Package ‘geeVerse’

October 13, 2025

Type Package

Title A Comprehensive Analysis of High Dimensional Longitudinal Data

Version 0.3.1

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, MASS

Suggests methods

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.3.2

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 2025-10-13 16:20:02 UTC

Contents

`compile_result` *Compile Results from qpgee()*

Description

This function reports correct percentage, TP, FP, MSE and MAD from a (list of) fitted qpgee model comparing to the true betas.

Usage

```
compile_result(qpgee_results, beta0, threshold = 10^-3)
```

Arguments

`qpgee_results`	A (list of) fitted qpgee model.
`beta0`	True beta used in true data generation process.
`threshold`	Integer, the threshold to determine whether a estimated beta should be consider as 0.

Value

a vector contains correct percentage, TP, FP, MSE and MAD and its standard error if Monte Carlo simulations.

compile_result.default *Compile Results from list of qpgee()*

Description

This function reports correct percentage, TP, FP, MSE and MAD from a (list of) fitted qpgee model comparing to the true betas.

Usage

```
## Default S3 method:
compile_result(qpgee_results, beta0, threshold = 0.1)
```

Arguments

`qpgee_results`	A (list of) fitted qpgee model.
`beta0`	True beta used in true data generation process.
`threshold`	Integer, the threshold to determine whether a estimated beta should be consider as 0.

Value

a vector contains correct percentage, TP, FP, MSE and MAD and its standard error if Monte Carlo simulations.

compile_result.qpgee *Compile Results from qpgee()*

Description

This function reports correct percentage, TP, FP, MSE and MAD from a (list of) fitted qpgee model comparing to the true betas.

Usage

```
## S3 method for class 'qpgee'
compile_result(qpgee_results, beta0, threshold = 10^-3)
```

Arguments

`qpgee_results`	A (list of) fitted qpgee model.
`beta0`	True beta used in true data generation process.
`threshold`	Integer, the threshold to determine whether a estimated beta should be consider as 0.

Value

a vector contains correct percentage, TP, FP, MSE and MAD and its standard error if Monte Carlo simulations.

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,
  pindex,
  eps,
  maxiter,
  tol,
  lambda.vec = exp(seq(log(10), log(0.1), length.out = 30)),
  corstr = "independence",
  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.

`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.

`lambda.vec` a vector of lambda values for which the cross-validation error will be calculated.

`corstr` the correlation structure used.

`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.

`geeVerse` *GeeVerse: Wrapper for Quantile Penalized Generalized Estimating Equations*

Description

This function is a wrapper for qpgee that allows running the model for multiple quantile levels (tau).

Usage

```
geeVerse(x, y, tau = c(0.25, 0.5, 0.75), ...)
```

Arguments

`x` A matrix of predictors.

`y` A numeric vector of response variables.

`tau` A vector of quantiles to be estimated (default is c(0.25, 0.5, 0.75)).

`...` Additional arguments to be passed to qpgee function.

Value

A list containing the results for each tau value and a combined beta matrix.

`generate_data` *Generate Data for Simulation*

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

```
generate_data(
  nsub,
  nobs,
  p,
  beta0,
  rho,
  corstr = "AR1",
  dis = "normal",
  ka = 0,
  SNPs = NULL
)
```

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.
`corstr`	Character, specifies the correlation of correlation structure for the covariance matrix. Options are "cs" or "exchangeable" for compound symmetry, "AR1" for autoregressive, and any other input will result in an identity matrix.
`dis`	Character, the distribution of errors ("normal" or "t").
`ka`	1 for heterogeneous errors and 0 for homogeneous errors.
`SNPs`	User can provide simulated or real SNPs for genetic data simulation.

Value

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

Examples

```
set.seed(123)
sim_data <- generate_data(
  nsub = 50, nobs = rep(5, 50), p = 10,
  beta0 = c(rep(1, 5), rep(0, 5)), rho = 0.3
)
```

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,
  fastginv = 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 correlation 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` correlation 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`.

`fastginv` A logical variable for usage of fast implementation of generalized matrix inverse.

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 <- generate_data(
  nsub = 100, nobs = rep(10, 100), p = 100,
  beta0 = c(rep(1, 7), rep(0, 93)), rho = 0.6, corstr = "AR1",
  dis = "normal", ka = 1
)


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

`print.summary.qpgee` *Print summary method for qpgee model objects*

Description

Print summary method for qpgee model objects

Usage

```
## S3 method for class 'summary.qpgee'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`x`	A ‘qpgee‘ model object.
`digits`	Default digits.
`...`	Additional arguments (not used).

Value

Prints a summary of the qpgee model.

`qpgee` *Quantile Penalized Generalized Estimating Equations (QPGEE)*

Description

Fits a quantile penalized generalized estimating equation (QPGEE) model for longitudinal data using penalized quantile regression with different working correlation structures.

Usage

```
qpgee(x, ...)

## S3 method for class 'formula'
qpgee(x, id, data = parent.frame(), ...)

## Default S3 method:
qpgee(
  x,
  y,
  nobs,
  tau = 0.5,
  corstr = "exchangeable",
  lambda = NULL,
  method = "HBIC",
  intercept = TRUE,
  betaint = NULL,
  nfold = 5,
  ncore = 1,
  control = qpgeeControl(),
  ...
)
```

Arguments

`x`	A matrix of predictors.
`...`	Other arguments passed to methods.
`id`	A vector identifying the clusters (subjects).
`data`	An optional data frame.
`y`	A numeric vector of response variables.
`nobs`	A numeric vector of observations per subject.
`tau`	The quantile to be estimated (default is 0.5).
`corstr`	A string specifying the working correlation structure. Options include "exchangeable" (Exchangeable), "AR1" (Autoregressive), "Tri" (Tri-diagonal), "independence" (Independent), and "unstructured".
`lambda`	A vector of penalty parameters. If NULL, auto-selection is performed.
`method`	Criterion for penalty selection ("HBIC" or "CV").
`intercept`	Logical; if TRUE, an intercept is added.
`betaint`	Initial values for the beta coefficients. If NULL, non-longitudinal quantile regression is used for initialization.
`nfold`	The number of folds used in cross-validation.
`ncore`	Number of cores for parallel processing.
`control`	A list of control parameters from ‘qpgeeControl()‘, such as max_it, epsilon, shrinkCutoff, standardize and trace.

Value

An object of class ‘qpgee‘.

Examples

```
# Quick Example:
# 1. Generate some data
set.seed(123)
sim_data <- generate_data(
  nsub = 50, nobs = rep(5, 50), p = 10,
  beta0 = c(rep(1, 5), rep(0, 5)), rho = 0.3
)

# 2. Fit the model using the formula interface
fit <- qpgee(
  y ~ . - id,
  data = sim_data,
  id = sim_data$id,
  tau = 0.5,
  method = "HBIC"
)

# 3. View the summary of the results
summary(fit)
```

---

qpgeeControl *Control Parameters for qpgee*

Description

Provides control parameters for the Quantile Penalized Generalized Estimating Equations (QPGEE) fitting procedure. Similar in spirit to ‘geese.control()‘ in the geepack package.

Usage

```
qpgeeControl(
  epsilon = 1e-04,
  decay = 1,
  maxit = 100,
  trace = FALSE,
  standardize = FALSE,
  shrinkCutoff = 1e-04
)
```

Arguments

`epsilon` Convergence tolerance for the parameter estimates. Iteration stops when the maximum change in coefficients is below this value.

`decay` Decay rate of learning step.

`maxit` Maximum number of iterations.

`trace` Logical indicating if output should be produced for each iteration. (You can decide how much information to show inside the C++/R loop.)

`standardize` Logical indicating whether to scale X.

`shrinkCutoff` Threshold below which coefficients are shrunk to zero (removal of “small” coefficients).

Value

A list with the components `epsilon`, `maxit`, `trace`, and `shrinkCutoff`.

Examples

```
ctrl <- qpgeeControl(epsilon = 1e-5, maxit = 200, trace = TRUE)
```

`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 "AR1" is specified.

Usage

```
Siga_cov(rho, corstr, nt)
```

Arguments

`rho` Numeric, the correlation coefficient used for generating the covariance matrix. For "cs" or "exchangeable", 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.

`corstr` Character, specifies the correlation of correlation structure for the covariance matrix. Options are "cs" or "exchangeable" 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 *A Simulated Genetic Data from HapGen2*

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