

# Package ‘sparsevar’

August 19, 2019

**Version** 0.0.11

**Date** 2019-08-19

**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, testthat

**Depends** R (>= 1.8.0)

**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  
<doi:10.1214/15-AOS1315>.

**License** GPL-2

**URL** <http://github.com/svazzole/sparsevar>

**BugReports** <http://github.com/svazzole/sparsevar>

**VignetteBuilder** knitr

**RoxygenNote** 6.1.1

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Simone Vazzoler [aut, cre]

**Repository** CRAN

**Date/Publication** 2019-08-19 21:10:03 UTC

## R topics documented:

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

checkIsVar . . . . .	4
companionVAR . . . . .	4
computeForecasts . . . . .	4
createSparseMatrix . . . . .	5
decomposePi . . . . .	5
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 . . . . .	13
plotIRFGrid . . . . .	14
plotMatrix . . . . .	14
plotVAR . . . . .	15
plotVECM . . . . .	15
simulateVAR . . . . .	16
simulateVARX . . . . .	17
sparsevar . . . . .	18
spectralNorm . . . . .	18
spectralRadius . . . . .	18
testGranger . . . . .	19
transformData . . . . .	19
varENET . . . . .	20
varMCP . . . . .	20
varSCAD . . . . .	21

**Index** **22**

---

accuracy

*Accuracy metric*

---

**Description**

Compute the accuracy of a fit

**Usage**

accuracy(referenceM, A)

**Arguments**

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

---

bootstrappedVAR	<i>Bootstrap VAR</i>
-----------------	----------------------

---

**Description**

Build the bootstrapped series from the original var

**Usage**

```
bootstrappedVAR(v)
```

**Arguments**

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

---

checkImpulseZero	<i>Check Impulse Zero</i>
------------------	---------------------------

---

**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	<i>Check is var</i>
------------	---------------------

---

**Description**

Check if the input is a var object

**Usage**

```
checkIsVar(v)
```

**Arguments**

v	the object to test
---	--------------------

---

companionVAR	<i>Companion VAR</i>
--------------	----------------------

---

**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	<i>Computes forecasts for VARs</i>
------------------	------------------------------------

---

**Description**

This function computes forecasts for a given VAR.

**Usage**

```
computeForecasts(v, numSteps)
```

**Arguments**

v	a VAR object as from fitVAR.
numSteps	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, sparsity, method = "normal", stationary = FALSE,
  p = 1, ...)
```

### 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 and the standard deviation sd).

### 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	<i>Error bands for IRF</i>
---------------	----------------------------

---

**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**

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

**Arguments**

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

**Value**

`A` the list (of length `p`) 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

`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 MCP+.

**Usage**

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

**Arguments**

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

**Value**

A the list (of length `p`) 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

`residuals` the time series of the residuals



---

fitVECM	<i>Multivariate VECM estimation</i>
---------	-------------------------------------

---

**Description**

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

**Usage**

```
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**

Pi the matrix  $P_i$  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	<i>Frobenius norm of a matrix</i>
----------	-----------------------------------

---

**Description**

Compute the Froebenius norm of M

**Usage**

```
frobNorm(M)
```

**Arguments**

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

---

impulseResponse	<i>Impulse Response Function</i>
-----------------	----------------------------------

---

**Description**

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

**Usage**

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

**Arguments**

v	the data in the for of a VAR
len	length of the impulse response function

**Value**

irf a 3d array containing the impulse response function.

---

informCrit	<i>Computes information criterias for VARs</i>
------------	--

---

**Description**

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

**Usage**

```
informCrit(v)
```

**Arguments**

v	a list of VAR objects as from fitVAR.
---	---------------------------------------

---

11norm	<i>L1 matrix norm</i>
--------	-----------------------

---

**Description**

Compute the L1 matrix norm of M

**Usage**

11norm(M)

**Arguments**

M                    the matrix (real or complex valued)

---

12norm	<i>L2 matrix norm</i>
--------	-----------------------

---

**Description**

Compute the L2 matrix norm of M

**Usage**

12norm(M)

**Arguments**

M                    the matrix (real or complex valued)

---

1InfTyNorm	<i>L-infinity matrix norm</i>
------------	-------------------------------

---

**Description**

Compute the L-infinity matrix norm of M

**Usage**

1InfTyNorm(M)

**Arguments**

M                    the matrix (real or complex valued)

---

maxNorm	<i>Max-norm of a matrix</i>
---------	-----------------------------

---

**Description**

Compute the max-norm of M

**Usage**

```
maxNorm(M)
```

**Arguments**

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

---

mcSimulations	<i>Monte Carlo simulations</i>
---------------	--------------------------------

---

**Description**

This function generates monte carlo simultaions of sparse VAR and its estimation (at the moment only for VAR(1) processes).

**Usage**

```
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.
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 nMx5 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	<i>IRF grid plot</i>
-------------	----------------------

---

**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 indeces 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	<i>Matrix plot</i>
------------	--------------------

---

**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	<i>VAR simulation</i>
-------------	-----------------------

---

### 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;

### 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



---

simulateVARX	<i>VARX simulation</i>
--------------	------------------------

---

**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	<i>sparsevar: A package to estimate multivariate time series models (such as VAR and VECM), under the sparsity hypothesis.</i>
-----------	--

---

**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, l1InftyNorm, maxNorm, frobNorm, spectralRadius, spectralNorm, impulseResponse

---

spectralNorm	<i>Spectral norm</i>
--------------	----------------------

---

**Description**

Compute the spectral norm of M

**Usage**

spectralNorm(M)

**Arguments**

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

---

spectralRadius	<i>Spectral radius</i>
----------------	------------------------

---

**Description**

Compute the spectral radius of M

**Usage**

spectralRadius(M)

**Arguments**

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

---

testGranger	<i>Test for Ganger Causality</i>
-------------	----------------------------------

---

**Description**

This function should retain only the coefficients of the matrices of the VAR that are statistically significant (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	<i>Transorm data</i>
---------------	----------------------

---

**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

---

`varENET`*VAR ENET*

---

**Description**

Estimate VAR using ENET penalty

**Usage**

```
varENET(data, p, lambdas, opt)
```

**Arguments**

<code>data</code>	the data
<code>p</code>	the order of the VAR
<code>lambdas</code>	a vector containing the lambdas to be used in the fit
<code>opt</code>	a list containing the options

---

`varMCP`*VAR MCP*

---

**Description**

Estimate VAR using MCP penalty

**Usage**

```
varMCP(data, p, lambdas, opt)
```

**Arguments**

<code>data</code>	the data
<code>p</code>	the order of the VAR
<code>lambdas</code>	a vector containing the lambdas to be used in the fit
<code>opt</code>	a list containing the options

---

varSCAD	<i>VAR SCAD</i>
---------	-----------------

---

**Description**

Estimate VAR using SCAD penalty

**Usage**

```
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, 5

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, 13

plotIRFGrid, 14

plotMatrix, 14

plotVAR, 15

plotVECM, 15

simulateVAR, 16

simulateVARX, 17

sparsevar, 18

sparsevar-package (sparsevar), 18

spectralNorm, 18

spectralRadius, 18

testGranger, 19

transformData, 19

varENET, 20

varMCP, 20

varSCAD, 21