Package ‘sparsevar’

October 14, 2022

Version 0.1.0
Date 2021-04-16
Title Sparse VAR/VECM Models Estimation
Maintainer Simone Vazzoler <svazzole@gmail.com>
Imports Matrix, ncvreg, parallel, doParallel, glmnet, ggplot2,
reshape2, grid, mvtnorm, picasso, corpcor,
Suggests knitr, rmarkdown, testthat,
Depends R (>= 3.5.0)
Description A wrapper for sparse VAR/VECM time series models estimation
using penalties like ENET (Elastic Net), SCAD (Smoothly Clipped
Absolute Deviation) and MCP (Minimax Concave Penalty).
Based on the work of Sumanta Basu and George Michailidis
License GPL-2
URL https://github.com/svazzole/sparsevar
BugReports https://github.com/svazzole/sparsevar
VignetteBuilder knitr
RoxygenNote 7.1.1
Encoding UTF-8
NeedsCompilation no
Author Simone Vazzoler [aut, cre]
Repository CRAN
Date/Publication 2021-04-18 04:50:02 UTC

R topics documented:

accuracy ................................................................. 2
bootstrappedVAR .................................................... 3
checkImpulseZero .................................................. 3
**accuracy**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute the accuracy of a fit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>accuracy(referenceM, A)</code></td>
</tr>
</tbody>
</table>
**bootstrappedVAR**

**Arguments**

- referenceM: the matrix to use as reference
- A: the matrix obtained from a fit

**Description**

Build the bootstrapped series from the original var

**Usage**

`bootstrappedVAR(v)`

**Arguments**

- v: the VAR object as from fitVAR or simulateVAR

---

**checkImpulseZero**

**Check Impulse Zero**

**Description**

A function to find which entries of the impulse response function are zero.

**Usage**

`checkImpulseZero(irf)`

**Arguments**

- irf: irf output from impulseResponse function

**Value**

A matrix containing the indices of the impulse response function that are 0.
checkIsVar

**Check is var**

**Description**
Check if the input is a var object

**Usage**
checkIsVar(v)

**Arguments**
- **v**
  the object to test

companionVAR

**Companion VAR**

**Description**
Build the VAR(1) representation of a VAR(p) process

**Usage**
companionVAR(v)

**Arguments**
- **v**
  the VAR object as from fitVAR or simulateVAR

computeForecasts

**Computes forecasts for VARs**

**Description**
This function computes forecasts for a given VAR.

**Usage**
computeForecasts(v, num_steps)

**Arguments**
- **v**
  a VAR object as from fitVAR.
- **num_steps**
  the number of forecasts to produce.
createSparseMatrix

Create Sparse Matrix

Description

Creates a sparse square matrix with a given sparsity and distribution.

Usage

createSparseMatrix(
  N,      # the dimension of the square matrix
  sparsity,  # the density of non zero elements
  method = "normal",  # the method used to generate the entries of the matrix. Possible values are "normal" (default) or "bimodal".
  stationary = FALSE,  # should the spectral radius of the matrix be smaller than 1? Possible values are TRUE or FALSE. Default is FALSE.
  p = 1,  # normalization constant (used for VAR of order greater than 1, default = 1)
  ...  # other options for the matrix (you can specify the mean mu_mat and the standard deviation sd_mat).
)

Arguments

- **N**: the dimension of the square matrix
- **sparsity**: the density of non zero elements
- **method**: the method used to generate the entries of the matrix. Possible values are "normal" (default) or "bimodal".
- **stationary**: should the spectral radius of the matrix be smaller than 1? Possible values are TRUE or FALSE. Default is FALSE.
- **p**: normalization constant (used for VAR of order greater than 1, default = 1)
- **...**: other options for the matrix (you can specify the mean mu_mat and the standard deviation sd_mat).

Value

An NxN sparse matrix.

Examples

M <- createSparseMatrix(
  N = 30, sparsity = 0.05, method = "normal",
  stationary = TRUE
)
decomposePi  Decompose Pi VECM matrix

Description
A function to estimate a (possibly big) multivariate VECM time series using penalized least squares methods, such as ENET, SCAD or MC+.

Usage
decomposePi(vecm, rk, ...)

Arguments
vecm  the VECM object
rk  rank
...  options for the function (TODO: specify)

Value
alpha
beta

errorBandsIRF  Error bands for IRF

Description
A function to estimate the confidence intervals for irf and oirf.

Usage
errorBandsIRF(v, irf, alpha, M, resampling, ...)

Arguments
v  a var object as from fitVAR or simulateVAR
irf  irf output from impulseResponse function
alpha  level of confidence (default alpha = 0.01)
M  number of bootstrapped series (default M = 100)
resampling  type of resampling: "bootstrap" or "jackknife"
...  some options for the estimation: verbose = TRUE or FALSE, mode = "fast" or "slow", threshold = TRUE or FALSE.
Value

A matrix containing the indices of the impulse response function that are 0.

**fitVAR**

**Multivariate VAR estimation**

**Description**

A function to estimate a (possibly high-dimensional) multivariate VAR time series using penalized least squares methods, such as ENET, SCAD or MC+.

**Usage**

```r
fitVAR(data, p = 1, penalty = "ENET", method = "cv", ...) 
```

**Arguments**

- **data** the data from the time series: variables in columns and observations in rows
- **p** order of the VAR model
- **penalty** the penalty function to use. Possible values are "ENET", "SCAD" or "MCP"
- **method** possible values are "cv" or "timeSlice"
- **...** the options for the estimation. Global options are: threshold: if TRUE all the entries smaller than the oracle threshold are set to zero; scale: scale the data (default = FALSE); nfolds: the number of folds used for cross validation (default = 10); parallel: if TRUE use multicore backend (default = FALSE); ncores: if parallel is TRUE, specify the number of cores to use for parallel evaluation. Options for ENET estimation: alpha: the value of alpha to use in elastic net (0 is Ridge regression, 1 is LASSO (default)); type.measure: the measure to use for error evaluation ("mse" or "mae"); nlambda: the number of lambdas to use in the cross validation (default = 100); leaveOut: in the time slice validation leave out the last leaveOutLast observations (default = 15); horizon: the horizon to use for estimating mse/mae (default = 1); picasso: use picasso package for estimation (only available for penalty = "SCAD" and method = "timeSlice").

**Value**

- **fit** the results of the penalized LS estimation
- **mse** the mean square error of the cross validation
- **time** elapsed time for the estimation
- **residuals** the time series of the residuals
**fitVARX**  
*Multivariate VARX estimation*

**Description**
A function to estimate a (possibly high-dimensional) multivariate VARX time series using penalized least squares methods, such as ENET, SCAD or MC+.

**Usage**

```r
fitVARX(data, p = 1, Xt, m = 1, penalty = "ENET", method = "cv", ...)
```

**Arguments**
- `data`: the data from the time series: variables in columns and observations in rows
- `p`: order of the VAR model
- `Xt`: the exogenous variables
- `m`: order of the exogenous variables
- `penalty`: the penalty function to use. Possible values are "ENET", "SCAD" or "MCP"
- `method`: possible values are "cv" or "timeSlice"
- `...`: the options for the estimation. Global options are: `threshold`: if `TRUE` all the entries smaller than the oracle threshold are set to zero; `scale`: scale the data (default = FALSE)? `nfold`: the number of folds used for cross validation (default = 10); `parallel`: if `TRUE` use multcore backend (default = FALSE); `ncore`: if `parallel` is `TRUE`, specify the number of cores to use for parallel evaluation. Options for ENET estimation: `alpha`: the value of alpha to use in elastic net (0 is Ridge regression, 1 is LASSO (default)); `type.measure`: the measure to use for error evaluation ("mse" or "mae"); `nlambda`: the number of lambdas to use in the cross validation (default = 100); `leaveOut`: in the time slice validation leave out the last `leaveOut` observations (default = 15); `horizon`: the horizon to use for estimating `mse/mae` (default = 1); `picasso`: use picasso package for estimation (only available for `penalty = "SCAD"` and `method = "timeSlice"`).

**Value**
- `fit`: the list (of length `p`) of the estimated matrices of the process
- `mse`: the mean square error of the cross validation
- `time`: elapsed time for the estimation
- `residuals`: the time series of the residuals
fitVECM

**Multivariate VECM estimation**

**Description**
A function to estimate a (possibly big) multivariate VECM time series using penalized least squares methods, such as ENET, SCAD or MC+.

**Usage**
```r
fitVECM(data, p, penalty, method, logScale, ...)
```

**Arguments**
- **data**: the data from the time series: variables in columns and observations in rows
- **p**: order of the VECM model
- **penalty**: the penalty function to use. Possible values are "ENET", "SCAD" or "MCP"
- **method**: "cv" or "timeSlice"
- **logScale**: should the function consider the log of the inputs? By default this is set to TRUE
- **...**: options for the function (TODO: specify)

**Value**
- **P**: the matrix \( P \) for the VECM model
- **G**: the list (of length \( p-1 \)) of the estimated matrices of the process
- **fit**: the results of the penalized LS estimation
- **mse**: the mean square error of the cross validation
- **time**: elapsed time for the estimation

---

**frobNorm**

**Frobenius norm of a matrix**

**Description**
Compute the Frobenius norm of \( M \)

**Usage**
```r
frobNorm(M)
```

**Arguments**
- **M**: the matrix (real or complex valued)
**impulseResponse**  
*Impulse Response Function*

**Description**

A function to estimate the Impulse Response Function of a given VAR.

**Usage**

```r
impulseResponse(v, len = 20)
```

**Arguments**

- `v`: the data in the form of a VAR
- `len`: length of the impulse response function

**Value**

`irf` a 3d array containing the impulse response function.

---

**informCrit**  
*Computes information criteria for VARs*

**Description**

This function computes information criteria (AIC, Schwartz and Hannan-Quinn) for VARs.

**Usage**

```r
informCrit(v)
```

**Arguments**

- `v`: a list of VAR objects as from `fitVAR`. 
**Description**

Compute the L1 matrix norm of M

**Usage**

```r
l1norm(M)
```

**Arguments**

- `M` the matrix (real or complex valued)

---

**Description**

Compute the L2 matrix norm of M

**Usage**

```r
l2norm(M)
```

**Arguments**

- `M` the matrix (real or complex valued)

---

**Description**

Compute the L-infinity matrix norm of M

**Usage**

```r
lInftyNorm(M)
```

**Arguments**

- `M` the matrix (real or complex valued)
maxNorm  \hspace{1cm} \textit{Max-norm of a matrix}

**Description**

Compute the max-norm of M

**Usage**

```r
maxNorm(M)
```

**Arguments**

- `M` the matrix (real or complex valued)

mcSimulations  \hspace{1cm} \textit{Monte Carlo simulations}

**Description**

This function generates Monte Carlo simulations of sparse VAR and its estimation (at the moment only for VAR(1) processes).

**Usage**

```r
mcSimulations(
  N,
  nobs = 250,
  nMC = 100,
  rho = 0.5,
  sparsity = 0.05,
  penalty = "ENET",
  covariance = "Toeplitz",
  method = "normal",
  modelSel = "cv",
  ...
)
```

**Arguments**

- `N` dimension of the multivariate time series.
- `nobs` number of observations to be generated.
- `nMC` number of Monte Carlo simulations.
- `rho` base value for the covariance.
multiplot

**sparsity**
- density of non zero entries of the VAR matrices.

**penalty**
- penalty function to use for LS estimation. Possible values are "ENET", "SCAD" or "MCP".

**covariance**
- type of covariance matrix to be used in the generation of the sparse VAR model.

**method**
- which type of distribution to use in the generation of the entries of the matrices.

**modelSel**
- select which model selection criteria to use ("cv" or "timeslice").

... (TODO: complete)

**Value**
- a nMcx5 matrix with the results of the Monte Carlo estimation

---

**multiplot**

*Multiplots with ggplot*

**Description**

Multiple plot function. ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)

**Usage**

`multiplot(..., plotlist = NULL, cols = 1, layout = NULL)`

**Arguments**

- ... a sequence of ggplots to be plotted in the grid.
- plotlist a list containing ggplots as elements.
- cols number of columns in layout
- layout a matrix specifying the layout. If present, 'cols' is ignored. If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE), then plot 1 will go in the upper left, 2 will go in the upper right, and 3 will go all the way across the bottom. Taken from R Cookbook

**Value**

A ggplot containing the plots passed as arguments
plotIRF

IRF plot

Description
Plot a IRF object

Usage
plotIRF(irf, eb, i, j, type, bands)

Arguments
irf the irf object to plot
eb the errorbands to plot
i the first index
j the second index
type type = "irf" or type = "oirf"
bands "quantiles" or "sd"

Value
An image plot relative to the impulse response function.

plotIRFGrid
IRF grid plot

Description
Plot a IRF grid object

Usage
plotIRFGrid(irf, eb, indexes, type, bands)

Arguments
irf the irf object computed using impulseResponse
eb the error bands estimated using errorBands
indexes a vector containing the indices that you want to plot
type plot the irf (type = "irf" by default) or the orthogonal irf (type = "oirf")
bands which type of bands to plot ("quantiles" (default) or "sd")

Value
An image plot relative to the impulse response function.
plotMatrix

Matrix plot

Description
Plot a sparse matrix

Usage
plotMatrix(M, colors)

Arguments
M the matrix to plot
colors dark or light

Value
An image plot with a particular color palette (black zero entries, red for the negative ones and green for the positive)

plotVAR
Plot VARs

Description
Plot all the matrices of a VAR model

Usage
plotVAR(..., colors)

Arguments
... a sequence of VAR objects (one or more than one, as from simulateVAR or fitVAR)
colors the gradient used to plot the matrix. It can be "light" (low = red – mid = white – high = blue) or "dark" (low = red – mid = black – high = green)

Value
An image plot with a specific color palette
plotVECM  

Plot VECMs

Description
Plot all the matrices of a VECM model

Usage
plotVECM(v)

Arguments
v a VECM object (as from fitVECM)

Value
An image plot with a specific color palette (black zero entries, red for the negative ones and green for the positive)

simulateVAR  

VAR simulation

Description
This function generates a simulated multivariate VAR time series.

Usage
simulateVAR(N, p, nobs, rho, sparsity, mu, method, covariance, ...)

Arguments
N dimension of the time series.
p number of lags of the VAR model.
nobs number of observations to be generated.
rho base value for the covariance matrix.
sparsity density (in percentage) of the number of nonzero elements of the VAR matrices.
mu a vector containing the mean of the simulated process.
method which method to use to generate the VAR matrix. Possible values are "normal" or "bimodal".
covariance type of covariance matrix to use in the simulation. Possible values: "toeplitz", "block1", "block2" or simply "diagonal".
... the options for the simulation. These are: muMat: the mean of the entries of the VAR matrices; sdMat: the sd of the entries of the matrices;
**simulateVARX**

### Description
This function generates a simulated multivariate VAR time series.

### Usage
```
simulateVARX(N, K, p, m, nobs, rho, 
               sparsityA1, sparsityA2, sparsityA3, 
               mu, method, covariance, ...)
```

### Arguments
- **N**: dimension of the time series.
- **K**: TODO
- **p**: number of lags of the VAR model.
- **m**: TODO
- **nobs**: number of observations to be generated.
- **rho**: base value for the covariance matrix.
- **sparsityA1**: density (in percentage) of the number of nonzero elements of the A1 block.
- **sparsityA2**: density (in percentage) of the number of nonzero elements of the A2 block.
- **sparsityA3**: density (in percentage) of the number of nonzero elements of the A3 block.
- **mu**: a vector containing the mean of the simulated process.
- **method**: which method to use to generate the VAR matrix. Possible values are "normal" or "bimodal".
- **covariance**: type of covariance matrix to use in the simulation. Possible values: "toeplitz", "block1", "block2" or simply "diagonal".
- ***...***: the options for the simulation. These are: muMat: the mean of the entries of the VAR matrices; sdMat: the sd of the entries of the matrices;

### Value
- **A**: a list of NxN matrices ordered by lag
- **data**: a list with two elements: series the multivariate time series and noises the time series of errors
- **S**: the variance/covariance matrix of the process
sparsevar

Description

It performs the estimation of the matrices of the models using penalized least squares methods such as LASSO, SCAD and MCP.

sparsevar functions

fitVAR, fitVECM, simulateVAR, createSparseMatrix, plotMatrix, plotVAR, plotVECM l2norm, l1norm, lInfyNorm, maxNorm, frobNorm, spectralRadius, spectralNorm, impulseResponse

spectralNorm

Description

Compute the spectral norm of M

Usage

spectralNorm(M)

Arguments

M the matrix (real or complex valued)

spectralRadius

Description

Compute the spectral radius of M

Usage

spectralRadius(M)

Arguments

M the matrix (real or complex valued)
testGranger

Test for Granger Causality

Description

This function should retain only the coefficients of the matrices of the VAR that are statistically
significative (from the bootstrap)

Usage

testGranger(v, eb)

Arguments

v
the VAR object as from fitVAR or simulateVAR

eb
the error bands as obtained from errorBands

transformData

Transform data

Description

Transform the input data

Usage

transformData(data, p, opt)

Arguments

data
the data

p
the order of the VAR

opt
a list containing the options
\textbf{varENET} \hspace{2cm} \textit{VAR ENET}

\textbf{Description}

Estimate VAR using ENET penalty

\textbf{Usage}

\texttt{varENET(data, p, lambdas, opt)}

\textbf{Arguments}

- \texttt{data} \hspace{1cm} the data
- \texttt{p} \hspace{1cm} the order of the VAR
- \texttt{lambdas} \hspace{1cm} a vector containing the lambdas to be used in the fit
- \texttt{opt} \hspace{1cm} a list containing the options

\textbf{varMCP} \hspace{2cm} \textit{VAR MCP}

\textbf{Description}

Estimate VAR using MCP penalty

\textbf{Usage}

\texttt{varMCP(data, p, lambdas, opt)}

\textbf{Arguments}

- \texttt{data} \hspace{1cm} the data
- \texttt{p} \hspace{1cm} the order of the VAR
- \texttt{lambdas} \hspace{1cm} a vector containing the lambdas to be used in the fit
- \texttt{opt} \hspace{1cm} a list containing the options
Description

Estimate VAR using SCAD penalty

Usage

```r
varSCAD(data, p, lambdas, opt, penalty)
```

Arguments

- **data**: the data
- **p**: the order of the VAR
- **lambdas**: a vector containing the lambdas to be used in the fit
- **opt**: a list containing the options
- **penalty**: a string "SCAD" or something else
Index

accuracy, 2
bootstrappedVAR, 3
checkImpulseZero, 3
checkIsVar, 4
companionVAR, 4
computeForecasts, 4
createSparseMatrix, 5
decomposePi, 6
errorBandsIRF, 6
fitVAR, 7
fitVARX, 8
fitVECM, 9
frobNorm, 9
impulseResponse, 10
informCrit, 10
l1norm, 11
l2norm, 11
lInftyNorm, 11
maxNorm, 12
mcSimulations, 12
multiplot, 13
plotIRF, 14
plotIRFGrid, 14
plotMatrix, 15
plotVAR, 15
plotVECM, 16
simulateVAR, 16
simulateVARX, 17
sparsevar, 18
spectralNorm, 18
spectralRadius, 18
testGranger, 19
transformData, 19
varENET, 20
varMCP, 20
varSCAD, 21