho = 0.1, correlation = "AR1",
                        dis = "normal", ka = 1)

X=sim_data$X
y=sim_data$y

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

qpgee.est	<i>Quantile Penalized Generalized Estimating Equations (QPGE) Estimation Function</i>
-----------	---

Description

This function implements Quantile Penalized Generalized Estimating Equations (QPGE) 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.est(
  x,
  y,
  tau = 0.5,
  nobs = rep(1, length(y)),
  correlation = "exchangeable",
  lambda = 0.1,
```



```

    intercept = FALSE,
    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).
nobs	A numeric vector indicating the number of observations per subject.
correlation	A string specifying the working correlation structure. Options include "exchangeable" (Exchangeable), "AR1" (Autoregressive), "Tri" (Tri-diagonal), and "independence" (Independent).
lambda	The penalty parameter for regularization (default is 0.1).
intercept	Whether to include an intercept when estimating.
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(nsub = 100, nobs = rep(10, 100), p = 100,
                        c(rep(1,7),rep(0,93)), rho = 0.6, correlation = "AR1",
                        dis = "normal", ka = 1)

X=sim_data$X
y=sim_data$y

```

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

Siga_cov	<i>Generate Covariance Matrix</i>
----------	-----------------------------------

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 "AR1" is specified.

Usage

```
Siga_cov(rho, correlation, 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 "AR1", it represents the correlation between two consecutive observations, with the correlation decreasing for observations further apart.
correlation	Character, specifies the correlation of correlation structure for the covariance matrix. Options are "cs" for compound symmetry, "AR1" 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.

simuGene	<i>A Simulated Genetic Data from HapGen2</i>
----------	--

Description

The 'simuGene' dataset contains 500 SNPs simulated data from a commonly used tool for genetic data, HapGen2. We re-sampled existing genotype data to create this simulated data. The genotype data we resample from is the publicly available 1000 Genomes Project data. More specifically, we use resampled from chromosome 14.

Usage

```
simuGene
```

Format

A data frame with 1000 rows (subjects) and 500 columns (SNPs).

Examples

```
data(simuGene)
head(simuGene)
```

yeastG1

A Subset of Yeast Cell Cycle Gene Expression Data (G1 Phase)

Description

The ‘yeastG1’ dataset contains gene expression data from the yeast cell cycle during the G1 phase. The original dataset (Spellman et al. 1998) includes expression levels for 6178 genes measured at 18 time points. And this is a subset of 283 cell-cycled-regularized genes observed over 4 time points at G1 stage and the standardized binding probabilities of a total of 96 TFs obtained from

Usage

```
yeastG1
```

Format

A data frame with 1132 rows and 99 columns.

The dataset contains gene expression levels for the following transcription factors: ABF1, ACE2, ADR1, ARG80, ARG81, ARO80, ASH1, BAS1, CAD1, CBF1, CIN5, CRZ1, CUP9, DAL81, DAL82, DIG1, DOT6, FHL1, FKH1, FKH2, FZF1, GAL4, GAT1, GAT3, GCN4, GCR1, GCR2, GLN3, GRF10.Pho2., GTS1, HAL9, HAP2, HAP3, HAP4, HAP5, HIR1, HIR2, HMS1, HSF1, IME4, INO2, INO4, IXR1, LEU3, MAC1, MAL13, MATa1, MBP1, MCM1, MET31, MET4, MIG1, MOT3, MSN1, MSN4, MSS11, MTH1, NDD1, NRG1, PDR1, PHD1, PHO4, PUT3, RAP1, RCS1, REB1, RFX1, RGM1, RLM1, RME1, ROX1, RPH1, RTG1, RTG3, SFP1, SIG1, SIP4, SKN7, SMP1, SOK2, SRD1, STB1, STE12, STP1, STP2, SUM1, SWI4, SWI5, SWI6, YAP1, YAP5, YAP6, YFL044C, YJL206C, ZAP1, ZMS1

Source

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., ... & Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular biology of the cell*, 9(12), 3273-3297.

Wang, L., Zhou, J., and Qu, A. (2012). Penalized generalized estimating equations for high-dimensional longitudinal data analysis. *Biometrics*, 68, 353–360.

Examples

```
data(yeastG1)  
head(yeastG1)
```

Index

* datasets

simuGene, [10](#)

yeastG1, [11](#)

CVfit, [2](#)

generateData, [3](#)

PGEE, [4](#)

predict.qpgee, [6](#)

qpgee, [7](#)

qpgee.est, [8](#)

Siga_cov, [10](#)

simuGene, [10](#)

yeastG1, [11](#)