Package 'GIMMEgVAR'

July 21, 2025

Type Package

Title Group Iterative Multiple Model Estimation with 'graphicalVAR'

Version 0.1.0

Imports graphicalVAR, here, qgraph, png

Description

Data-driven approach for arriving at person-specific time series models from within a Graphical Vector Autoregression (VAR) framework. The method first identifies which relations replicate across the majority of individuals to detect signal from noise. These group-level relations are then used as a foundation for starting the search for person-specific (or individual-level) relations. All estimates are obtained uniquely for each individual in the final models. The method for the 'graphicalVAR' approach is found in Epskamp, Waldorp, Mottus & Borsboom (2018) <doi:10.1080/00273171.2018.1454823>.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

NeedsCompilation no

Suggests knitr, rmarkdown

VignetteBuilder knitr

Author Sandra Williams Lee [aut, cre],
Kathleen M. Gates [aut]

Maintainer Sandra Williams Lee <wsandra@live.unc.edu>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2024-05-16 14:10:17 UTC

Contents

`environmentSetup` *GIMMEgVAR: Environment Setup*

Description

Loads needed packages and creates directories where logfiles and output from 'GIMMEgVAR' will be stored

Usage

```
environmentSetup(outputPath)
```

Arguments

`outputPath`	The user specified path to the directory where results files should be stored.

`fitGIMMEgVAR` *FIT 'GIMMEgVAR'*

Description

This function fits 'GIMMEgVAR' using the original input data supplied by the user and the logical matrices that result across people after fitting individual 'graphicalVAR'. The logical matrices are determined via the gimmeThreshold specified by the user

Usage

```
fitGIMMEgVAR(
  inData,
  variableNames,
  gimmeGVARThreshold = 0.5,
  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 = lambda_min_kappa,
  mimic = c("current", "0.1.2", "0.1.4", "0.1.5", "0.2"),
  beepvar,
  dayvar,
  idvar,
  lags = 1,
  centerWithin = TRUE,
  likelihood = c("unpenalized", "penalized"),
  logicals,
  RES_matrixData,
  outputPath,
  labelNames = variableNames
)
```

Arguments

inData	A matrix or data frame containing repeated measures (rows) on a set of variables (columns). Must not contain missing data.
variableNames	The vector containing name of variables to be analyzed
gimmeGVARThreshold	The cutoff value for group-level paths. Defaults to .50, indicating that a path must be non-zero across >= .50 included as a group-level path.
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
mimic	Allows one to mimic earlier versions of 'graphicalVAR'

beepvar	String indicating assessment beep per day (if missing, is added). Adding this argument will cause non-consecutive beeps to be treated as missing!
dayvar	String indicating assessment day. Adding this argument makes sure that the first measurement of a day is not regressed on the last measurement of the previous day. IMPORTANT: only add this if the data has multiple observations per day.
idvar	String indicating the subject ID
lags	Vector of lags to include
centerWithin	logical, should subject data be within-person centered before estimating fixed effects?
likelihood	Should likelihood be computed based on penalized contemporaneous matrix or
logicals	Logical matrices that determine which elements of the beta and kappa matrices will be regularized when fititng 'GIMMEgVAR'. The matrices are determined in the algorithm by obtaining counts across all fitted individual network results for beta and kappa, respectively. Elements with proportions >= gimmeGVARThreshold value will not be regularized when fitting 'GIMMEgVAR'.
RES_matrixData	List containing results of fitting individual 'graphicalVAR' to each individual.
outputPath	The user specified path to the directory where results files should be stored.
labelNames	Vector of names used to label nodes in Network graph. Defaults to variable names if no vector is supplied.

Details

The user can pass in additional 'graphicalVAR' options as specified in the R package 'graphicalVAR' created by Sacha Epskamp. See 'graphicalVAR' documentation for details.

The following results are returned in the 'GIMMEgVAR' Files directory: (1) Data frames containing individual paths. These are the person-specific results for each individual obtained by not regularizing the group paths indicated by the logical matrices for both beta and kappa. They are prefaced RESULTS_GIMMEgVAR_SUBJECT_.

(2) Data frames that indicate the proportion of paths across individuals: These are dataframes containing the proportion of paths present across individuals after fitting 'GIMMEgVAR'. These proportions are used to determine the final group model. They are named proportionKappa.RData and proportionBeta.RData.

(3) Group path data frames for beta and kappa: These are dataframes with 1 indicating the presence of a group path and 0 indicating the absence of a group path. These data frames are derived from #2 above for both beta and kappa. They are the dataframes used to graph the final group models and are prefaced GIMMEgVAR_RESULTS_GRP_BETA_THRESHOLD_ and GIMMEgVAR_RESULTS_GRP_KAPPA_THRESHOLD_. The number following threshold indicates the gimmeThreshold used for the analysis.

(4) Final network graphs are added to the gimmegvarFiles folder in .png format. These contain overlay graphs of the group and individual paths for the beta and kappa network respectively.

`fitGraphicalVAR` *fitgraphicalVAR*

Description

Fits 'graphicalVAR' with defaults used in 'graphicalVAR' package developed by Sacha Epskamp. See 'graphicalVAR' documentation by Sacha Epskamp for details.

Usage

```
fitGraphicalVAR(
  data,
  nLambda = 50,
  verbose = TRUE,
  gamma,
  scale = TRUE,
  lambda_beta,
  lambda_kappa,
  regularize_mat_beta,
  regularize_mat_kappa,
  maxit.in = 100,
  maxit.out = 100,
  deleteMissings = TRUE,
  penalize.diagonal = TRUE,
  lambda_min_kappa = 0.05,
  lambda_min_beta = lambda_min_kappa,
  mimic = c("current"),
  variableNames,
  beepvar,
  dayvar,
  idvar,
  lags = 1,
  centerWithin = TRUE,
  likelihood = c("unpenalized", "penalized"),
  outputPath
)
```

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.
regularize_mat_beta	A logical matrix indicating which elements of the beta matrix should be regularized (experimental).
regularize_mat_kappa	A logical matrix indicating which elements of the kappa matrix should be regularized (experimental).
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
mimic	Allows one to mimic earlier versions of 'graphicalVAR'
variableNames	The vector containing name of variables to be analyzed
beepvar	String indicating assessment beep per day (if missing, is added). Adding this argument will cause non-consecutive beeps to be treated as missing!
dayvar	String indicating assessment day. Adding this argument makes sure that the first measurement of a day is not regressed on the last measurement of the previous day. IMPORTANT: only add this if the data has multiple observations per day.
idvar	String indicating the subject ID
lags	Vector of lags to include
centerWithin	logical, should subject data be within-person centered before estimating fixed effects?
likelihood	Should likelihood be computed based on penalized contemporaneous matrix or
outputPath	The user specified path to the directory where results files should be stored.

Details

The following results are returned in the gvarFiles directory: (1) Separate data frames containing the usual individual 'graphicalVAR' results. These dataframes contain the person-specific results for each individual obtained by fitting 'graphicalVAR'. They are prefixed RESULTS_GVAR_SUBJECT The number of data frames returned equals the number of individuals whose models were successfully fitted from the input data file.

GIMMEgVAR *GIMMEgVAR ' This function calls all the functions needed to fit 'GIMMEgVAR'.*

Description

GIMMEgVAR ' This function calls all the functions needed to fit 'GIMMEgVAR'.

Usage

```
GIMMEgVAR(
  outputPath,
  data,
  nLambda = 50,
  verbose = TRUE,
  gamma,
  scale = TRUE,
  lambda_beta,
  lambda_kappa,
  regularize_mat_beta,
  regularize_mat_kappa,
  maxit.in = 100,
  maxit.out = 100,
  deleteMissings = TRUE,
  penalize.diagonal = TRUE,
  lambda_min_kappa = 0.05,
  lambda_min_beta = lambda_min_kappa,
  mimic = c("current"),
  variableNames,
  beepvar,
  dayvar,
  idvar,
  lags = 1,
  centerWithin = TRUE,
  likelihood = c("unpenalized", "penalized"),
  gimmeGVARThreshold = 0.5,
  labelNames = variableNames
)
```

Arguments

outputPath	The user specified path to the directory where results files should be stored.
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.

regularize_mat_beta — A logical matrix indicating which elements of the beta matrix should be regularized (experimental). Note: In 'GIMMEgVAR' this matrix is determined and set for the user based on their gimmeThreshold and the results of fitting individual 'graphicalVAR' across all individuals. The logical matrix used to fit determine which elements of the beta matrix are regularized is returned in the gimmegvarFiles folder and is named logicalKappa.

regularize_mat_kappa — A logical matrix indicating which elements of the kappa matrix should be regularized (experimental). Note: In 'GIMMEgVAR' this matrix is determined and set for the user based on their gimmeThreshold and the results of fitting individual 'graphicalVAR' across all individuals. The logical matrix used to fit determine which elements of the beta matrix are regularized is returned in the gvarFiles folder and is named logicalKappa.

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

mimic — Allows one to mimic earlier versions of graphicalVAR

variableNames — The vector containing name of variables to be analyzed in the network model

beepvar — String indicating assessment beep per day (if missing, is added). Adding this argument will cause non-consecutive beeps to be treated as missing!

dayvar — String indicating assessment day. Adding this argument makes sure that the first measurement of a day is not regressed on the last measurement of the previous day. IMPORTANT: only add this if the data has multiple observations per day.

idvar — String indicating the subject ID

lags — Vector of lags to include

centerWithin — logical, should subject data be within-person centered before estimating fixed effects?

likelihood — Should likelihood be computed based on penalized contemporaneous matrix or unpenalized contemporaneous matrix. Set to "penalized" to mimic version 2.5 and later of sparseTSCGM.

gimmeGVARThreshold
: The cutoff value for group-level paths. Defaults to .50, indicating that a path must be non-zero across >= .50 included as a group-level path.

labelNames
: Vector of names used to label nodes in Network graph. Defaults to variable names if no vector is supplied.

Value

Returns a list containing 7 elements. The first 6 elements are the group level results indicating the count, proportion and presence (coded 1 for present, 0 for absent) of the edges estimated by the algorithm in the kappa and beta matrices. These results are stored in matrix objects named pathCountBeta, pathProportionBeta, groupBeta and pathCountKappa, pathProportionKappa, and groupKappa respectively and are used to construct the final group level network. The remaining element is a list which contains the person-specific network results for every subject in the data. This is estimated using information from the group-level network model. Along with the the data used to estimate the networks, it contains the following for each subject:

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

data
: A "tsData" object detailing the features of the data used for estimating the network

---

graphKappa *GIMMEgVAR: GraphGIMMEgVAR*

Description

Produces network graphs for beta and kappa matrices as .png files.

Usage

```
graphKappa(
  outputPathGIMMEgVAR,
  gimmeGVARThreshold,
  pathProportionKappa,
  lableNames,
  groupKappa
)
```

Arguments

`outputPathGIMMEgVAR`
: Directory where network graphs will be saved.

`gimmeGVARThreshold`
: The cutoff value for group-level paths. Defaults to .50, indicating that a path must be non-zero across >= .50 included as a group-level path.

`pathProportionKappa`
: The proportion of individuals identified as having a path in the Kappa matrix. Used to determine group-level paths in the final group Kappa network.

`lableNames`
: Vector of names used to label nodes in Network graph. Defaults to variable names if no vector is supplied.

`groupKappa`
: Matrix that identifies the presence of a group path in the Kappa matrix. 0 indicates no group path present, 1 indicates group path present.

Details

Group and individual network paths are overlayed to form a single network graph for each matrix, respectively.

Kappa matrix is depicted using solid lines, beta network is depicted using dashed lines. In both networks, group paths appear as black lines. Individual paths appear as light grey lines.

For individual paths, line thickness is weighted by the proportion of individuals with paths present.

`prepareData` *PREPARE DATA FOR FITTING 'GIMMEgVAR'*

Description

Reads in results data from fitting 'graphicalVAR' and creates the logical matrices for kappa and beta needed to fit 'GIMMEgVAR'. The logical matrices determine which paths will not be regularized in the fitting of 'GIMMEgVAR' to determine the group level network model.

Usage

```
prepareData(gimmeGVARThreshold = 0.5, RES_matrixData, outputPath)
```

Arguments

`gimmeGVARThreshold`
: The cutoff value for group-level paths. Defaults to .50, indicating that a path must be non-zero across >= .50 included as a group-level path.

`RES_matrixData`
: List containing results of fitting individual 'graphicalVAR' to each individual.

`outputPath`
: The user specified path to the directory where results files should be stored.

Details

The following results are returned in the gvarFiles directory: (1) Returns two data frames that hold the logical matrices for beta and kappa. These are named logicalBeta and logicalKappa

(2) Returns two data frames that indicate the proportion of paths across individuals. These are data frames containing the proportion of paths present across individuals after fitting 'graphicalVAR'. These proportions are used to determine potential group paths which will not be regularized for individuals when fitting the final 'GIMMEgVAR'. The output files are named proportionKappa.RData and proportionBeta.RData

---

`sampleData` *sampleData*

Description

List containing 10 elements. Each element consists of a data frame with time series data that was generated via simulation.

Format

'sampleData' Each element in the list contains a data frame for an individual subject. Each data frame has 150 observations of 10 variables (V1-V10) that can be used for fitting GIMMEgVAR. The variable Idnum is the unique subject ID.

V1-V10 Ten variables availalbe for us in fitting GIMMEgVAR

IDnum Unique subject ID

Author(s)

Sandra AW Lee <wsandra@live.unc.edu>

Index