Package 'mvecr'

September 1, 2019

Type Package

Title Multivariate Error Components Regression

Version 0.1.0

Author Yuan Yue

Maintainer Yuan Yue <yuanyue.work@gmail.com>

Description An implementation of multivariate error components regression in R. It converts individual level data to a 'stacked form' panel data which facilitates regression.
The dependent variable as well as independent variables are assumed to be multivariate. The error structure is assumed to consist of multiple (usually two or three) indepedent multivariate error components. This structure enables breaking down the variance matrix into matrices of much smaller dimension.

License GPL-3

Depends R (>= 3.5)

Imports dplyr (>= 0.7.8)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

ec_reg *ec_reg.R*

Description

This function implements the main optimization procedure for the regression model with multivariate error components.

Usage

```
ec_reg(data_X, data_y, data_H, colName.i = "i", colName.t = "t",
  colName.p = "Type", district, time, type, var, par.include = rep(1,
  18), par.init = rep(0.5, 18))
```

Arguments

data_X	Matrix of independent variables.
data_y	Vector of dependent variables.
data_H	Vector of number of observations in each i-t-p combination.
colName.i	Column name in data that contains district information.
colName.t	Column name in data that contains time information.
colName.p	Column name in data that contains type information.
district	Unique name of districts.
time	Unique time periods in the data.
type	Unique names of the type in the data.
var	Vector of names of the columns in data_X to include in the regression.
par.include	A vector of logical values indicating whether or not to include a certain parameter in the regression. If FALSE, the parameter is constrained to be 0.
par.init	A vector of initial value of the parameters. Default value is 0.5 for each parameter. The number of parameter is determined by $p*(1+p)/2*[numberoferrorcomponents]$.

Value

results, a dataframe containing parameter estimates and t statistics.

gradient *Gradient of the loglikelihood function*

Description

This function is the gradient function of the full model (with $3 * p * (p + 1)/2$ parameters).

Usage

```
gradient(par, include, N, Tn, HX, Hy, p = 3)
```

Arguments

par	A vector of parameters (upper diagonal elements of the Cholesky decomposition of the error matrices).
include	A vector of logical values the same dimension as par, indicating whether the parameter should be included as one of the values to optimize. If a certain element corresponds to FALSE, that parameter is constrained to be 0.
N	A numeric scalar, number of districts.
Tn	A numeric scalar, number of time periods.
HX	Matrix of independent variables.
Hy	Matrix of dependent variables.
p	A numeric scalar, number of dimensions (covariates).

Value

A vector of gradient.

---

negloglik *Negative loglikelihood function*

Description

This function is the negative loglikelihood function of the full model (with $3 * p * (p + 1)/2$ parameters).

Usage

```
negloglik(par, include, N, Tn, HX, Hy, p = 3)
```

Arguments

par	A vector of parameters (upper diagonal elements of the Cholesky decomposition of the error matrices).
include	A vector of logical values the same dimension as par, indicating whether the parameter should be included as one of the values to optimize. If a certain element corresponds to FALSE, that parameter is constrained to be 0.
N	A numeric scalar, number of districts.
Tn	A numeric scalar, number of time periods.
HX	Matrix of independent variables.
Hy	Matrix of dependent variables.
p	A numeric scalar, number of dimensions (covariates).

Value

A numeric scalar.

opt_psi *opt_psi.R*

Description

Given a list of parameter psi's, this function finds the psi that maximizes loglikelihood.

Usage

```
opt_psi(data_X, data_y, data_H, list_psi = 1, method = "p", district,
  time, type, var, par.include = rep(1, 18), par.init = rep(0.5, 18),
  plot = FALSE)
```

Arguments

data_X	Matrix of independent variables.
data_y	Vector of dependent variables.
data_H	Vector of number of observations in each i-t-p combination.
list_psi	A numeric vector of the psi parameter.
method	A character indicating which weighting function to choose. Options are "p" (prelec function) or "t" (tversky function).
district	Unique name of districts.
time	Unique time periods in the data.
type	Unique names of the type in the data.
var	Vector of names of the columns in data_X to include in the regression.
par.include	A vector of logical values indicating whether or not to include a certain parameter in the regression. If FALSE, the parameter is constrained to be 0.
par.init	A vector of initial value of the parameters. Default value is 0.5 for each parameter. The number of parameter is determined by $p*(1+p)/2*[numberoferrorcomponents]$.
plot	A logical value indicating whether or not to plot a simple graph of the likelihood curve.

Value

Numeric value of the psi parameter that maximizes loglikelihood.

prelec *Prelec probability weighting function*

Description

A function that converts probability into weighted probability using the Prelec(1998) functional form.

Usage

```
prelec(p, psi, psi2 = 1)
```

Arguments

p	Original probability, numeric value between 0 and 1.
psi	Parameter of the function (single parameter form).
psi2	Optional, second parameter of the function (two parameter form). Default value is 1 in which case the function degenerates into single parameter form.

Value

Weighted probability, numeric value between 0 and 1.

References

Prelec D. The probability weighting function[J]. ECONOMETRICA-EVANSTON ILL-, 1998, 66: 497-528.

Examples

```
prelec(0.2, 0.5)
prelec(0.2, 5)
```

---

reg_psi *reg_psi.R*

Description

When one of the regressors depend on a parameter (psi), adjust the corresponding variance of beta and calculate the variance of psi. Here we assume Xpsi(psi) is a function of Xpsi_obj with parameter psi. The functional form can be a Prelec function (method="p") or a Tversky-Kahneman function (method="t").

Usage

```
reg_psi(data_X, data_y, data_H, colName.i = "i", colName.t = "t",
  colName.p = "Type", psi = 1, method = "p", district, time, type,
  var, par.include = rep(1, 18), par.init = rep(0.5, 18))
```

Arguments

data_X	Matrix of independent variables.
data_y	Vector of dependent variables.
data_H	Vector of number of observations in each i-t-p combination.
colName.i	Column name in data that contains district information.
colName.t	Column name in data that contains time information.
colName.p	Column name in data that contains type information.
psi	A numeric value of the psi parameter.
method	A character indicating which weighting function to choose. Options are "p" (prelec function) or "t" (tversky function).

district	Unique name of districts.
time	Unique time periods in the data.
type	Unique names of the type in the data.
var	Vector of names of the columns in data_X to include in the regression.
par.include	A vector of logical values indicating whether or not to include a certain parameter in the regression. If FALSE, the parameter is constrained to be 0.
par.init	A vector of initial value of the parameters. Default value is 0.5 for each parameter. The number of parameter is determined by $p*(1+p)/2*[numberoferrorcomponents]$.

Value

results, a dataframe containing parameter estimates, t statistics, loglikelihood value, value of psi (given as input), and the t-statistic w.r.t. psi.

---

tversky *Tversky and Kahneman probability weighting function*

Description

A function that converts probability into weighted probability using the Tversky and Kahneman (1992) functional form.

Usage

```
tversky(p, psi = 1)
```

Arguments

p	Original probability, numeric value between 0 and 1.
psi	Parameter of the function (single parameter form). Default value is 1 in which case the function degenerates into identity function w(p)=p.

Value

Weighted probability, numeric value between 0 and 1.

References

Tversky A, Kahneman D. Advances in prospect theory: Cumulative representation of uncertainty[J]. Journal of Risk and uncertainty, 1992, 5(4): 297-323.

Examples

```
tversky(0.2, 0.5)
tversky(0.2, 5)
```

`vectorize` *vectorize.R*

Description

This function reads the data frames containing y, X, and information for i(district), t(time period) and p(type), and take the average over each i-t-p combination.

Usage

```
vectorize(data, colName.i, colName.t, colName.p, colName.y, colName.X)
```

Arguments

`data`	Input data frame.
`colName.i`	Column name in data that contains district information.
`colName.t`	Column name in data that contains time information.
`colName.p`	Column name in data that contains type information.
`colName.y`	Column name in data that contains the dependent variable.
`colName.X`	Column name in data that contains the independent variables.

Value

A list containing: Unique name of districts, unique time periods, unique type names, Vector y, matrix X, and H (number of observations for each i-t-p combination).

`X_invOmega_Y` *X_invOmega_Y.R*

Description

This function calculates the functions in the form $X'(Omega)^{-1}y$.

Usage

```
X_invOmega_Y(X, Y, N, Tn, p, O1inv, O2inv, O3inv, O4inv)
```

Arguments

`X`	Matrix or vector of dimension $NTp \times k_1$.
`Y`	Matrix or vector of dimension $NTp \times k_2$.
`N`	A numeric scalar, number of districts.
`Tn`	A numeric scalar, number of time periods.
`p`	A numeric scalar, number of dimensions (covariates).
`O1inv`	A matrix of dimension $p \times p$, component of the Ω matrix.
`O2inv`	A matrix of dimension $p \times p$, component of the Ω matrix.
`O3inv`	A matrix of dimension $p \times p$, component of the Ω matrix.
`O4inv`	A matrix of dimension $p \times p$, component of the Ω matrix.

Value

$X\Omega^{-1}Y$

`Z_prelec` *Derivative of the Prelec probability weighting function*

Description

A function that calculates the first derivative of the probability weighting function w.r.t. its parameter.

Usage

```
Z_prelec(p, psi, log = FALSE)
```

Arguments

`p`	Original probability, numeric value between 0 and 1.
`psi`	Parameter of the function (single parameter form).
`log`	Logical value indicating whether the weighted probability is taken natural logarithm. Default value is FALSE.

Value

Derivative of the Prelec function w.r.t. psi evaluated at p, a numeric value.

Examples

```
Z_prelec(0.5,3)
Z_prelec(0.5, 0.5, log = TRUE)
```

`Z_tversky` *Derivative of the Tversky and Kahneman probability weighting function*

Description

A function that calculates the first derivative of the probability weighting function w.r.t. its parameter.

Usage

```
Z_tversky(p, psi, log = FALSE)
```

Arguments

`p`	Original probability, numeric value between 0 and 1.
`psi`	Parameter of the function (single parameter form).
`log`	Logical value indicating whether the weighted probability is taken natural logarithm. Default value is FALSE.

Value

Derivative of the Tversky and Kahneman function w.r.t. psi evaluated at p, a numeric value.

Examples

```
Z_tversky(0.5,3)
Z_tversky(0.5, 0.5, log = TRUE)
```

Index