Package 'SIRE'

July 30, 2018

Type Package

Title Finding Feedback Effects in SEM and Testing for Their Significance

Version 1.0.2

Maintainer Gianmarco Vacca <gianmarco.vacca@unicatt.it>

Description Provides two main functionalities.
1 - Given a system of simultaneous equation,
it decomposes the matrix of coefficients weighting the endogenous variables
into three submatrices: one includes the subset of coefficients that have a causal nature
in the model, two include the subset of coefficients that have a interdependent nature
in the model, either at systematic level or induced by the correlation between error terms.
2 - Given a decomposed model,
it tests for the significance of the interdependent relationships acting in the system,
via Maximum likelihood and Wald test, which can be built starting from the function output.
For theoretical reference see Faliva (1992) <doi:10.1007/BF02589085> and
Faliva and Zoia (1994) <doi:10.1007/BF02589041>.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.1.0)

Imports systemfit, psych, igraph, matrixcalc, MASS, numDeriv, Matrix, stringr

RoxygenNote 6.0.1

NeedsCompilation no

Author Gianmarco Vacca [aut, cre]

Repository CRAN

Date/Publication 2018-07-30 13:20:03 UTC

R topics documented:

`causal.decompose` *Estimation and decomposition of simultaneous equation model*

Description

Estimate and decompose a Simultaneous Equation Model into its recursive and Interdependent subsystems

Usage

```
causal.decompose(data, eq.system, resid.est = "noDfCor", instruments,
  p.adj = TRUE, alpha = 0.05)
```

Arguments

`data`	the data frame containing the data
`eq.system`	the system of equations (a list of formula objects, e.g. as in pkg systemfit)
`resid.est`	the estimation methods for the residual covariance matrix (as in pkg systemfit)
`instruments`	the intruments used to estimate the model via 3-SLS (as in pkg systemfit)
`p.adj`	logical indication of whether to adjust the significance level for multiple comparisons, when testing the significance of the elements of Σ
`alpha`	the significance level α for the tests on the elements of Σ

Value

A list with components

- `output`: a list containing
  - `eq.system`: the system of equations given as input
  - `Gamma`: the 3-SLS estimate of Γ'
  - `C`: the matrix highlighting the interdependent mechanisms at deterministic level.
  - `Psi1`: the matrix highlighting the interdependent mechanisms at stochastic level.
  - `Psi0`: the matrix highlighting the causal mechanisms.
  - `A`: the 3-SLS estimate of A
  - `Sigma`: the 3-SLS estimate of $Sigma$, with non-significant elements set to zero
  - `systemfit`: the output from the `systemfit` function used to estimate the model
  - `corrtest`: the output from the `corr.test` function used to test the significance on the elements of the residual correlation matrix

  - `all.graph`: the path diagram of the model, using the package `igraph`
  - `dec.graph`: the path diagram of the decomposed model, with color coding for each vertex
- `path`: a data frame in which every row is a path coefficient, along with indication of the nature of the link (recursive/interdependent) and the index of the equation in which it belongs

Examples

```
data("macroIT")
eq.system = list(
               eq1 = C ~  CP  + I + CP_1,
               eq2 = I ~ K + CP_1,
               eq3 = WP ~ I + GDP + GDP_1,
               eq4 = GDP ~ C + I + GDP_1,
               eq5 = CP ~ WP + T,
               eq6 = K ~ I + K_1)

instruments = ~ T + CP_1 + GDP_1 + K_1

causal.decompose(data = macroIT,
               eq.system = eq.system,
               resid.est = "noDfCor",
               instruments = instruments,
               p.adj = TRUE,
               alpha = 0.05)
```

`causal.decompose.sim` *Decomposition of simultaneous equation model*

Description

Decompose a Simultaneous Equation Model into its recursive and Interdependent sub-systems

Usage

```
causal.decompose.sim(eq.system, Sigma = NULL)
```

Arguments

`eq.system`	the system of equations (a list of formula objects)
`Sigma`	the Σ matrix

Value

A list with components

- `output`: a list containing
  - `eq.system`: the system of equations given as input
  - `Gamma`: the binary matrix $\Gamma^{b'}$

  - C: the binary matrix highlighting the interdependent mechanisms at deterministic level.
  - Psi1: the binary matrix highlighting the interdependent mechanisms at stochastic level.
  - Psi0: the binary matrix highlighting the causal mechanisms.
  - Sigma: the matrix Σ given as input
  - all.graph: the path diagram of the model, using the package igraph
  - dec.graph: the path diagram of the decomposed model, with color coding for each vertex. Blue means causality, red means interdependence, dashed red means interdependence by effect of error correlation
- path: a data frame in which every row is a path coefficient, along with indication of the nature of the link (recursive/interdependent) and the index of the equation in which it belongs

Examples

```
eq.system = list(
            eq1 = y1 ~ y5 + y7,
            eq2 = y2 ~ z,
            eq3 = y3 ~ y11,
            eq4 = y4 ~ y3,
            eq5 = y5 ~ y10,
            eq6 = y6 ~ y5 + y9,
            eq7 = y7 ~ y6,
            eq8 = y8 ~ y12,
            eq9 = y9 ~ y7,
            eq10 = y10 ~ y5,
            eq11 = y11 ~ y12,
            eq12 = y12 ~ y4 + y11,
            eq13 = y13 ~ y2 + y6)

 # indexes of non-null elements of Sigma
 sigma.idx = cbind(c(2,1),
                   c(1,5),
                   c(13,2),
                   c(2,13),
                   c(5,1),
                   c(1,2))

 #fictitious Sigma matrix
 Sigma = as.matrix(
         Matrix::sparseMatrix(i = sigma.idx[1,] , j = sigma.idx[2,], x = 0.1)) +
         diag(length(eq.system))

 causal.decompose.sim(eq.system , Sigma)
```

---

dec.calc *Decomposition starting from Gamma and Sigma*

Description

Function to decompose Γ' into recursive and interdependent sub-matrices (internal use)

Usage

```
dec.calc(Gamma, Sigma)
```

Arguments

Gamma	the Γ' matrix.
Sigma	the Σ matrix.

Value

A list with components

- C: the matrix highlighting the interdependent mechanisms at deterministic level.
- Psi1: the matrix highlighting the interdependent mechanisms at stochastic level.
- Psi0: the matrix highlighting the causal mechanisms.
- powers: a list containing the matrix powers of Γ'.

---

feedback.ml *Testing for Feedback Effects in a Simultaneous Equation Model*

Description

Testing for Feedback Effects in a Simultaneous Equation Model

Usage

```
feedback.ml(data, out.decompose, eq.id, seed.in = 10, n.init = 50,
  range.init = c(-0.5, 0.5), info = T, maxit = 20000, perc.print = 0)
```

Arguments

data	the data frame containing the data
out.decompose	the decomposition object resulting from causal.decompose()
eq.id	the equation to be tested for feedback effects
seed.in	seed number for the perturbation parameters draws
n.init	number of different initializations of the parameters
range.init	range of the Uniform distribution for the perturbation on the initial values
info	logical, prints the trace of the Nelder-Mead optimization routine
maxit	maximum number of iterations for the Nelder-Mead optimization routine
perc.print	print option: decimal number indicating how often (in percentage) information on the progress must be printed.

Value

A list with components

- `rho.est`: a data frame with the maximum likelihood estimate of rho and the equations with which each element is involved in feedback-like mechanisms
- `loglik`: the value of the log-likelihood
- `theta.hessian`: the hessian matrix for the estimated parameters
- `rho.jacobian`: the Jacobian matrix of ρ with respect to the entire set of parameters

Examples

```
data("macroIT")
eq.system = list(
              eq1 = C ~  CP  + I + CP_1,
              eq2 = I ~ K + CP_1,
              eq3 = WP ~ I + GDP + GDP_1,
              eq4 = GDP ~ C + I + GDP_1,
              eq5 = CP ~ WP + T,
              eq6 = K ~ I + K_1)

instruments = ~ T + CP_1 + GDP_1 + K_1

c.dec = causal.decompose(data = macroIT,
                         eq.system = eq.system,
                         resid.est = "noDfCor",
                         instruments = instruments,
                         p.adj = TRUE,
                         alpha = 0.05)

feedback.ml(data = macroIT,
              out.decompose = c.dec,
              eq.id = 2,
              seed.in = 10,
              n.init = 50,
              range.init = c(-1,1),
              info = TRUE,
              maxit = 1000,
              perc.print = 0)
```

`macroIT` *Italian Macroeconomic Data*

Description

Italian macroeconomic variables from Q3-1996 to Q2-2011 (T = 60 observations). The variables are

- `QTR`: quarter and year of the observation
- `C`: expenses for consumption for Italian families
- `CP`: value added
- `WP`: private wages from dependent employment
- `I`: gross investment
- `K`: gross capital stock
- `GDP`: gross domestic product
- `T`: taxes
- `CP_1`: lagged value added
- `K_1`: lagged gross capital stock
- `GDP_1`: lagged gross domestic product

Usage

```
data(macroIT)
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 60 rows and 11 columns.

Source

http://dati.istat.it/

Examples

```
data(macroIT)
```

---

rho.calc *Rho Calculation*

Description

Function to calculate rho (internal use)

Usage

```
rho.calc(l, Gamma, A, Sigma)
```

Arguments

`l`	the equation index for which to calculate rho
`Gamma`	the Γ' matrix
`A`	the A matrix
`Sigma`	the Σ matrix

Value

A list with components

- `S0`: the selection matrix for p_j.
- `S1`: the selection matrix for Γ'.
- `S2`: the selection matrix. for A

Index