Package ‘graphicalVAR’

July 3, 2016

Type Package

Title Graphical VAR for Experience Sampling Data

Version 0.1.4

Author Sacha Epskamp

Maintainer Sacha Epskamp <mail@sachaepskamp.com>

Description Estimates within and between time point interactions in experience sampling data, using the Graphical VAR model in combination with LASSO and EBIC.

License GPL (>= 2)

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp (>= 0.11.3), Matrix, glasso, glmnet, mvtnorm, qgraph (>= 1.3.1), dplyr, methods

Depends R (>= 3.1.0)

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-07-03 10:54:01

R topics documented:

graphicalVAR *Estimate the graphical VAR model.*

Description

Estimates the graphical VAR (Wild et al., 2010) model through LASSO estimation coupled with extended Bayesian information criterion for choosing the optimal tuning parameters. The estimation procedure is outlined by Rothman, Levina and Zhu (2010) and is further described by Abegaz and Wit (2013). The procedure here is based on the work done in the R package SparseTSCGM (Abegaz and Wit, 2014).

Usage

```
graphicalVAR(data, nLambda = 50, verbose = TRUE, gamma = 0.5, scale
                 = TRUE, lambda_beta, lambda_kappa, maxit.in = 100,
                 maxit.out = 100, deleteMissings = TRUE,
                 penalize.diagonal = TRUE, lambda_min_kappa = 0.05,
                 lambda_min_beta = 0.05)
```

Arguments

`data`	A matrix or data frame containing repeated measures (rows) on a set of variables (columns). Must not contain missing data.
`nLambda`	The number of both lambda parameters to test. Defaults to 50, which results in 2500 models to evaluate.
`verbose`	Logical, should a progress bar be printed to the console?
`gamma`	The EBIC hyper-parameter. Set to 0 to use regular BIC.
`scale`	Logical, should responses be standardized before estimation?
`lambda_beta`	An optional vector of lambda_beta values to test. Set `lambda_beta = 0` argument and `lambda_kappa = 0` for unregularized estimation.
`lambda_kappa`	An optional vector of lambda_kappa values to test. Set `lambda_beta = 0` argument and `lambda_kappa = 0` for unregularized estimation.
`maxit.in`	Maximum number of iterations in the inner loop (computing beta)
`maxit.out`	Maximum number of iterations in the outer loop
`deleteMissings`	Logical, should missing responses be deleted?
`penalize.diagonal`	Logical, should the diagonal of beta be penalized (i.e., penalize auto-regressions)?
`lambda_min_kappa`	Multiplier of maximal tuning parameter for kappa
`lambda_min_beta`	Multiplier of maximal tuning parameter for beta

Details

Let y_t denote the vector of centered responses of a subject on a set of items on time point t. The graphical VAR model, using only one lag, is defined as follows:

y_t = Beta y_y-1 + epsilon_t

In which epsilon_t is a vector of error and is independent between time points but not within time points. Within time points, the error is normally distributed with mean vector 0 and precision matrix (inverse covariance matrix) Kappa. The Beta matrix encodes the between time point interactions and the Kappa matrix encodes the within time point interactions. We aim to find a sparse solution for both Beta and Kappa, and do so by applying the LASSO algorithm as detailed by Rothman, Levina and Zhu (2010). The LASSO algorithm uses two tuning parameters, lambda_beta controlling the sparsity in Beta and lambda_kappa controlling the sparsity in Kappa. We estimate the model under a (by default) 50 by 50 grid of tuning parameters and choose the tuning parameters that optimize the extended Bayesian Information Criterion (EBIC; Chen and Chen,2008).

After estimation, the Beta and Kappa matrices can be standardized as described by Wild et al. (2010). The Kappa matrix can be standardized to partial contemporaneous correlations (PCC) as follows:

PCC(y_i,t, y_j,t) = - kappa_ij / (sqrtkappa_ii kappa_jj)

Similarly, the beta matrix can be standardized to partial directed correlations (PDC):

PDC(y_i,t-1, y_j,t) = beta_ji / sqrtsigma_jj kappa_ii + beta_ji^2

In which sigma is the inverse of kappa. Note that this process transposes the beta matrix. This is done because in representing a directed network it is typical to let rows indicate the node of origin and columns the node of destination.

Set `lambda_beta = 0` argument and `lambda_kappa = 0` for unregularized estimation.

Value

A `graphicalVAR` object, which is a list containing:

`PCC`	The partial contemporaneous correlation network
`PDC`	The partial directed correlation network
`beta`	The estimated beta matrix
`kappa`	The estimated kappa matrix
`EBIC`	The optimal EBIC
`path`	Results of all tested tuning parameters
`labels`	A vector containing the node labels

Author(s)

Sacha Epskamp <mail@sachaepskamp.com>

References

Chen, J., & Chen, Z. (2008). Extended Bayesian information criteria for model selection with large model spaces. Biometrika, 95(3), 759-771.

Fentaw Abegaz and Ernst Wit (2013). Sparse time series chain graphical models for reconstructing genetic networks. Biostatistics. 14, 3: 586-599.

Fentaw Abegaz and Ernst Wit (2014). SparseTSCGM: Sparse time series chain graphical models. R package version 2.1.1. http://CRAN.R-project.org/package=SparseTSCGM

Rothman, A.J., Levina, E., and Zhu, J. (2010). Sparse multivariate regression with covariance estimation. Journal of Computational and Graphical Statistics. 19: 947-962.

Wild, B., Eichler, M., Friederich, H. C., Hartmann, M., Zipfel, S., & Herzog, W. (2010). A graphical vector autoregressive modelling approach to the analysis of electronic diary data. BMC medical research methodology, 10(1), 28.

Examples

```
# Simulate model:
Mod <- randomGVARmodel(4,probKappaEdge = 0.8,probBetaEdge = 0.8)

# Simulate data:
Data <- graphicalVARsim(100,Mod$beta,Mod$kappa)

# Estimate model:
Res <- graphicalVAR(Data, gamma = 0, nLambda = 10)

# Plot results:
layout(t(1:2))
plot(Mod, "PCC", layout = "circle")
plot(Res, "PCC", layout = "circle")

plot(Mod, "PDC", layout = "circle")
plot(Res, "PDC", layout = "circle")
```

---

graphicalVARsim *Simulates data from the graphical VAR model*

Description

Simulates data from the graphical VAR model, see `graphicalVAR` for details.

Usage

```
graphicalVARsim(nTime, beta, kappa, mean = rep(0, ncol(kappa)), init =
                   mean, warmup = 100, lbound = rep(-Inf, ncol(kappa)),
                   ubound = rep(Inf, ncol(kappa)))
```

Arguments

`nTime`	Number of time points to sample
`beta`	The Beta matrix to use
`kappa`	The Kappa matrix to use
`mean`	Means to use
`init`	Initial values
`warmup`	The amount of samples to use as warmup (not returned)
`lbound`	Lower bound, at every time point values below this bound are set to the bound.
`ubound`	Upper bound, at every time point values above this bound are set to the bound.

Value

A matrix containing the simulated data.

Author(s)

Sacha Epskamp <mail@sachaepskamp.com>

`plot.graphicalVAR` *Plot method for graphicalVAR objects*

Description

Sends the estimated PCC and PDC networks to `qgraph`.

Usage

```
## S3 method for class 'graphicalVAR'
plot(x, include = c("PCC", "PDC"), repulsion = 1,
                horizontal = TRUE, titles = TRUE, sameLayout = TRUE,
                unweightedLayout = FALSE, ...)
```

Arguments

`x`	A `graphicalVAR` object
`include`	A vector of at most two containing `"PCC"` and `"PDC"` indicating which networks should be plotted and in what order.
`repulsion`	The repulsion argument used in `qgraph`
`horizontal`	Logical, should the networks be plotted horizontal or vertical?
`titles`	Logical, should titles be added to the plots?
`sameLayout`	Logical, should both networks be plotted in the same layout?
`unweightedLayout`	Logical, should the layout be based on the unweighted network instead of the weighted network?
`...`	Arguments sent to `qgraph`

Author(s)

Sacha Epskamp <mail@sachaepskamp.com>

`print.graphicalVAR` *S3 methods for graphicalVAR objects.*

Description

Prints a short overview of the results of `graphicalVAR`

Usage

```
## S3 method for class 'graphicalVAR'
print(x, ...)
## S3 method for class 'graphicalVAR'
summary(object, ...)
```

Arguments

`x`	A `graphicalVAR` object
`object`	A `graphicalVAR` object
`...`	Not used.

Author(s)

Sacha Epskamp <mail@sachaepskamp.com>

`randomGVARmodel` *Simulate a graphical VAR model*

Description

Simulates an contemporaneous and temporal network using the method described by Yin and Li (2001)

Usage

```
randomGVARmodel(Nvar, probKappaEdge = 0.1, probKappaPositive = 0.5, probBetaEdge = 0.1,
      probBetaPositive = 0.5, maxtry = 10, kappaConstant = 1.1)
```

Arguments

`Nvar`	Number of variables
`probKappaEdge`	Probability of an edge in contemporaneous network
`probKappaPositive`	Proportion of positive edges in contemporaneous network
`probBetaEdge`	Probability of an edge in temporal network
`probBetaPositive`	Propotion of positive edges in temporal network
`maxtry`	Maximum number of attempts to create a stationairy VAR model
`kappaConstant`	The constant used in making kappa positive definite. See Yin and Li (2001)

Details

The resulting simulated networks can be plotted using the plot method.

Value

A list containing:

`kappa`	True kappa structure (residual inverse variance-covariance matrix)
`beta`	True beta structure
`PCC`	True partial contemporaneous correlations
`PDC`	True partial temporal correlations

Author(s)

Sacha Epskamp

References

Yin, J., & Li, H. (2011). A sparse conditional gaussian graphical model for analysis of genetical genomics data. The annals of applied statistics, 5(4), 2630-2650.

Index