# Package 'VariableScreening'

July 28, 2016

**Type** Package

**Title** High-Dimensional Screening for Semiparametric Longitudinal Regression

**Version** 0.1.1

**Depends** R (>= 3.2.1)

**Description** Implements a screening procedure proposed by Wanghuan Chu, Runze Li and Matthew Reimherr (2016) <DOI:10.1214/16-AOAS912> for varying coefficient longitudinal models with ultra-high dimensional predictors . The effect of each predictor is allowed to vary over time, approximated by a low-dimensional B-spline. Within-subject correlation is handled using a generalized estimation equation approach with structure specified by the user. Variance is allowed to change over time, also approximated by a B-spline.

**Copyright** The Pennsylvania State University

**Imports** gee, expm, splines, MASS

**License** GPL (>= 2)

**LazyData** TRUE

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Runze Li [aut],
Wanghuan Chu [aut],
Liying Huang [aut, cre],
John Dziak [aut]

**Maintainer** Liying Huang <lxh37@PSU.EDU>

**Repository** CRAN

**Date/Publication** 2016-07-28 17:28:27

## R topics documented:

screenlong *Perform high-dimensional screening for semiparametric longitudinal regression*

## Description

Implements a screening procedure proposed by Chu, Li, and Reimherr (2016) for varying coefficient longitudinal models with ultra-high dimensional predictors. The effect of each predictor is allowed to vary over time, approximated by a low-dimensional B-spline. Within-subject correlation is handled using a generalized estimation equation approach with structure specified by the user. Variance is allowed to change over time, also approximated by a B-spline.

## Usage

```
screenlong(x, y, z, id, subset = 1:ncol(x), time, degree = 3, df = 4,
  corstr = "stat_M_dep", M = NULL)
```

## Arguments

| | |
|---|---|
| x | Matrix of features (for example, SNP's). There should be one row for each observation. |
| y | Vector of responses. It should have the same length as the number of rows of x. |
| z | Optional matrix of covariates to be included in all models. They may include demographic covariates such as gender or ethnic background, or some other theoretically important constructs. It should have the same number of rows as the number of rows of x. We suggest a fairly low dimensional z. If the model is intended to include an intercept function (which is recommended), then z should include a column of 1's representing the constant term. |
| id | Vector of integers identifying the subject to which each observation belongs. It should have the same length as the number of rows of x. |
| subset | Vector of integers identifying a subset of the features of x to be screened, the default is 1:ncol(x), i.e., to screen all columns of x. |
| time | Vector of real numbers identifying observation times. It should have the same length as the number of rows of x. We suggest using the convention of scaling time to the interval [0,1]. |
| degree | Degree of the piecewise polynomial for the B-spline basis for the varying coefficient functions; see the documentation for the bs() function in the splines library. |
| df | Degrees of freedom of the B-spline basis for the varying coefficient functions; see the documentation for the bs() function in the splines library. |
| corstr | Working correlation structure for the generalized estimation equations model used to estimate the coefficient functions; see the documentation for the gee() function in the gee library. Options provided by the gee() function include "independence", "fixed", "stat_M_dep", "non_stat_M_dep", "exchangeable", "AR-M" and "unstructured". |

M — An integer indexing the M value (complexity) of the dependence structure, if corstr is M-dependent or AR-M; see the documentation for the gee() function in the gee library. This will be ignored if the correlation structure does not require an M parameter. The default value is set to be 1.

## Value

A list with following components: error A vector of length equal to the number of columns in the input matrix x. It contains sum squared error values for regression models which include the time-varying effects of the z covariates (if any) as well as each x covariate by itself. The lower this error is, the more desirable it is to retain the corresponding x covariate in a later predictive model. rank The rank of the error measures. This will have length equal to the number of columns in the input matrix x, and will consist of a permutation of the integers 1 through that length. A rank of 1 indicates the feature which appears to have the best predictive performance, 2 represents the second best and so on.

## Examples

```
set.seed(12345678)
results <- simulateLD(p=1000)
subset1 <- seq(1,5,2)
subset2 <- seq(100,200,2)
subset3 <- seq(202,400,2)
subset4 <- seq(401,999,2)
set <-c(subset1,subset2,subset3,subset4)
Jmin <- min(table(results$id)) - 1
screenResults <- screenlong(x = results$x,
                            y = results$y,
                            z = results$z,
                            id = results$id,
                            subset = set,
                            time = results$time,
                            degree = 3,
                            df = 4,
                            corstr = "stat_M_dep",
                            M = Jmin
                            )


rank <- screenResults$rank
unlist(rank)
trueIdx <- c(5,100,200,400)
rank[which(set %in% trueIdx)]
```

---

simulateLD *Simulate a dataset for testing the performance of screenlong*

---

## Description

Simulates a dataset that can be used to test the screenlong function, and to test the performance of the proposed method under different scenarios. The simulated dataset has two z-covariates and p x-covariates, only a few of which have nonzero effect. There are n subjects in the simulated dataset, each having J observations, which are not necessarily evenly timed, we randomly draw a subset to create an unbalanced dataset. The within-subject correlation is assumed to be AR-1.

## Usage

```
simulateLD(n = 100, J = 10, rho = 0.6, p = 500, trueIdx = c(5, 100,
  200, 400), beta0Fun = NULL, betaFun = NULL, gammaFun = NULL,
  varFun = NULL)
```

## Arguments

| | |
|---|---|
| n | Number of subjects in the simulated dataset |
| J | Number of observations per subject |
| rho | The correlation parameter for the AR-1 correlation structure. |
| p | The total number of features to be screened from |
| trueIdx | The indexes for the active features in the simulated x matrix. This should be a vector, and the values should be a subset of 1:p. |
| beta0Fun | The time-varying intercept for the data-generating model, as a function of time. If left as null, it will default to `f(t) 2 * t^2 - 1`. Time is assumed to be scaled to the interval [0,1]. |
| betaFun | The time-varying coefficients for z in the data-generating model, as a function of time. If left as null, it will be specified as two functions. The first is `f(t) exp(t + 1)/2`. The second is `f(t) t^2 + 0.5`. Time is assumed to be scaled to the interval [0,1]. |
| gammaFun | A list of functions of time, one function for each entry in trueIdx, giving the time-varying effects of each active feature in the simulated x matrix. If left as null, it will be specified as four functions. The first is a step function `f(t)=(t > 0.4)`. The second is `f(t)=- cos(2 * pi * t)`. the third is `f(t)=(2 - 3 * t)^2/2 - 1`. The fourth is `f(t)=sin(2 * pi * t)`. |
| varFun | A function of time telling the marginal variance of the error function at a given time. If left as null, it will be specified as `function(t) 0.5 + 3 * t^3`. |

## Value

A list with following components: x Matrix of features to be screened. It will have n*J rows and p columns. y Vector of responses. It will have length of n*J. z A matrix representing covariates to be included in each of the screening models. The first column will be all ones, representing the intercept. The second will consist of random ones and zeros, representing simulated genders. id Vector of integers identifying the subject to which each observation belongs. time Vector of real numbers identifying observation times. It should have the same length as the number of rows of x.

## Examples

```
set.seed(12345678)
results <- simulateLD(p=1000)
```

# Index