

sparsevar package

2016-05-04

Introduction

`sparsevar` is an R package that estimates sparse VAR and VECM model using penalized least squares methods (PLS).

The authors of `sparsevar` are [...] and the R package is maintained by Simone Vazzoler.

Installation

The simplest way to install the package is by using the CRAN repositories, by typing in the R console

```
install.packages("sparsevar", repos = "http://cran.us.r-project.org")
```

It is also possible to install the developing version of the package by typing

```
install.packages("devtools", repos = "http://cran.us.r-project.org")
devtools::install_github("svazzole/sparsevar")
```

Quick start

To load the `sparsevar` package simply type

```
suppressMessages(library(sparsevar))
```

Using a function included in the package, we simply generate a 20×20 VAR(2) process

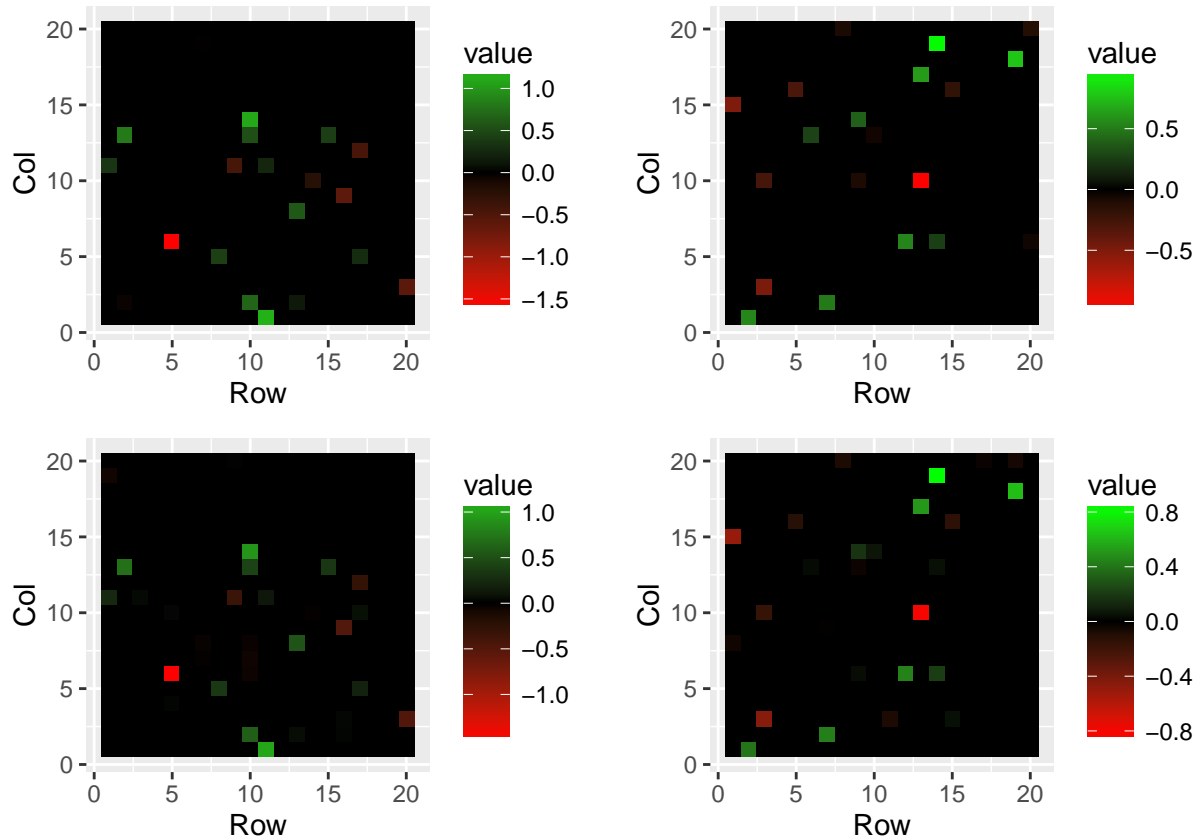
```
set.seed(1)
sim <- simulateVAR(N = 20, p = 2)
```

and we can estimate the matrices of the process using

```
est <- estimateVAR(sim$data$series, p = 2, options = list(foldsIDs = TRUE))
```

The results can be seen by plotting the matrices

```
plotComparisonVAR(sim, est)
```



Description of the package's functions

Estimation of a VAR or VECM models

Use `estimateVAR` for VAR model estimation or `estimateVECM` for VECM estimation.

The common arguments for the two functions are:

- **data**: the multivariate time series (variables in columns, observations in rows);
- **p**: the order of the VAR model to be estimated; default `p=1` for `estimateVAR` and `p=2` for `estimateVECM`.
- **penalty**: the penalty used in least squares. Possible values are: "ENET", "SCAD" or "MCP";
- **options**: list of options. Some of them depend on the penalty and some are global.

Global options

- **parallel**: TRUE or FALSE (default). Parallel cross-validation (on the folds);
- **ncores**: if `parallel = TRUE` then you must specify the number of cores used for the parallelization (default = 1).
- **nfolds**: number of folds to use in the cross validation (default `nfolds = 10`)
- **threshold**: TRUE or FALSE (default). If TRUE all the elements of the VAR/VECM matrices that are small "enough" are set to 0.

Options for penalty = "ENET"

- **lambda**: "lambda.min" (default) or "lambda.1se";

- **alpha**: a value in (0,1) (default **alpha** = 1). **alpha** = 1 is LASSO regression, **alpha** = 0 is Ridge LS;
- **type.measure**: "mse" (default) or "mae";
- **nlambda**: number of lambdas used for cross validation.
- **repeatedCV**: TRUE or FALSE; if TRUE repeated cross-validation is performed [...]
- **foldsID**: the vector containing the IDs for the folds in the cross validation.

Options for **penalty** = "SCAD" or "MCP"

- **eps**: convergence tolerance

Output

- **A**: a list of length **p** containing the matrices estimated for the VAR(p) model;
- **time**: elapsed time for the estimation;
- **mse**: the mean square error of the cross validation;
- **fit**: the estimated model.

Simulation of VAR models

Use `simulateVAR`. The parameters for the function are:

- **N**: the dimension of the process;
- **nobs**: the number of observations of the process;
- **rho**: the variance/covariance "intensity";
- **sparsity**: the percentage of non zero elements in the matrix of the VAR;
- **method**: "normal" or "bimodal".

Estimations' examples

```
results <- estimateVAR(rets)
```

will estimate VAR(1) process using LASSO regression on the dataset **rets**.

The command

```
results <- estimateVAR(rets, p = 3, penalty = "ENET",
                      options = list(parallel = TRUE, ncores = 5, alpha = 0.95,
                                     type.measure = "mae", lambda = "lambda.1se"))
```

will estimate a VAR(3) model on the dataset **rets** using the penalty "ENET" with **alpha** = 0.95 (between LASSO and Ridge). For the cross validation it will use "mae" (mean absolute error) instead of mean square error and it will choose as model the one correspondent to the lambda which is at 1 std deviation from the minimum. Moreover it will parallelize the cross validation over 5 cores.

Simulations' examples

```
sim <- simulateVAR(N = 100, nobs = 250, rho = 0.75, sparsity = 0.05, method = "normal")
```